Fast arbitrary-precision evaluation of special functions in the Arb library

Fredrik Johansson
(INRIA & IMB, Bordeaux)

OPSFA13, NIST, June 2015

Arb

C library for **arbitrary-precision interval arithmetic**. Supports complex numbers, polynomials, power series, matrices, **special functions**.

$\zeta(s)$ **Arb**

C library for **arbitrary-precision interval arithmetic**. Supports complex numbers, polynomials, power series, matrices, **special functions**.

Open source (GPL). Depends on GMP/MPIR, MPFR, FLINT. Thread safe. 100 000 lines of code, extensively tested.

Python, Sage and Julia interfaces in progress.

`http://fredrikj.net/arb/`

```
>>> sin(1)
[0.841470984807897 ± 6.08e-16]
```

```
>>> sin(1)
[0.841470984807897 ± 6.08e-16]
```

- Real numbers are [mid ± rad] intervals ("balls")
- The internal representation uses binary numbers
- Decimal pretty-printing shows only the digits of the midpoint that are known to be correct (up to ±1 ulp)

```
>>> sin(1)
[0.841470984807897 ± 6.08e-16]
```

- Real numbers are [mid ± rad] intervals ("balls")
- The internal representation uses binary numbers
- Decimal pretty-printing shows only the digits of the midpoint that are known to be correct (up to ±1 ulp)

```
>>> sin(pi() + exp(-10))
[-4.539992975e-5 ± 4.34e-15]
```

```
>>> sin(1)
[0.841470984807897 ± 6.08e-16]
```

- Real numbers are [mid ± rad] intervals ("balls")
- The internal representation uses binary numbers
- Decimal pretty-printing shows only the digits of the midpoint that are known to be correct (up to ±1 ulp)

```
>>> sin(pi() + exp(-10))
[-4.539992975e-5 ± 4.34e-15]

>>> sin(pi() + exp(-100))
[± 1.02e-15]
```

```
>>> sin(1)
[0.841470984807897 ± 6.08e-16]
```

- Real numbers are [mid ± rad] intervals ("balls")
- The internal representation uses binary numbers
- Decimal pretty-printing shows only the digits of the midpoint that are known to be correct (up to ±1 ulp)

```
>>> sin(pi() + exp(-10))
[-4.539992975e-5 ± 4.34e-15]

>>> sin(pi() + exp(-100))
[± 1.02e-15]

>>> ctx.dps = 60
>>> sin(pi() + exp(-100))
[-3.7200759760208359e-44 ± 8.42e-61]
```

Adaptive precision

```
def N(function, digits):
    ctx.dps = digits + max(5, digits * 0.05)
    while True:
        y = function()
        print("%s (at %s digits)" % (y.str(digits), ctx.dps))
        if accurate_digits(y) >= digits:
            break
        ctx.dps = ctx.dps * 2
```

Adaptive precision

```
def N(function, digits):
    ctx.dps = digits + max(5, digits * 0.05)
    while True:
        y = function()
        print("%s (at %s digits)" % (y.str(digits), ctx.dps))
        if accurate_digits(y) >= digits:
            break
        ctx.dps = ctx.dps * 2
```

```
>>> N(lambda:  sin(pi() + exp(-1000)), 20)
[± 1.37e-25] (at 25 digits)
[± 1.51e-50] (at 50 digits)
[± 6.01e-101] (at 100 digits)
[± 7.96e-201] (at 200 digits)
[± 1.07e-400] (at 400 digits)
[-5.0759588975494567653e-435 ± 8.20e-456] (at 800 digits)
```

Stress test: a Bessel function

$J_\nu(z) \qquad \nu = 10\,000 + 10\,000\,i \qquad z = 10\,000\,\pi$

Stress test: a Bessel function

$J_\nu(z) \qquad \nu = 10\,000 + 10\,000\,i \qquad z = 10\,000\,\pi$

```
>>> N(lambda:  bessel_j(10000 + 10000j, 10000 * pi()), 15)
[± 9.85e+9587] + [± 9.85e+9587]j (at 20 digits)
[± 9.85e+9587] + [± 9.85e+9587]j (at 40 digits)
⋮
[± 7.19e+9682] + [± 7.19e+9682]j (at 1280 digits)
[± 9.01e+8402] + [± 9.01e+8402]j (at 2560 digits)
[± 5.62e+5842] + [± 5.62e+5842]j (at 5120 digits)
[-1.20973469401861e+5438 ± 4.77e+5423] +
 [1.21911522763864e+5438 ± 3.09e+5423]j (at 10240 digits)
```

