High-Precision Computation: Mathematical Physics and Dynamics

D.H. Bailey[*] J.M. Borwein[†] R. Barrio[‡]

October 8, 2009

Abstract

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, studies of the fine structure constant, scattering amplitudes of quarks, gluons and bosons, nonlinear oscillator theory, experimental mathematics, evaluation of orthogonal polynomials, numerical integration of ODEs, computation of periodic orbits, studies of the splitting of separatrices, detection of SNAs, Ising theory, quantum field theory, and discrete dynamical systems. We conclude that high-precision arithmetic facilities are now an indispensable component of a modern large-scale scientific computing environment.

[*] 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, Callaghan, NSW 2308, Australia, jonathan.borwein@newcastle.edu.au jborwein@cs.dal.ca. Supported in part by the Australian Research Council and the Canada Research Chair Programme.

[‡] GME, Depto. Matemática Aplicada and IUMA, Universidad de Zaragoza, E-50009 Zaragoza, Spain rbarrio@unizar.es. Supported in part by the Spanish research project MTM2009-10767.

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. At the same time, the majority of persons performing numerical computations at the present time 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 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.

As a simple example of such difficulties, consider the following innocuous-looking problem. Suppose we wish to fit the following data to a polynomial: $5, 2304, 118101, 1838336, 14855109, 79514880, 321537749, 1062287616, 3014530821$, for integer arguments $(0, ..., 8)$. The usual approach is to employ polynomial least squares curve fitting, which amounts to solving a $(n + 1 \times n + 1)$ linear system of equations (written in matrix form):

$$\begin{bmatrix} n & \sum_{k=1}^n x_k & \cdots & \sum_{k=1}^n x_k^n \\ \sum_{k=1}^n x_k & \sum_{k=1}^n x_k^2 & \cdots & \sum_{k=1}^n x_k^{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^n x_k^n & \sum_{k=1} x_k^{n+1} & \cdots & \sum_{k=1}^n x_k^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^n y_k \\ \sum_{k=1}^n x_k y_k \\ \vdots \\ \sum_{k=1}^n x_k^n y_k \end{bmatrix}.$$

In a computer implementation of this algorithm, the linear equation solution is most commonly (and most wisely) done with library software, such as the Linpack [44] or LAPACK [43]. But whether or not library software is used, a double-precision implementation of this algorithm fails to find the correct underlying polynomial coefficients. However, an implementation of this scheme using “double-double” precision (i.e., roughly 31-digit precision) correctly deduces that the original data sequence is given by the polynomial function

$$f(k) = 5 + 220k^2 + 990k^4 + 924k^6 + 165k^8.$$

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. One concrete illustration of these difficulties is provided by the following example, for which the authors are indebted to Bastian Pentenrieder of

ETH Zurich. 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

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

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

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

where $n \times n$ is the size of the linear system above . Note that even when $n = 10^7$, which is a fairly modest size compared to some systems currently 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 [28], and software packages implementing these schemes have been available since the early days of computing. 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.

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 [14, 38]. 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 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 [40].

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 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." [32, 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 [41]. We also mention the recent studies of W. Hayes [39], where some comparisons of the stability of the Solar System is performed using various numerical ODE integrators and checked via a high-precision integration done using a Taylor series integrator.

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 [12] and [33].

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 [49]. 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 [50].

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 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 [30].

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 [24].

Two other recent examples of employing high-precision arithmetic in fundamental physics calculations of this type are [45] and [29].

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)

$$c = 0.60544365719673274947892284244720747522208996\ldots$$

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 [7].

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 [25][4]. One of the key techniques used here is the PSLQ integer relation detection algorithm [9]. 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

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

(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 [3][25, pg. 135-143]. Since 1996, numerous other formulas of this type have been found, using the PSLQ-based computational approach, and then subsequently proven [25, 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 [10][25, 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} [11][25, pg. 174–178].

3.9 Evaluating orthogonal polynomials

The use of the classical families of orthogonal polynomials has been extended to almost all mathematical and physical disciplines, including approximation theory, spectral methods, representation of potentials and others. In the last few years, researchers have studied orthogonal polynomials in Sobolev spaces. One particular case of interest is when measures related to derivatives are purely atomic, with a finite number of mass points. That is, given a set of K evaluation points $\{c_1, \ldots, c_K\}$ (the support of the discrete measure), a set of indexes that indicate the maximum order of derivatives in each evaluation point $\{r_1, \ldots, r_K\}$, and a set of non-negative coefficients $\{\lambda_{ji} \,|\, j = 1, \ldots, K;\; i = 0, \ldots, r_j\}$, we define the Sobolev inner product

$$\langle p, q\rangle_W = \int_{\mathbb{R}} p(x)\, q(x)\, \mathrm{d}\mu_0(x) + \sum_{j=1}^{K} \sum_{i=0}^{r_j} \lambda_{ji}\, p^{(i)}(c_j)\, q^{(i)}(c_j), \qquad \lambda_{ji} \geq 0. \tag{1}$$

This particular case is an important instance of the class of discrete Sobolev inner products. Note that the standard orthogonal polynomials are orthogonal with respect to a "standard" inner product

$$\langle p, q\rangle = \int_{\mathbb{R}} p(x)\, q(x)\, \mathrm{d}\mu_0(x), \tag{2}$$

where μ_0 is a positive Borel measure on the real line with infinitely many points at the support.

Sobolev orthogonal polynomials $\{q_n(x)\}$ satisfy a $(2g+1)$-term recurrence relation

$$h(x)\, q_{n-g}(x) = \sum_{k=\max\{0,n-2g\}}^{n} b_{n,k}\, q_k(x), \qquad (n \geq g). \tag{3}$$

The reference [16] presents the complete set of formulas to obtain the coefficients $\{b_{ij}\}$ of (3). In order to show the complexity of the process, the proposition below presents just one of the algorithms of [16] needed to obtain the coefficients in the general case $(n \geq g_i)$, respectively.

