High-precision methods for zeta functions
Part 1: functions, formulas

Fredrik Johansson
INRIA Bordeaux

UNCG Summer School in Computational Number Theory
May 18–22, 2015

Why high precision?

- Identify special values, e.g. $\zeta(20150518)/\pi^{20150518}$
- Investigate behavior for large input / near singularities (catastrophic cancellation)
- Guaranteeing correct results (error bounds may blow up)
- Computational complexity

Zeta functions and generalizations

Riemann zeta function: $\zeta(s) = \sum_{k=1}^{\infty} \frac{1}{k^s}$

Dirichlet L-functions: $L(s, \chi) = \sum_{k=1}^{\infty} \frac{\chi(k)}{k^s}$

Hurwitz zeta function: $\zeta(s, a) = \sum_{k=0}^{\infty} \frac{1}{(k+a)^s}$

Polylogarithm: $\mathrm{Li}_s(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^s}$

Lerch transcendent: $\Phi(z, s, a) = \sum_{k=0}^{\infty} \frac{z^k}{(k+a)^s}$

General L-functions: $L(s) = \sum_{n=1}^{\infty} \frac{a_n}{n^s}$

Connection formulas

Hurwitz zeta function (with rational parameter) ⇔ Dirichlet L-functions:

$$\zeta(s, n/k) = \frac{k^s}{\varphi(k)} \sum_{\chi \bmod k} \overline{\chi}(n) L(s, \chi)$$

$$L(s, \chi) = \frac{1}{k^s} \sum_{n=1}^{k} \chi(n)\, \zeta\left(s, \frac{n}{k}\right)$$

Hurwitz zeta function ⇔ polylogarithms:

$$\mathrm{Li}_s(z) = \frac{\Gamma(1-s)}{(2\pi)^{1-s}} \left[i^{1-s}\, \zeta\left(1-s,\ \frac{1}{2} + \frac{\ln(-z)}{2\pi i}\right) + i^{s-1}\, \zeta\left(1-s,\ \frac{1}{2} - \frac{\ln(-z)}{2\pi i}\right) \right]$$

$$\mathrm{Li}_s(z) + (-1)^s\, \mathrm{Li}_s(1/z) = \frac{(2\pi i)^s}{\Gamma(s)}\, \zeta\left(1-s,\ \frac{1}{2} + \frac{\ln(-z)}{2\pi i}\right)$$

[modulo poles, branch cuts]

Supporting functions

Elementary functions: $\exp, \log, \sin, \cos, \text{atan}$

Bernoulli numbers B_n, integer zeta values $\zeta(2), \zeta(3), \ldots$

Finite power sums, e.g. $\sum_{k=1}^{N} k^{-s}$

Gamma function: $\Gamma(s)$ and $\log \Gamma(s) \neq \log(\Gamma(s))$

Incomplete gamma function: $\Gamma(s, z) = \displaystyle\int_z^{\infty} t^{z-1} e^{-t} dt$

Generalized hypergeometric functions:

$$ {}_pF_q(a_1, \ldots, a_p; b_1, \ldots, b_q; z) = \sum_{n=0}^{\infty} \frac{(a_1)_n \ldots (a_p)_n}{(b_1)_n \ldots (b_q)_n} \frac{z^n}{n!} $$

D-finite (holonomic) functions: Bessel functions, hypergeometric functions, Meijer G-functions, ...

Software

Open source:

- Sage
- Pari/GP
- Dokchitser's L-function calculator
- mpmath
- Arb

Non-open-source:

- Mathematica
- Magma

Various special-purpose programs...