Stress test: a Bessel function

$J_\nu(z) \qquad \nu = 10\,000 + 10\,000\,i \qquad z = 10\,000\,\pi$

```
>>> N(lambda: bessel_j(10000 + 10000j, 10000 * pi()), 15)
[± 9.85e+9587] + [± 9.85e+9587]j (at 20 digits)
[± 9.85e+9587] + [± 9.85e+9587]j (at 40 digits)
⋮
[± 7.19e+9682] + [± 7.19e+9682]j (at 1280 digits)
[± 9.01e+8402] + [± 9.01e+8402]j (at 2560 digits)
[± 5.62e+5842] + [± 5.62e+5842]j (at 5120 digits)
[-1.20973469401861e+5438 ± 4.77e+5423] +
 [1.21911522763864e+5438 ± 3.09e+5423]j (at 10240 digits)
```

This takes

- **1 second** in Arb
- **200 seconds** in mpmath
- **100 000 seconds** in Mathematica

Stress test: the partition function $p(n)$

$$p(n) \qquad 1, 1, 2, 3, 5, 7, 11, 15, 22, 30, 42, 56, 77, 101, \ldots$$

$$p(n) = \frac{1}{\pi\sqrt{2}} \sum_{k=1}^{\infty} \sqrt{k}\, A_k(n)\, \frac{d}{dn} \left(\frac{1}{\sqrt{n - \frac{1}{24}}} \sinh \left(\frac{\pi}{k} \sqrt{\frac{2}{3}\left(n - \frac{1}{24}\right)} \right) \right)$$

Stress test: the partition function $p(n)$

$$p(n) \qquad 1, 1, 2, 3, 5, 7, 11, 15, 22, 30, 42, 56, 77, 101, \ldots$$

$$p(n) = \frac{1}{\pi\sqrt{2}} \sum_{k=1}^{\infty} \sqrt{k}\, A_k(n)\, \frac{d}{dn} \left(\frac{1}{\sqrt{n - \frac{1}{24}}} \sinh\left(\frac{\pi}{k} \sqrt{\frac{2}{3}\left(n - \frac{1}{24}\right)} \right) \right)$$

Record computation done with Arb:

$$p(10^{20}) = \underbrace{18381765\ldots88091448}_{11\,140\,086\,260 \text{ digits}}$$

[1 710 193 158 terms, 200 CPU hours, 130 GB memory]

THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES®

founded in 1964 by N. J. A. Sloane

Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

A110375 Numbers n such that Maple 9.5, Maple 10, Maple 11 and Maple 12 give the wrong answers for the number of partitions of n. 2

11269, 11566, 12376, 12430, 12700, 12754, 15013, 17589, 17797, 18181, 18421, 18453, 18549, 18597, 18885, 18949, 18997, 20865, 21531, 21721, 21963, 22683, 23421, 23457, 23547, 23691, 23729, 23853, 24015, 24087, 24231, 24339, 24519, 24591, 24627, 24681, 24825, 24933, 25005, 25023, 25059, 25185, 25293, 27020 (list; graph; refs; listen; history; text; internal format)

OFFSET 1,1

COMMENTS

```
Based on various postings on the Web, sent to N. J. A. Sloane by R. J. Mathar. Thanks
  to several correspondents who sent information about other versions of Maple.
Mathematica 6.0, DrScheme and pari-2.3.3 all give the correct answers.
Ramanujan's congruence says that numbpart(5*k+4)==0 mod 5, so
  numbpart(11269)=...851==1 mod 5 can't be correct. [Robert Gerbicz, May 13 2008]
```

LINKS

```
Table of n, a(n) for n=1..44.
Author?, Concerning this sequence
```

EXAMPLE

```
From PARI, the correct answer:
numbpart(11269)
2311391772313039755144117876494556289590601993601099725578515191051551761\
803182158917958749053182741632480330718850
From Maple 11, incorrect:
combinat[numbpart](11269);
2311391772313039755144117876494556289590601993601099725578515191051551761\
80318215891795874905318274163248033071851
On the other hand, the old Maple 6 gives the correct answer.
```