Proposition 1 *Let $\{q_0(x),\, q_1(x), \ldots,\, q_{g-1}(x)\}$ be a monic orthogonal polynomial basis of $\mathcal{P}_{g-1}$ (where $g := \mathrm{degree}(h_1(x))$ or $\mathrm{degree}(h_2(x))$) with respect to (1) and $\{p_0(x),\, p_1(x), \ldots,\, p_{g-1}(x)\}$ with respect to (2). Then*

$$
\begin{aligned}
&q_0(x) = 1, \\
&x\, q_{l-1}(x) = q_l(x) + \sum_{s=0}^{l-1} b_{l,s}\, q_s(x), \qquad 1 \le l < g,
\end{aligned}
$$

where $b_{l,s} = \delta_{l,s} + \dfrac{1}{\|q_s\|_W^2} \left\{ \displaystyle\sum_{m=s+1}^{l} \delta_{l,m} \sum_{j=1}^{K} \sum_{i=0}^{r_j} \lambda_{ji}\, p_m^{(i)}(c_j)\, q_s^{(i)}(c_j) \right\}$ *being*

$$
\begin{cases}
\delta_{l,l} & = 1, \\
\delta_{l,l-1} & = a_{l-1,l-2} + \beta_{l-1}, \\
\delta_{l,l-2} & = a_{l-1,l-3} + a_{l-1,l-2}\, \beta_{l-2} + \gamma_{l-1}, \\
\delta_{l,m} & = a_{l-1,m-1} + a_{l-1,m}\, \beta_m + a_{l-1,m+1}\, \gamma_{m+1}, \qquad m = s, \ldots, l-3,
\end{cases}
$$

with $a_{s,t}$ given by

$$
\begin{cases}
a_{s,t} = -\dfrac{1}{\|p_t(x)\|^2} \displaystyle\sum_{j=1}^{K} \sum_{i=0}^{r_j} \lambda_{ji}\, q_s^{(i)}(c_j)\, p_t^{(i)}(c_j), & t \ge 0, \\
a_{s,t} = 0, & t < 0.
\end{cases}
\tag{4}
$$

The above formulas to obtain the coefficients $\{b_{ij}\}$ are, in general, quite unstable numerically. The main reasons are the appearance of $\|p_i\|$ in the formulas and the necessity of computing derivatives of polynomials at the support of the discrete measures. It is well known that the evaluation of derivatives is a highly unstable problem and can lead to severe rounding errors. On the other hand, the L_2-norms $\|p_i\|$ decrease very fast in the case of Jacobi polynomials and grow in the case of Hermite and Laguerre polynomials. As a result, terms of very different sizes can appear, which result in numerical errors due to cancelation.

In Figure 1 we present the evaluation of the square of the L_2-norm, with respect to their own inner products, of the classical and the Sobolev polynomials of two families: Chebyshev and Hermite. The computations have been done by using 128 and 256 bits of precision in the mantissa (note that 53 bits is the standard double precision). Rounding errors render the computation completely inaccurate in some cases using 128 bits. One of the reasons is the decay of $\|p_i\|^2$, from 1 to 10^{-30}, which requires the use of a high precision. In the figures we have plotted both precisions (128 and 256 bits) in the cases with three mass points in the discrete measure. We observe that for low degrees both computations are similar, but for degrees higher than 15 the results are completely different (cases b-c, e-f), generating, in the case of 128 bits, inaccurate coefficients $\{b_{ij}\}$.

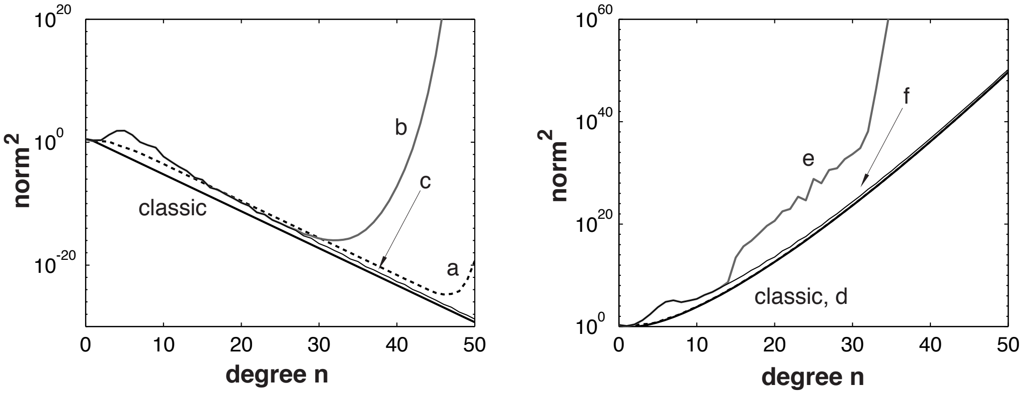


Figure 1: Evaluation (degree 0 to 50) of the square of the L_2-norm of four families of Sobolev orthogonal polynomials compared with the associated classical orthogonal polynomials. On the left, Chebyshev-Sobolev polynomials with: (a) one mass point $c = 1.5$ up to 1st derivative, $\lambda = 1/10$, using 128 bits, (b) three mass points $c_j = -1, 0, 0.5$ up to 3rd derivative, $\lambda_{ij} = 1/10$, using 128 bits, and (c) the same as (b) but using 256 bits. On the right, Hermite-Sobolev polynomials with: (d) one mass point $c = 1.5$ up to 1st derivative, $\lambda = 1/10$, using 128 bits, (e) three mass points $c_j = -1, 0, 0.5$ up to 5th derivative, $\lambda_{ij} = 1/10$, using 128 bits, and (f) the same as (e) but using 256 bits.

From Figure 1 it is clear that in the computation of the recurrence coefficients $\{b_{ij}\}$ it is necessary to use multiple-precision software. Besides, when we want to evaluate a finite series of Sobolev orthogonal polynomials it is necessary to control the rounding errors.

In Figure 2 we show the behavior of some theoretical error bounds [17]: T4 a backward error bound and T5 for the running error bound, and the relative error in a multiple-precision evaluation of a Sobolev series. Note that we present relative error bounds and relative rounding errors, that is, for $q(x) \not\approx 0$ we divide by $|q(x)|$. We have up to degree 50 of the function $f(x) = (x+1)^2 \sin(4x)$ in Chebyshev-Sobolev orthogonal polynomials, considering one mass point $c = 1$ up to first derivative in the discrete part of the inner product. In the figures on the left we use double precision (53 bits on the mantissa) and on the right we use multiple precision (96 bits on the mantissa for $x < -0.5$ (on the left of the vertical line) and 64 for $x > -0.5$). The turning point $x = -0.5$ is the point where the relative running error in double precision is greater than 10^{-10}. Therefore, from the figures we can observe how the combined use of rounding error bounds (in this case the running error bound) and multiple-precision libraries permits us to evaluate Sobolev series accurately.