Representing numbers in Arb

arb type (real mid-rad interval, "ball"):

$$[\underbrace{3.14159265358979323846264338328}_{\text{arbitrary-precision floating-point}} \pm \underbrace{8.65 \cdot 10^{-31}}_{\text{30-bit precision}}]$$

acb type (complex rectangular "ball"):

$$[1.414213562 \pm 3.74 \cdot 10^{-10}] + [1.732050808 \pm 4.32 \cdot 10^{-10}]i$$

Goal

- $\{\text{cost of interval arithmetic}\} = (1+\varepsilon) \cdot \{\text{cost of floating-point}\}$
- $\{\text{effort for error analysis}\} = \varepsilon \cdot \{\text{effort with floating-point}\}$

Computing functions in Arb

Example (simplified):

$$f(x) = e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Input: $X = [m \pm r]$ with $x \in X$

$$f(X) \subseteq [A \pm E], \qquad \underbrace{A = \sum_{k=0}^{N-1} \frac{X^k}{k!}}_{\text{Using ball arithmetic}}, \quad \underbrace{\left| \sum_{k=N}^{\infty} \frac{X^k}{k!} \right| \leq E}_{\text{Upper bound}}$$

Better error propagation:

$$f(X) \subseteq [A \pm (E_1 + E_2)], \quad A = \sum_{k=0}^{N-1} \frac{m^k}{k!}, \quad \left| \frac{m^k}{k!} \right| \leq E_1, \quad \sup_{t \in X} |e^m - e^t| \leq E_2$$

Note $\exp(-100) \approx 10^{-44}$ but $(-100)^{100}/100! \approx 10^{42} \approx \exp(100)$

Toolchain

Fast arithmetic

$\Downarrow$

Evaluation of sums/recurrences

$\Downarrow$

Evaluation of supporting functions

$\Downarrow$

Evaluation of zeta functions

Fast arithmetic ("reduce everything to multiplication")

Bit complexity for multiplying two n-bit integers: $M_{\mathbb{Z}}(n)$

Arithmetic complexity for multiplying two degree-n polynomials over a ring R: $M_{R[x]}(n)$

Classical multiplication: $M(n) = O(n^2)$
Karatsuba multiplication: $M(n) = O(n^{1.585})$
FFT multiplication: $M(n) = M(n \log n \log \log n) = O\tilde{~}(n)$

Bit complexity for multiplying degree-n polynomials over $R = \mathbb{Z}$ with p-bit coefficients: $O\tilde{~}(np)$

How high can we go?

- In the critical strip: $\zeta(1/2 + ti)$ for $t \approx 10^{30}$ – recent work by Hiary and Bober. Complexity: $\approx O(t^{1/3})$
- Precise values of $\zeta^{(n)}(s)$ with s small: to about $p = 1\,000\,000$ bits, or up to $n = 100\,000$ without too much effort [more about this later]. Complexity: $\approx \tilde{O}(t + p^2 + n^2)$
- Very special values (like $\zeta(3)$ or $\gamma = 0.577\ldots$) to $1\,000\,000\,000$ bits or more. Note: Record for computing γ is $119\,377\,958\,182$ digits, set by Alexander Yee in 2013. Complexity: $\tilde{O}(p)$

Effective formulas for computing zeta functions

- Integral representations (numerical integration)
- Infinite series (directly or using convergence acceleration)
- Euler-Maclaurin summation
- The Riemann-Siegel formula
- The approximate functional equation

Integral representations

Contour integrals:

$$\zeta(s) = \frac{\Gamma(1-s)}{2\pi i} \oint \frac{t^{s-1}}{e^{-t}-1} dt$$

$$\frac{1}{\Gamma(s)} = \frac{i}{2\pi} \oint (-t)^{-s} e^{-t}\, dt$$

On the real axis (valid at least for $\Re(a) > 0$):

$$\Phi(z,s,a) = \sum_{k=0}^{\infty} \frac{z^k}{(k+a)^s} = \frac{1}{2a^s} + \frac{\log^{s-1}(1/z)}{z^a} \Gamma(1-s, a\log(1/z))$$
$$+ \frac{2}{a^{s-1}} \int_0^\infty \frac{\sin(s \operatorname{atan}(t) - ta\log(z))}{(1+t^2)^{s/2}(e^{2\pi a t}-1)}\, dt$$

High-precision integration algorithms

- Trapezoidal rule (on closed contours)
- Gaussian quadrature
- Clenshaw-Curtis quadrature (Chebyshev series)
- Taylor series methods
- Double exponential

Double exponential integration

For an analytic integrand on $(-1, +1)$, the change of variables $x = \tanh(\frac{1}{2}\pi \sinh t)$ gives an integral on $(-\infty, +\infty)$ that is extremely well approximated by a simple step sum.

$$\int_{-1}^{1} f(x)\, dx = \int_{-\infty}^{\infty} g(x)\, dx \approx \sum_{k=-\infty}^{\infty} w_k f(x_k)$$

$$x_k = \tanh(\tfrac{1}{2}\pi \sinh kh), \quad w_k = \frac{\frac{1}{2} h\pi \cosh kh}{\cosh^2(\frac{1}{2}\pi \sinh kh)}$$

Halving the step size h typically doubles the number of correct digits! This still works when $f(x)$ has (sufficiently nice) singularities at the endpoints.