Coverage of special functions

NIST Digital Library of Mathematical Functions

Foreword
Preface
Mathematical Introduction
1 Algebraic and Analytic Methods
2 Asymptotic Approximations
3 Numerical Methods
4 Elementary Functions
5 Gamma Function
6 Exponential, Logarithmic, Sine, and Cosine Integrals
7 Error Functions, Dawson's and Fresnel Integrals
8 Incomplete Gamma and Related Functions
9 Airy and Related Functions
10 Bessel Functions
11 Struve and Related Functions
12 Parabolic Cylinder Functions
13 Confluent Hypergeometric Functions
14 Legendre and Related Functions
15 Hypergeometric Function
16 Generalized Hypergeometric Functions and Meijer G-Function
17 q-Hypergeometric and Related Functions
18 Orthogonal Polynomials
19 Elliptic Integrals
20 Theta Functions
21 Multidimensional Theta Functions
22 Jacobian Elliptic Functions
23 Weierstrass Elliptic and Modular Functions
24 Bernoulli and Euler Polynomials
25 Zeta and Related Functions
26 Combinatorial Analysis
27 Functions of Number Theory
28 Mathieu Functions and Hill's Equation
29 Lamé Functions
30 Spheroidal Wave Functions
31 Heun Functions
32 Painlevé Transcendents
33 Coulomb Functions
34 $3j, 6j, 9j$ Symbols
35 Functions of Matrix Argument
36 Integrals with Coalescing Saddles
Bibliography
Index
Notations
Software
Errata

Most functions can be evaluated over $\mathbb{C}$

Many functions can be evaluated over $\mathbb{C}[[x]]/\langle x^n \rangle$

Generalized hypergeometric functions

$$ {}_pF_q(a_1 \ldots a_p; b_1 \ldots b_q; z) = \sum_{k=0}^{\infty} \frac{(a_1)_k \cdots (a_p)_k}{(b_1)_k \cdots (b_q)_k} \frac{z^k}{k!} $$

$$ S \pm R \qquad \underbrace{S = \sum_{k=0}^{N-1} T(k)}_{\text{Using interval arithmetic}} \qquad \underbrace{\left| \sum_{k=N}^{\infty} T(k) \right| \le R}_{\text{Upper bound}} $$

Generalized hypergeometric functions

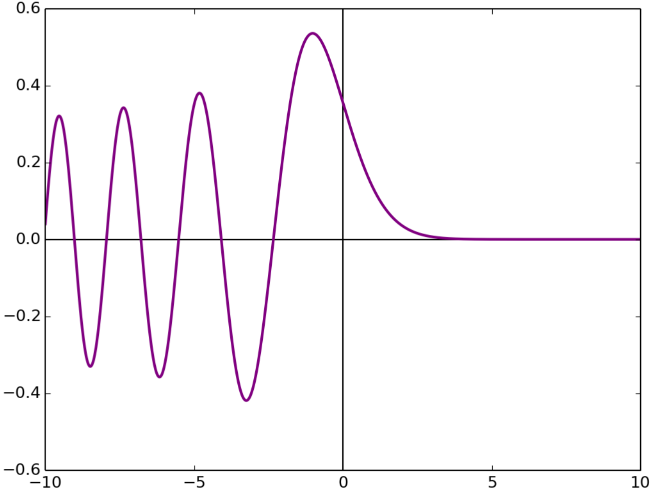
$$ {}_pF_q(a_1 \ldots a_p; b_1 \ldots b_q; z) = \sum_{k=0}^{\infty} \frac{(a_1)_k \cdots (a_p)_k}{(b_1)_k \cdots (b_q)_k} \frac{z^k}{k!} $$

$$ S \pm R \qquad \underbrace{S = \sum_{k=0}^{N-1} T(k)}_{\text{Using interval arithmetic}} \qquad \underbrace{\left| \sum_{k=N}^{\infty} T(k) \right| \le R}_{\text{Upper bound}} $$

Evaluation supported for $a_i, b_i, z \in \mathbb{C}[[x]]/\langle x^n \rangle$ (when convergent)

Error bounds for the *divergent* asymptotic series ${}_2F_0(a, b, z)$ with $a, b, z \in \mathbb{C}$ based on Olver (DLMF 13.7).

Let's compute the Airy function Ai(z)



Let's compute the Airy function Ai(z)

$$\text{Ai}(z) = \frac{1}{\pi}\sqrt{\frac{z}{3}}\, K_{1/3}\left(\frac{2}{3}z^{3/2}\right), \qquad \text{re}(z) > 0$$

Let's compute the Airy function Ai(z)

$$\mathrm{Ai}(z) = \frac{1}{\pi}\sqrt{\frac{z}{3}}\, K_{1/3}\left(\frac{2}{3}z^{3/2}\right), \qquad \mathrm{re}(z) > 0$$

$$K_\nu(z) = \sqrt{\pi}\,(2z)^{-\nu} e^{-z}\, U(\nu + \tfrac{1}{2}, 2\nu + 1, 2z)$$