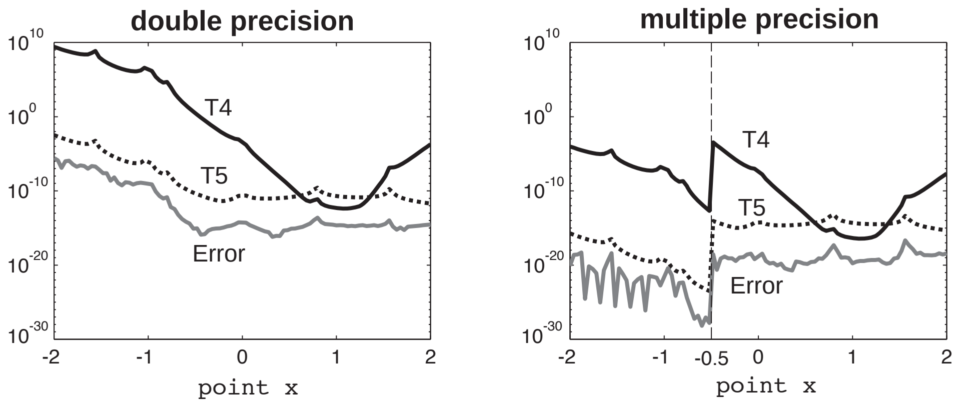


Figure 2: Behavior of the theoretical error bounds (T4 a backward error bound and T5 for the running error bound) and the relative error in the double- and multiple-precision evaluation of the Chebyshev-Sobolev approximation of degree 50 of the function $f(x) = (x+1)^2 \sin(4x)$, where the discrete Sobolev measure have one mass point $c = 1$ up to 1st derivative in the discrete part of the inner product. In the figure on the left we use double precision and on the right multiple-precision (on the left of the vertical line we use 96 bits on the mantissa and 64 on the right part).

3.10 High-precision solution of ODEs: Taylor method

In several applications of dynamical systems we need to integrate the relevant differential equation, normally for a short time, with very high precision. Moreover, in the study of the bifurcations and stability of periodic orbits (by instance) we also have to integrate the first order variational equations using as initial conditions the identity matrix. To reach this goal we may, obviously, use any numerical ODE method such as Runge-Kutta. During the last few years, the Taylor method has emerged as a preferred method in the computational dynamics community.

The Taylor method is one of the oldest numerical methods for solving ordinary differential equations, but it is scarcely used in the numerical analysis community. The formulation is quite simple [15, 19]. Let us consider the initial value problem $\dot{\boldsymbol{y}} = \boldsymbol{f}(t, \boldsymbol{y})$. Now, the value of the solution at t_i (that is, $\boldsymbol{y}(t_i)$) is approximated by $\boldsymbol{y}_i$ from the n-th degree Taylor series of $\boldsymbol{y}(t)$ at $t = t_i$ (the function $\boldsymbol{f}$ has to be a smooth function). So, denoting $h_i = t_i - t_{i-1}$,

$$\begin{aligned} \boldsymbol{y}(t_0) &=: \boldsymbol{y}_0, \\ \boldsymbol{y}(t_i) &\simeq \boldsymbol{y}_{i-1} + \boldsymbol{f}(t_{i-1}, \boldsymbol{y}_{i-1})\, h_i + \ldots + \frac{1}{n!} \frac{d^{n-1}\boldsymbol{f}(t_{i-1}, \boldsymbol{y}_{i-1})}{dt^{n-1}}\, h_i^n \quad =: \boldsymbol{y}_i. \end{aligned}$$

Therefore, the problem is reduced to the determination of the Taylor coefficients $\{1/(j+1)!\, d^j \boldsymbol{f}/dt^j\}$. This may be done quite efficiently by means of the automatic differentiation (AD) techniques. Note that the Taylor method has several good features (for details see [15, 19]).

In the Table 1 we present some comparisons on the Henón-Heiles problem using the Taylor method (`TIDES`) and the well established code `dop853` developed by Hairer and Wanner [36]. Both methods are only compared in double and quadruple precision using the Lahey LF 95 compiler because the `dop853` cannot be directly used in multiple precision. Note for low precision the `dop853` code is faster, but when the precision demands are increased the Taylor method is by far the fastest – indeed it is the only reliable method for very high precision. The small figure on the bottom right shows the evolution of the orbit.

		`TIDES` (Taylor)		`dop853`	
CO	`Tol`	CPU	RelErr	CPU	RelErr
`dp`	10^{-10}	0.53E−02	0.201E−10	0.34E−02	0.205E−06
`dp`	10^{-15}	0.12E−01	0.345E−13	0.15E−01	0.113E−11
`qp`	10^{-20}	0.30E+00	0.300E−20	0.30E+01	0.102E−17
`qp`	10^{-25}	0.61E+00	0.165E−26	0.12E+02	0.325E−23
`mpf90`	10^{-32}	0.13E+01	0.782E−29		
`mpf90`	10^{-64}	0.89E+01	0.144E−65		
`mpf90`	10^{-128}	0.74E+02	0.432E−131		

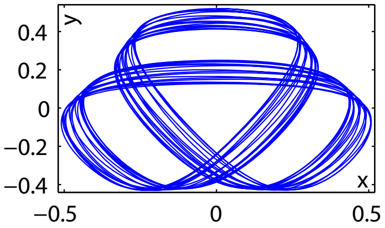


Table 1: CPU time and final error using `dop853` and a Taylor method (`TIDES`) with VSVO formulation for the HH problem using different compiler options (CO): double precision (`dp`) for tolerance levels $10^{-10}, 10^{-15}$, quadruple precision (`qp`) for tolerance levels $10^{-20}, 10^{-25}$ and multiple precision (`mpf90`) for tolerance levels $10^{-32}, 10^{-64}, 10^{-128}$. The figure shows the computed orbit (an orbit on a KAM tori).

It is important to remark that nowadays there are excellent free-software implementations of the Taylor series method, with arbitrary high-precision, for the numerical solution of ODEs and for the automatic determination of the solution of high-order variational equations. The software `TIDES` [1] (Taylor series Integrator for Differential EquationS) is a powerful implementation of this technology (see `http://gme.unizar.es/software/tides` or send an email to `tides@unizar.es`).

3.11 Computing the "skeleton" of periodic orbits

In words of H. Poincaré, periodic orbits form the "skeleton" of a dynamical system and provide much useful information. Therefore, the search for periodic orbits is a quite old problem and numerous numerical and analytical methods have been designed for them. Here we mention just two methods that have been used with high-precision in the literature: the Lindstedt-Poincaré technique [47] and one of the most simple and powerful method to find periodic orbits, namely the systematic search method [18], where one takes advantage of symmetries of the system to find symmetric periodic orbits.