[D. H. Bailey and J. M. Borwein, *Effective Error Bounds in Euler-Maclaurin-Based Quadrature Schemes*, 2008]

[P. Molin, *Intégration numérique et calculs de fonctions L*, PhD thesis, 2010]

Direct summation

$$\zeta(s) \approx \sum_{k=1}^{n} \frac{1}{k^s} \approx \prod_{p \le n} \frac{1}{1 - p^{-s}}$$

This is only good if $\text{Re}(s) \sim prec^{1-\varepsilon}$. In fact, this is a good way to compute the Bernoulli numbers B_n, for n large.

For polylogarithm, Lerch transcendent:

$$\text{Li}_s(z) \approx \sum_{k=1}^{n} \frac{z^k}{k^s}$$

works well when, say, $|z| < 0.9$

Methods based on convergence acceleration

Some methods to approximate

$$\sum_{k=1}^{\infty} f(k)$$

from just a few terms when $f(k)$ decreases slowly:

- Richardson extrapolation
- Alternating series convergence acceleration
  - Shanks transformation (Wynn's ε-algorithm)
  - Euler transformation
  - Chebyshev polynomial algorithm
- Euler-Maclaurin summation

Try the `nsum` function in `mpmath` on your favorite slowly converging (or diverging) series!

Alternating series convergence acceleration

$$\eta(s) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^s} = (1 - 2^{1-s})\zeta(s)$$

$$\eta(s) = -\frac{1}{d_n} \sum_{k=0}^{n-1} \frac{(-1)^k (d_k - d_n)}{(k+1)^s} + \gamma_n(s), \quad d_k = n \sum_{i=0}^{k} \frac{(n+i-1)! 4^i}{(n-i)!(2i)!}$$

For $\mathrm{Re}(s) \geq \frac{1}{2}$,

$$|\gamma_n| \leq \frac{2}{(3+\sqrt{8})^n} \frac{1}{|(1-2^{1-s})\Gamma(s)|}$$

[Cohen, Rodriguez Villegas, Zagier, *Convergence acceleration of alternating series*, 2000]
[P. Borwein, *An efficient algorithm for the Riemann zeta function*, 2000]

The Euler-Maclaurin formula

For any sufficiently differentiable function f,

$$\sum_{k=N}^{U} f(k) = I + T + R$$

$$I = \int_N^U f(t)\,dt$$

$$T = \frac{1}{2}\left(f(N) + f(U)\right) + \sum_{k=1}^{M} \frac{B_{2k}}{(2k)!}\left(f^{(2k-1)}(U) - f^{(2k-1)}(N)\right)$$

$$R = -\int_N^U \frac{\tilde{B}_{2M}(t)}{(2M)!} f^{(2M)}(t)\,dt$$

Computing $\zeta(s, a)$ using Euler-Maclaurin

$$\zeta(s,a) = \underbrace{\sum_{k=0}^{N-1} f(k)}_{S} + \underbrace{\sum_{k=N}^{\infty} f(k)}_{I+T+R}, \quad f(k) = \frac{1}{(a+k)^s}$$

$$S = \sum_{k=0}^{N-1} \frac{1}{(a+k)^s}$$

$$I = \int_N^{\infty} \frac{1}{(a+t)^s} dt = \frac{(a+N)^{1-s}}{s-1}$$

$$T = \frac{1}{(a+N)^s} \left(\frac{1}{2} + \sum_{k=1}^{M} \frac{B_{2k}}{(2k)!} \frac{(s)_{2k-1}}{(a+N)^{2k-1}} \right)$$

$$R = -\int_N^{\infty} \frac{\tilde{B}_{2M}(t)}{(2M)!} \frac{(s)_{2M}}{(a+t)^{s+2M}} dt$$

[This also provides the analytic continuation.]