Theorem 1 *Let $o(\boldsymbol{x})$ be an orbit of a flow of an autonomous vector field with a reversal symmetry S. Then, an orbit $o(\boldsymbol{x})$ intersects* $\mathrm{Fix}(S) := \{\, \boldsymbol{x} \,|\, S(\boldsymbol{x}) = \boldsymbol{x} \,\}$ *in precisely two points if and only if the orbit is periodic (and not a fixed point) and symmetric with respect to S.*

The above results were already known by Birkhoff, DeVogelaere and Strömgren (among others) and were used to find symmetric periodic orbits.

The usage of high-precision numerical integrators in the determination of periodic orbits is required in the search of highly unstable periodic orbits. For instance, in Figure 3 we show the computed symmetric periodic orbit for the $7+2$ Ring problem using double and quadruple precision [20]. The $(n+2)$-body Ring problem describes the motion of an infinitesimal particle attracted by the gravitational field of $n+1$ primary bodies, n in the vertices of a regular polygon that is rotating on its own plane about the center with a constant angular velocity. Each point corresponds to the initial conditions of one symmetric periodic orbit, and the grey area corresponds to regions of forbidden motion (delimited by the limit curve). Note that in order to avoid "false" initial conditions it is useful to check if the initial conditions generate a periodic orbit up to a given tolerance level. But in the case of highly unstable periodic orbits we may lose several digits in each period, so that double precision is not enough in many unstable cases, resulting in gaps in the figure.

The Lindstedt-Poincaré method [47] for computing periodic orbits is based on the Lindstedt-Poincaré technique of perturbation theory, Newton's method for solving nonlinear systems and Fourier interpolation. Viswanath [48] uses this algorithm in combination with high-precision libraries to obtain periodic orbits for the Lorenz model at the classical Saltzman's parameter values. This procedure permits one to compute, to high accuracy (more than 100 digits of precision), highly unstable periodic orbits (for instance the orbit with symbolic dynamics $ABA^2B^2 \cdots A^{15}B^{15}$ has a leading characteristic multiplier 3.06×10^{59}, which means that we can expect that at each period we lose around 59 digits of precision). For these reasons, high-precision arithmetic plays a fundamental role in the study of the fractal properties of the Lorenz attractor.

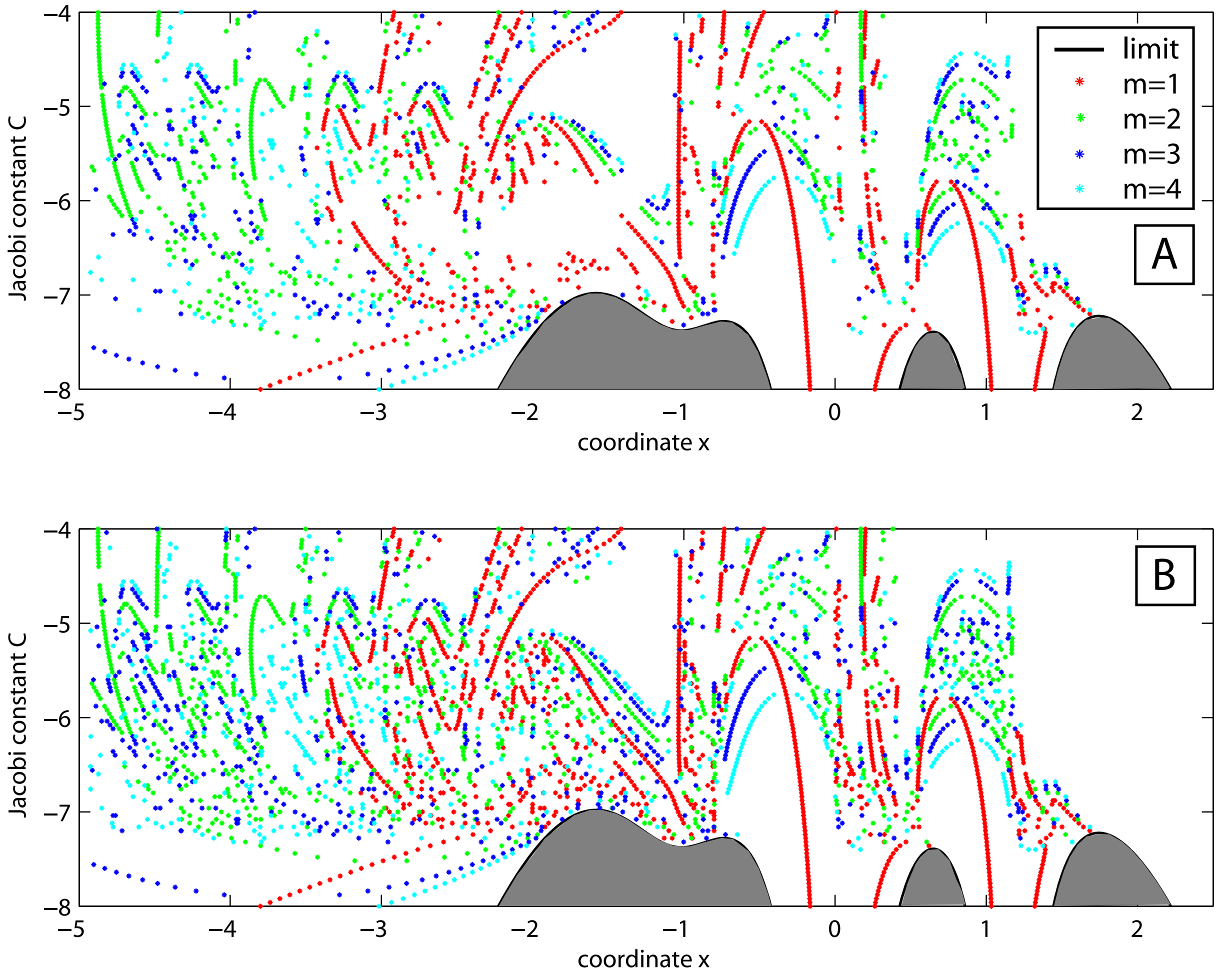


Figure 3: Symmetric periodic orbits (m denotes the multiplicity of the periodic orbit) in the most chaotic zone of the $7+2$ Ring problem using double (A) and quadruple (B) precision.

3.12 Divergent asymptotic series and homoclinic phenomena

One interesting phenomena in dynamical systems is the study of the splitting of separatrices of area preserving maps. Numerical difficulties arise because this phenomena can exhibit exponentially small splitting [34]. For instance, the most common paradigmatic example is the standard map defined by $(x, y) \mapsto (\hat{x}, \hat{y})$ where

$$\hat{y} = y + \varepsilon \sin x, \quad \hat{x} = x + \hat{y}$$

and ε is a small positive constant. An asymptotic formula for the angle between the stable and the unstable separatrices at the primary homoclinic point was given by Lazutkin [42]:

$$\alpha = \frac{\pi}{\varepsilon} \mathrm{e}^{-\frac{\pi^2}{\sqrt{\varepsilon}}} \big(1118.8277059409\ldots + \mathcal{O}(\sqrt{\varepsilon})\big).$$

As a result, the separatrices are transversal, but the angle between them is exponentially small compared to ε. This leads to severe problems in numerical simulations. Gelfreich and Simó [34] use a homoclinic invariant ω that gives the area of a parallelogram defined by two vectors tangent to the stable and the unstable manifolds at the homoclinic point. While ω in the standard map can be represented by an asymptotic series, one question is what happens when we use several generalizations of the standard map. In [34], the authors employed high-precision computation of the homoclinic invariant and consecutive extraction of coefficients of an asymptotic expansion, in order to obtain a numerical evidence that various different types of asymptotic expansions arise in this class of problems. These results are unachievable using standard double precision; in some numerical simulations 1000-digit precision was required.

3.13 Detecting SNA

In the study of dynamics of dissipative systems the detection of the attractors is quite important, because they are the visible invariant sets of the dynamics of the problem. An attractor is defined as *strange* if it is not a piecewise smooth manifold and *chaotic* if any orbit on it exhibits sensitive dependence on initial conditions. All the first examples of strange attractors in the literature where strange chaotic attractors, but soon some strange nonchaotic attractors were identified. Several authors suggested that in the transition to chaos in quasiperiodically forced dissipative systems, in particular in the so called fractalization route in which a smooth torus seems to fractalize, strange nonchaotic attractors appear. In [37], Haro and Simó showed that in truth these attractors are nonstrange. These authors found that multiprecision arithmetic with more than 30 digits was needed to reliably study this behavior at very small scales.

3.14 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 [46] (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 [13].

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 [6]. 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* [2]. 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.,

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

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

$$C_{1024} = 0.63047350337438679612204019271087890435458707871273234\ldots,$$

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 [6].

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:

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

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 [6]. In a follow-on study [8], 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 [26]—that there are linear relations in each of the rows of this array (considered as a doubly-infinite rectangular matrix), e.g.,

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

In yet a more recent study, co-authored with physicists David Broadhurst and Larry Glasser [5], 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

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

where ${}_pF_q$ denotes the *generalized hypergeometric* function [2]. 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 $\mathbf{K}$ denotes the elliptic integral of the first kind [2]. 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

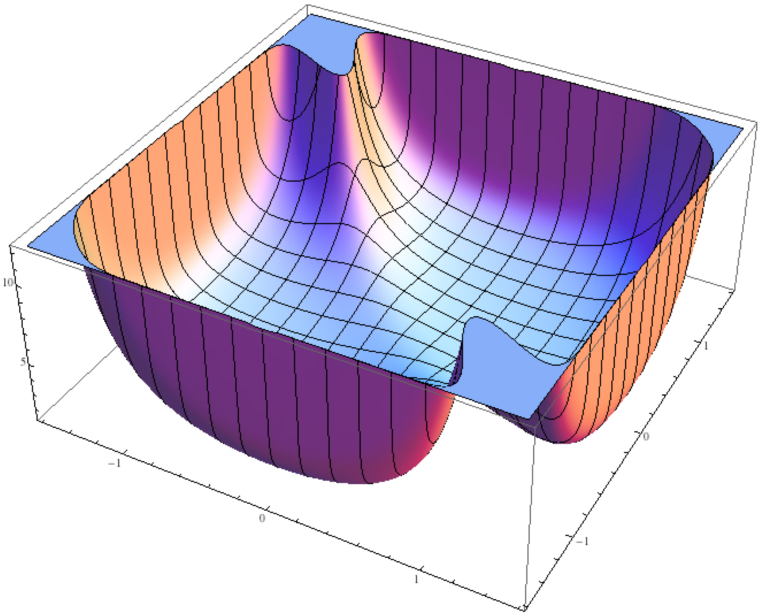


Figure 4: Plot of $c_{5,0}$ integrand function.

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 A Discrete Dynamical System: Discovery and Partial Proof

Let $R_A(x) := 2\,P_A(x) - x, R_B(x) := 2\,P_B(x) - x$, where P_A, P_B denote the Euclidean metric projections, or nearest point maps, on closed sets A and B. In our setting, the *Lions-Mercier* (LM) iteration (which can be given many other names [22] such as *Douglas-Rachford* or *Feinup*'s algorithm) is the procedure: *reflect, reflect and average*:

$$x \mapsto T(x) := \frac{x + R_A\left(R_B(x)\right)}{2}. \tag{5}$$

Note that a fixed point z of T produces precisely a point w such that $w := P_B(z) = P_A\left(R_B(z)\right)$ is an element of $A \cap B$. Moreover, if one shows that $\|T(z_n) - z_n\| \to 0$ (known as *asymptotic regularity* of $z_{n+1} := T(z_n)$) then every cluster point of the corresponding orbit produces a fixed point z.

The consequent theory of this and related iterations is well understood in the convex case [21, 22, 23]. In the non-convex case the iteration, also called "divide-and-concur" [35], has been very successful in a variety of reconstruction problems (such as protein folding, 3SAT, spin glasses, giant Sudoku puzzles, etc.). As discovered very recently, "divide and concur" works better than theory can explain [31, 35]. Even the most special case is subtle and illustrative of general phase reconstruction problems and the like.

Let $P_A(x)$ and $R_A(x) := 2\,P_A(x) - x$ denote respectively the *projector* and *reflector* on a set A as shown in Figure 5 where A is the boundary of the shaded ellipse. Then "divide and concur" is the natural geometric iteration "reflect-reflect-average":

$$x_{n+1} \Longrightarrow \frac{x_n + R_A\left(R_B(x_n)\right)}{2}. \tag{6}$$

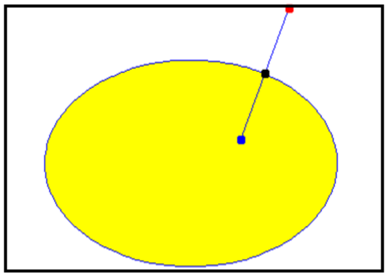


Figure 5: Reflector (interior) and Projector (boundary) of a point external to an ellipse.