The Riemann-Siegel formula

Other methods essentially need $O(t)$ operations at height t

$$\mathcal{R}(s) = \sum_{n=1}^{\lfloor a \rfloor} \frac{1}{n^s} + (\ldots) \left[\sum_{k=0}^{K} \frac{(\ldots)}{a^k} + RS_K \right]$$

$$a = \sqrt{t/(2\pi)}$$

[Arias de Reyna, *High precision computation of Riemann's zeta function by the Riemann-Siegel formula, I*, 2011]

$$|RS_K| \leq c_1 \frac{\Gamma((K+1)/2)}{((10/11)a)^{K+1}}$$

At height t, we can compute about $0.057t$ digits rigorously

Implementation (semi-heuristic) in `mpmath` by Juan Arias de Reyna (plus code to locate zeros)

Saving time in the power sum

The terms $f(k) = k^{-s}$ in $\sum_{k=1}^{N} f(k)$ are completely multiplicative, i.e. $f(k_1 k_2) = f(k_1) f(k_2)$. Only need values at prime k.

Can also extract multiples of small primes. Extracting powers of two gives a polynomial in $f(2)$, e.g. for $\sum_{k=1}^{10} f(k) =$

$$\begin{aligned} & [f(1) + f(3) + f(5) + f(7) + f(9)] \\ +f(2) & [f(1) + f(3) + f(5)] \\ +f(4) & [f(1)] \\ +f(8) & [f(1)]. \end{aligned}$$

Need $\pi(N) \approx N / \log N$ evaluations of $f(k)$ and $N/2$ multiplications. Must store about $N/6$ function values.

Fast computation for large t (low precision)

[Hiary, *Fast methods to compute the Riemann zeta function*, 2011]: $\zeta(1/2 + it)$ can be evaluated to within $t^{-\lambda}$ for any fixed λ using $t^{4/13+o_\lambda(1)} \approx t^{0.307}$ arithmetic operations

Basic idea: break zeta sum into blocks of a well-chosen size K, and use Taylor expansion

$$\sum_{k=0}^{K-1} \exp(it \log(n_0 + k)) = \exp(it \log(n_0)) \sum_{k=0}^{K-1} \exp(it \log(1 + k/n_0))$$

$$= \exp(it \log(n_0)) \sum_{k=0}^{K-1} \exp(itk/n_0 - itk^2/(2n_0^2) + \ldots)$$

Fast way to compute such exponential sums when the expansion is truncated to a quadratic or cubic polynomial.

The approximate functional equation

$$\pi^{-s/2}\zeta(s)\Gamma(s/2) = \pi^{-s/2}S_1 + \pi^{(s-1)/2}S_2 + \frac{1}{s(s-1)}$$

$$S_1 = \sum_{n=1}^{\infty} n^{-s}\Gamma(s/2, \pi n^2), \quad S_2 = \sum_{n=1}^{\infty} n^{s-1}\Gamma((1-s)/2, \pi n^2)$$

$$\Gamma(s,z) = \int_z^{\infty} t^{s-1}e^{-t}dt \quad \sim \quad z^{s-1}e^{-z}, \quad z \to +\infty$$

Only need $O(p^{1/2})$ terms for p-bit precision (but the terms are complicated)

The incomplete gamma function

Convergent series:

$$\Gamma(s,z) = \Gamma(s) - \frac{z^s}{s} {}_1F_1(s, s+1, -z) = \Gamma(s) - \frac{z^s e^{-z}}{s} {}_1F_1(1, s+1, z)$$

Continued fraction:

$$\Gamma(s,z) = \cfrac{z^s e^{-z}}{z + \cfrac{1-s}{1 + \cfrac{1}{z + \cfrac{2-s}{1 + \ddots}}}}$$