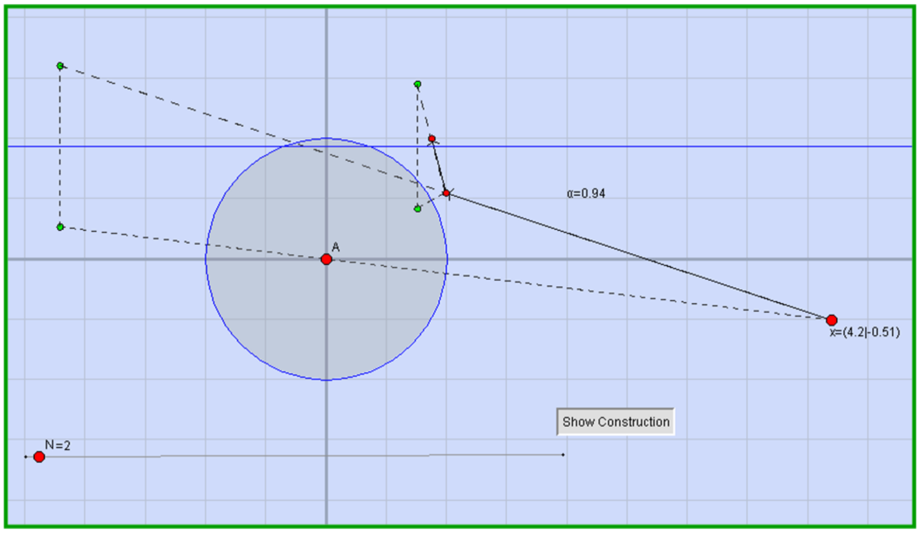


Figure 6: The first three iterates of (7) in *Cinderella*.

Consider the simplest case of a line A of height α (all lines may be assumed horizontal) and the unit circle B.With $z_n := (x_n, y_n)$ we obtain the explicit iteration

$$x_{n+1} := \cos\theta_n, \quad y_{n+1} := y_n + \alpha - \sin\theta_n, \quad (\theta_n := \arg z_n). \tag{7}$$

For the infeasible case with $\alpha > 1$ it is easy to see the iterates go to infinity vertically. For the tangent $\alpha = 1$ we provably converge to an infeasible point. For $0 < \alpha < 1$, the pictures are lovely but proofs escape the authors. Spiraling is ubiquitous in this case. Two representative *Maple* pictures follow:

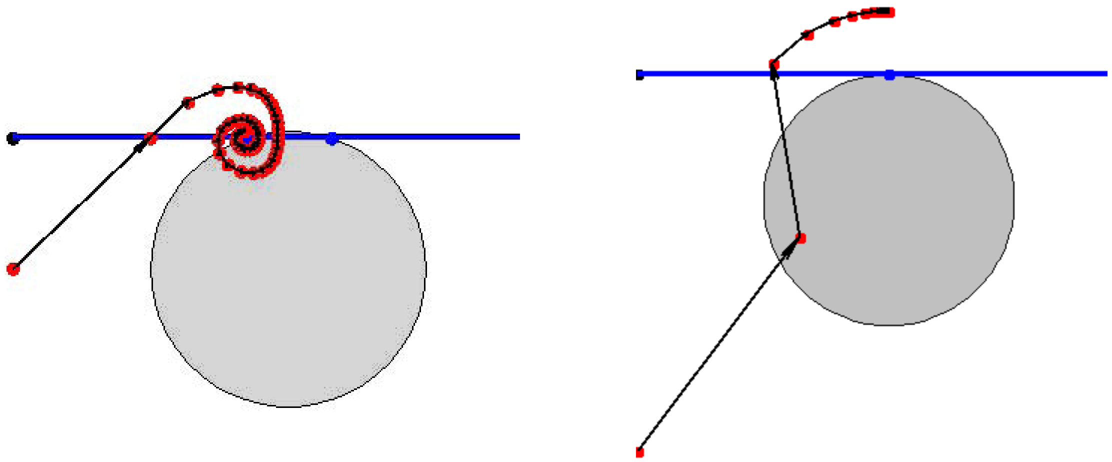


Figure 7: The behavior of (7) for $\alpha = 0.95$ (L) and $\alpha = 1$ (R).

For $\alpha = 0$ we can prove convergence to one of the two points in $A \cap B$ if and only if we do not start on the vertical axis, where we provably have *chaos*. The iteration is illustrated in Figure 6 starting at $(4.2, -0.51)$ with $\alpha = 0.94$. Let us sketch how the interactive geometry *Cinderella* (available at `http://www.cinderella.de`) leads one both to discovery and a proof in this equatorial case. Interactive applets are easily made; the next two figures are based on material available online at, respectively:

A1. `http://users.cs.dal.ca/∼jborwein/reflection.html`

A2. `http://users.cs.dal.ca/∼jborwein/expansion.html`

Figure 8 illustrates the applet **A1** at work: by dragging the trajectory (with $N = 28$) one quickly discovers that

(i) as long as the iterate is outside the unit circle the next point is *always* closer to the origin;

(ii) once inside the circle the iterate *never* leaves;

(iii) the angle now *oscillates* to zero and the trajectory hence converges to $(1, 0)$.

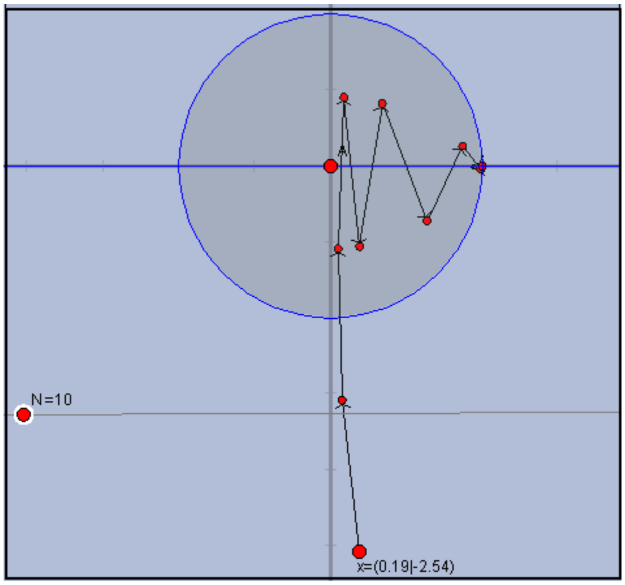


Figure 8: Discovery of the proof with $\alpha = 0$.

All of this is quite easily made algebraic in the language of (7).