Asymptotic expansion:

$$\Gamma(s,z) \sim z^{s-1} e^{-z} \left(1 + \frac{s-1}{z} + \ldots\right)$$

More general L-functions

[T. Dokchitser, *Computing special values of motivic L-functions*, 2004]

$$L(s) = \sum_{n=1}^{\infty} \frac{a_n}{n^s}$$

Given: Dirichlet coefficients $a_n \in \mathbb{C}$, weight $w \geq 0$, sign $\epsilon = \pm 1$, exponential factor $A > 0$, dimension d, constants $\lambda_1, \ldots, \lambda_d \in \mathbb{C}$ such that

$$\gamma(s) = \prod_{i=1}^{d} \Gamma\left(\frac{s+\lambda_i}{2}\right), \quad L^*(s) = A^s \gamma(s) L(s), \quad L^*(s) = \epsilon L^*(w-s)$$

The singularities $L^*(s)$ are assumed to be a finite list of simple poles p_j with residues r_j.

Computational formula

Define $\phi(t)$ as the inverse Mellin transform of $\gamma(s)$,

$$\gamma(s) = \int_0^\infty \phi(t) t^s \frac{dt}{t}$$

Then $\phi(t)$ decays exponentially, and

$$L^*(s) = \int_0^\infty \Theta(t) t^s \frac{dt}{t}$$

where

$$\Theta(t) = \sum_{n=1}^\infty a_n \phi\left(\frac{nt}{A}\right)$$

We have the functional equation

$$\Theta(1/t) = \epsilon t^w \Theta(t) - \sum_j r_j t^{p_j}$$

Computational formula

$$L^*(s) = \int_1^\infty \Theta(t) t^s \frac{dt}{t} + \int_0^1 \Theta(t) t^s \frac{dt}{t}$$

$$L^*(s) = \int_1^\infty \Theta(t) t^s \frac{dt}{t} + \epsilon \int_1^\infty \Theta(t) t^{w-s} \frac{dt}{t} + \sum_j \frac{r_j}{p_j - s}$$

This gives

$$L^*(s) = \sum_{n=1}^\infty a_n G_s\left(\frac{n}{A}\right) + \epsilon \sum_{n=1}^\infty a_n G_{w-s}\left(\frac{n}{A}\right) + \sum_j \frac{r_j}{p_j - s}$$

where

$$G_s(t) = t^{-s} \int_t^\infty \phi(t) x^s \frac{dx}{x}$$

Dirichlet L-functions

If χ is a primitive Dirichlet character mod N, then the data for $L(s,\chi) = \sum_{k=1}^{\infty} \frac{\chi(k)}{k^s}$ consists of:

$$w = d = 1$$

$$\lambda_1 = 0 \quad (\chi(-1) = 1)$$

$$\lambda_1 = 1 \quad (\chi(-1) = -1)$$

$$A = \sqrt{N/\pi}, \quad |\epsilon| = 1$$

[How does this algorithm compare to Euler-Maclaurin summation evaluation of $\zeta(s,z)$ for large N?]

The function $G_s(z)$

Numerical evaluation of $G_s(z)$ in general is analogous to $\Gamma(s, z)$ (see Dokchitser's paper for formulas):

- Convergent series
- Continued fraction
- Asymptotic series

Effective error bounds for continued fractions and asymptotic series are an open problem (except in special cases).

The function $G_s(z)$ is a special case of the Meijer G-function

$$G_{p,q}^{m,n}\left(\begin{matrix} a_1, \ldots, a_n; a_{n+1} \ldots a_p \\ b_1, \ldots, b_m; b_{m+1} \ldots b_q \end{matrix} \middle| \; z; r\right)$$

$$= \frac{1}{2\pi i} \int_L \frac{\prod_{j=1}^{m} \Gamma(b_j + s) \prod_{j=1}^{n} \Gamma(1 - a_j - s)}{\prod_{j=n+1}^{p} \Gamma(a_j + s) \prod_{j=m+1}^{q} \Gamma(1 - b_j - s)} z^{-s/r} ds$$

D-finite (holonomic) functions

A formal power series (or function) $f(x)$ is called *D-finite* (or *holonomic*) if there are polynomials $a_r, \ldots, a_0$ such that

$$a_r(x)f^{(r)}(x) + \ldots + a_1(x)f'(x) + a_0(x)f(x) = 0$$

A sequence $g(n)$ is called *P-recursive* (or *holonomic*) if there are polynomials $b_s, \ldots, b_0$ such that

$$b_s(n)g(n+s) + \ldots + b_1(n)g(n+1) + b_0(n)g(n) = 0$$

If $f(x) = \sum_{n=0}^{\infty} g(n)x^n$, then

$$f(x) \text{ is D-finite} \Leftrightarrow g(n) \text{ is P-recursive}$$

(In general, with $r \neq s$.)

Examples of holonomic functions

$\exp(x)$ is D-finite, as is $\exp(x^2/(x-1)) + x^{1/2}\int_0^x t\log(1+t)dt$

$\Gamma(s,z)$ is D-finite in z:

$$z\Gamma''(s,z) + (1-s+z)\Gamma'(s,z) = 0$$

$\Gamma(s,z)$ is P-recursive in s:

$$\Gamma(s+2,z) + (-1-s-z)\Gamma(s+1,z) + sz\Gamma(s,z) = 0$$

(Inhomogeneous version: $\Gamma(s+1,z) = s\Gamma(s,z) + z^s e^{-z}$)

The generalized hypergeometric function

$${}_pF_q(a_1,\ldots,a_p;b_1,\ldots,b_q;z) = \sum_{n=0}^{\infty} \frac{(a_1)_n \ldots (a_p)_n}{(b_1)_n \ldots (b_q)_n} \frac{z^n}{n!}$$

as well as the Meijer G-function are also D-finite in z.

Closure properties for holonomic functions

If $f(x)$ and $g(x)$ are D-finite, then so are:

- $Cf(x) + Dg(x)$, where C, D are constants
- $f(x)g(x)$
- $f'(x)$
- $\int f(x)dx$
- $f(A(x))$, where $A(x)$ is an algebraic function

Annihilating operators for the results of these operations and others can be computed effectively (linear algebra)

Computation

In general, D-finite and P-recursive functions/sequences permit fast evaluation (numerically or in an exact setting).

Very large numerical values: asymptotic expansions (details are subtle)
In general: complexity-reduction techniques [more on this later]

Useful symbolic software:

- `ore_algebra` (Sage)
- `gfun` (Maple)
- `HolonomicFunctions` (Mathematica)

Numerical evaluation:

- `numgfun` (Maple)

Analytic continuation

Initial values: $F(z_a) = (f(z_a), f'(z_a), \ldots, f^{(r-1)}(z_a))$
Desired values: $F(z_b) = (f(z_b), f'(z_b), \ldots, f^{(r-1)}(z_b))$

Transition matrix: $\Delta_{z_a \to z_b}$

$$F(z_1) = \Delta_{z_0 \to z_1} F(z_0)$$

$$F(z_2) = \Delta_{z_1 \to z_2} \Delta_{z_0 \to z_1} F(z_0)$$

Each transition matrix can be computed numerically using local Taylor expansion of the differential equation.

Baby example: $\exp(-20) = \exp(-10)\exp(-10)\exp(0)$

Non-examples of holonomic functions

The function

$$\frac{x}{e^x - 1} = \sum_{k=0}^{\infty} \frac{B_k}{k!} x^k$$

is not D-finite $\Leftrightarrow$ the Bernoulli numbers B_n are not P-recursive.

The functions $\Gamma(s)$ and $\zeta(s)$ are not D-finite.

The sequence $\log(n)$, the prime numbers $p(n)$ and the partition numbers $p(n)$ are examples of sequences that are not P-recursive.

Computing the gamma function

The workhorse method is the Stirling series

$$\log \Gamma(s) = (s - 1/2)\log(s) - s + \frac{2\pi}{2} + \sum_{k=1}^{N-1} \frac{B_{2k}}{2k(2k-1)s^{2k-1}} + R_N(s)$$

$$R_N(s) = \int_0^\infty \frac{B_{2N} - \tilde{B}_{2N}(t)}{2n(t+s)^{2n}} dt$$

Argument reduction:

$$\Gamma(s+n) = \Gamma(s) \cdot (s(s+1) + \cdots (s+n-1))$$

For p bits: $N \sim n \sim p$