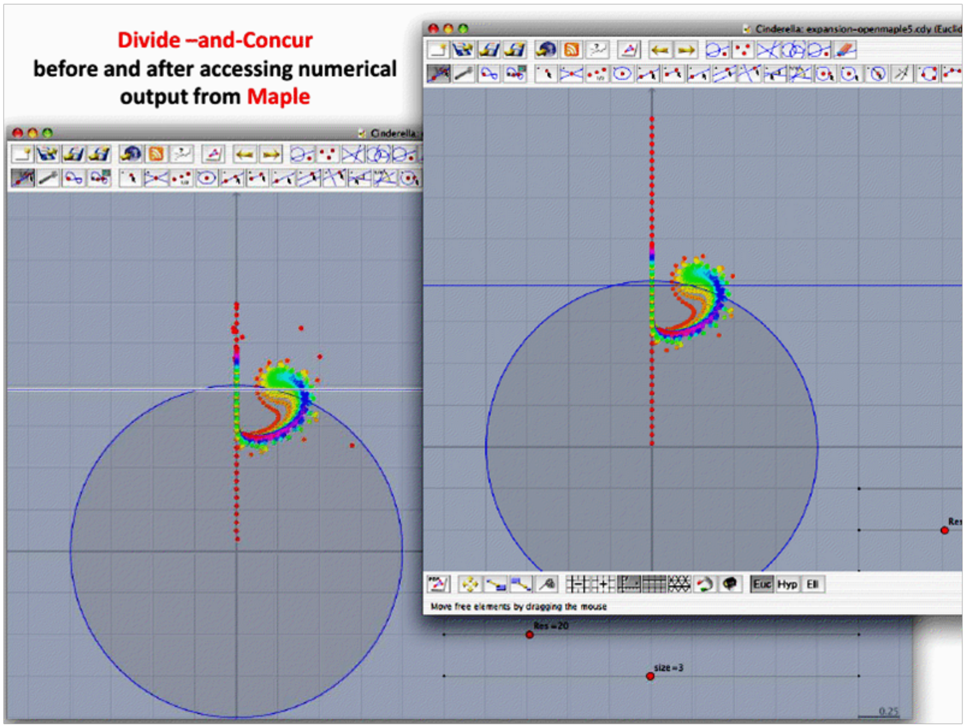
Figure 9 illustrates the applet **A2**, which takes up to $10,000$ starting points in the rectangle $\{(x,y)\colon 0 \leq x \leq 1, |y-\alpha|| \leq 1\}$ colored by distance from the vertical axis with red on the axis and violet at $x = 1$, and produces the first hundred iterations in gestalt. Thus we see clearly, but cannot yet rigorously prove, that all points not on the y-axis are swept into the feasible point $(\sqrt{1-\alpha^2}, \alpha)$.

This graphic, namely Figure 9, demonstrates in clear graphical terms the numerical difficulty in these examples. Comparing the left-hand side (based solely on computations done in *Cinderella* using ordinary 64-bit IEEE arithmetic) with the right-hand side (based on data computing using *Maple*, employing higher-precision arithmetic), it is clear that *Cinderella*'s double precision (14 digits) is inadequate. Indeed, the limitations of ordinary 64-bit IEEE arithmetic (approximately 15 digits) loom as a major obstacle in further explorations of this type – the usage of higher-precision arithmetic will be mandatory.

Littlewood once wrote:

> *"A heavy warning used to be given [by lecturers] that pictures are not rigorous; this has never had its bluff called and has permanently frightened its victims into playing for safety. Some pictures, of course, are not rigorous, but I should say most are (and I use them whenever possible myself)."*—J. E. Littlewood, (1885-1977)[1]

[1] From p. 53 of the 1953 edition of Littlewood's *Miscellany* and so said long before the current fine graphic, geometric, and other visualization tools were available.



Divide –and-Concur
before and after accessing numerical
output from Maple

Move free elements by dragging the mouse

Figure 9: Gestalt of 400 third steps in *Cinderella* without (L) and with *Maple* data (R).

In a similar vein, we find it hard to be persuaded that the applet **A2** does not constitute a proof of sorts of what it displays in Figure 10.

We have also considered the analogous differential equation, since asymptotic techniques for such differential equations are better developed. We decided that

$$x'(t) = \frac{x(t)}{r(t)} - x(t), \quad y'(t) = \alpha - \frac{y(t)}{r(t)},$$

where $r(t) := \sqrt{x(t)^2 + y(t)^2}$, was a reasonable counterpart to the Cartesian formulation of (7)—we have replaced the difference $x_{n+1} - x_n$ by $x'(t)$, etc.—as shown in Figure 11. This led to a proof of local convergence for $0 < \alpha < 1$ [27] and of the spiraling as seen in the pictures. But we have no global result in this case and now we have a whole other class of discoveries without explanation.

We should add that this is an ideal problem to introduce early undergraduates to research, since it involves only school geometry notions and has many accessible extensions in two or three dimensions. Much can be discovered and most of it will be both original and unproven. Consider, for instance, what happens when B is a line segment or a finite set rather than a line or when A is a more general conic section. Corresponding algorithms, like "project-project-average," are representative of techniques used to correct

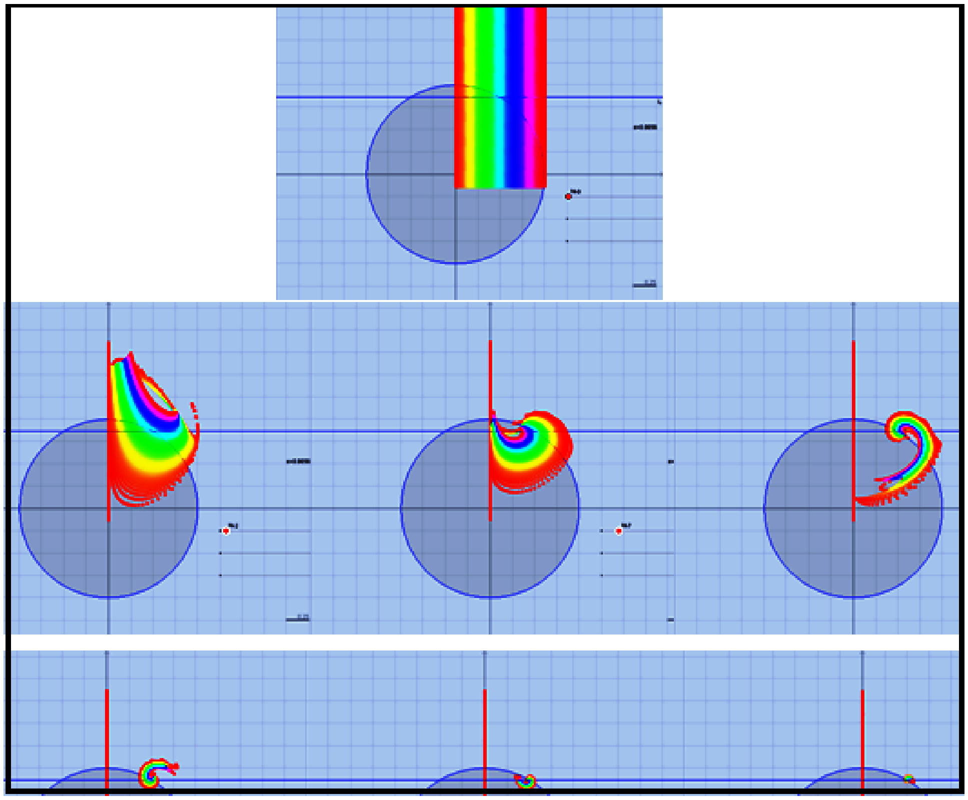


Figure 10: Snapshots of 10,000 points after 0, 2, 7, 13, 16, 21, and 27 steps in *Cinderella*.

the Hubble telescope's early optical aberration problems.

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

References

[1] A. Abad, R. Barrio, F. Blesa and M. Rodriguez, "TIDES: a Taylor series Integrator for Differential EquationS," preprint (2009). `http:gme.unizar.es/software/tides`.

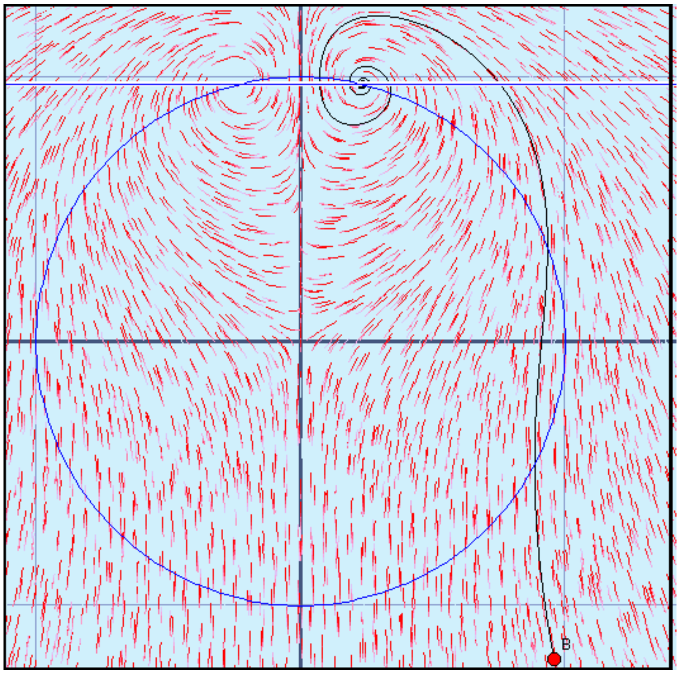


Figure 11: ODE solution and vector field for (8) with $\alpha = 0.97$ in *Cinderella*.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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`.

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

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

[15] R. Barrio, "Performance of the Taylor series method for ODEs/DAEs," *Appl. Math. Comput.*, vol. 163 (2005), 525–545.

[16] R. Barrio, B. Melendo and S. Serrano, "Generation and evaluation of orthogonal polynomials in discrete Sobolev spaces I. Algorithms," *J. Comput. Appl. Math.*, vol. 181 (2005), 280–298.

[17] R. Barrio and S. Serrano, "Generation and evaluation of orthogonal polynomials in discrete Sobolev spaces II. Numerical stability," *J. Comput. Appl. Math.*, vol. 181 (2005), 299–320.

[18] R. Barrio and F. Blesa, "Systematic search of symmetric periodic orbits in 2DOF Hamiltonian systems," *Chaos, Solitons and Fractals*, vol. 41 (2009), 560-582.

[19] R. Barrio, F. Blesa, M. Lara, "VSVO formulation of the Taylor method for the numerical solution of ODEs," *Comput. Math. Appl.*, vol. 50 (2005), 93–111.

[20] R. Barrio, F. Blesa and S. Serrano, "Qualitative analysis of the $(n + 1)$-body ring problem," *Chaos Solitons Fractals*, vol. 36 (2008), 1067–1088.

[21] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Finding best approximation pairs relative to two closed convex sets in Hilbert spaces." *J. Approx. Theory*, **127** (2004), 178–192.

[22] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization," *J. Opt. Soc. Amer. A* **19** (2002), 1334–1345.

[23] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "A strongly convergent reflection method for finding the projection onto the intersection of two closed convex sets in a Hilbert space," *J. Approx. Theory*, **141** (2006), 63–69.

[24] 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`.

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

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

[27] J. M. Borwein and B. Sims, "Convergence of non-convex Douglas-Ratchford iterations," preprint, 2009.

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

[29] 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`.

[30] 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`.

[31] V. Elser, I. Rankenburg, and P. Thibault, "Searching with iterated maps", *Proceedings of the National Academy of Sciences* **104** (2007), 418–423.

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

[33] 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.

[34] V. Gelfreich and C. Simó, "High-precision computations of divergent asymptotic series and homoclinic phenomena," *Discrete Contin. Dyn. Syst. Ser. B*, vol. 10 (2008), 511–536

[35] S. Gravel, and V. Elser, "Divide and concur: A general approach to constraint satisfaction," preprint, 2008, `http://arxiv.org/abs/0801.0222v1`.

[36] E. Hairer, S. Nørsett and G. Wanner, "Solving ordinary differential equations. I. Nonstiff problems." Second edition. Springer Series in Computational Mathematics, 8. Springer-Verlag, Berlin, 1993.

[37] A. Haro and C. Simó, "To be or not to be a SNA: That is the question," Preprint 2005-17 of the Barcelona UB-UPC Dynamical Systems Group (2005).

[38] 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.

[39] W. Hayes, "Is the outer Solar System Chaotic?," *Nature Physics* vol. 3 (2007), 689-691.

[40] 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.

[41] 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.

[42] V. F. Lazutkin, "Splitting of separatrices for the Chirikov standard map," *J. Math. Sci.* vol. 128 (2005), 2687-2705.

[43] Jack Dongarra, "LAPACK – Linear Algebra Package," `http://www.netlib.org/lapack`.

[44] Jack Dongarra, "LINPACK," `http://www.netlib.org/linpack`.

[45] 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`.

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

[47] D. Viswanath, "The Lindstedt-Poincaré technique as an algorithm for computing periodic orbits," *PSIAM Rev* 43 (2001), 478–495.

[48] D. Viswanath, “The fractal property of the Lorenz attractor,” *Phys. D* 190 (2004), 115–128.

[49] 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.

[50] 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.