Let's compute the Airy function Ai(z)

$$\text{Ai}(z) = \frac{1}{\pi}\sqrt{\frac{z}{3}}\, K_{1/3}\left(\frac{2}{3}z^{3/2}\right), \qquad \text{re}(z) > 0$$

$$K_\nu(z) = \sqrt{\pi}\,(2z)^{-\nu} e^{-z}\, U(\nu + \tfrac{1}{2}, 2\nu + 1, 2z)$$

$$U(a, b, z) = \frac{\Gamma(1-b)}{\Gamma(a-b+1)}\,{}_1F_1(a, b, z) + \frac{\Gamma(b-1)}{\Gamma(a)z^{b-1}}\,{}_1F_1(a-b+1, 2-b, z)$$

$$U(a, b, z) \sim z^{-a}\,{}_2F_0\left(a, a-b+1; ; -\frac{1}{z}\right), \qquad |z| \text{ large}$$

Let's compute the Airy function Ai(z)

$$\mathrm{Ai}(z) = \frac{1}{\pi}\sqrt{\frac{z}{3}}\, K_{1/3}\left(\frac{2}{3}z^{3/2}\right), \qquad \mathrm{re}(z) > 0$$

$$K_\nu(z) = \sqrt{\pi}\,(2z)^{-\nu} e^{-z}\, U(\nu + \tfrac{1}{2}, 2\nu + 1, 2z)$$

$$U(a, b, z) = \frac{\Gamma(1-b)}{\Gamma(a-b+1)}\,{}_1F_1(a, b, z) + \frac{\Gamma(b-1)}{\Gamma(a) z^{b-1}}\,{}_1F_1(a-b+1, 2-b, z)$$

$$U(a, b, z) \sim z^{-a}\,{}_2F_0\left(a, a-b+1; ; -\frac{1}{z}\right), \qquad |z| \text{ large}$$

Error propagation is automatic. We only need to select a correct (optionally, efficient) formula in each region.

```
>>> N(lambda: airy_ai(1), 20)
[0.13529241631288141552 ± 4.15e-21] (at 25 digits)
```

```
>>> N(lambda: airy_ai(1), 20)
[0.13529241631288141552 ± 4.15e-21] (at 25 digits)

>>> N(lambda: airy_ai(10), 20)
[1.1048e-10 ± 5.45e-15] (at 25 digits)
[1.1047532552898685934e-10 ± 4.50e-30] (at 50 digits)
```

```
>>> N(lambda: airy_ai(1), 20)
[0.13529241631288141552 ± 4.15e-21] (at 25 digits)

>>> N(lambda: airy_ai(10), 20)
[1.1048e-10 ± 5.45e-15] (at 25 digits)
[1.1047532552898685934e-10 ± 4.50e-30] (at 50 digits)

>>> N(lambda: airy_ai(100), 20)
[2.6344821520881844896e-291 ± 4.95e-311] (at 25 digits)
```

```
>>> N(lambda: airy_ai(1), 20)
[0.13529241631288141552 ± 4.15e-21] (at 25 digits)

>>> N(lambda: airy_ai(10), 20)
[1.1048e-10 ± 5.45e-15] (at 25 digits)
[1.1047532552898685934e-10 ± 4.50e-30] (at 50 digits)

>>> N(lambda: airy_ai(100), 20)
[2.6344821520881844896e-291 ± 4.95e-311] (at 25 digits)

>>> N(lambda: log(airy_ai(10000000000000)), 20)
[-6.6666666666666666668e+20 ± 4.02] (at 25 digits)
```

```
>>> N(lambda: airy_ai(1), 20)
[0.13529241631288141552 ± 4.15e-21] (at 25 digits)

>>> N(lambda: airy_ai(10), 20)
[1.1048e-10 ± 5.45e-15] (at 25 digits)
[1.1047532552898685934e-10 ± 4.50e-30] (at 50 digits)

>>> N(lambda: airy_ai(100), 20)
[2.6344821520881844896e-291 ± 4.95e-311] (at 25 digits)

>>> N(lambda: log(airy_ai(10000000000000)), 20)
[-6.6666666666666666668e+20 ± 4.02] (at 25 digits)

>>> N(lambda: log(airy_ai(10000000000000)), 40)
[-666666666666666666675.9912266156304719572 ± 1.87e-20]
(at 45 digits)
```

Gamma, zeta and polylogarithm functions

$$\Gamma(s), \quad \zeta(s,z) = \sum_{k=0}^{\infty} \frac{1}{(k+z)^s}, \quad \mathrm{Li}_s(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^s}$$

Evaluation supported for $s \in \mathbb{C}[[x]]/\langle x^n \rangle, \quad z \in \mathbb{C}$

Gamma, zeta and polylogarithm functions

$$\Gamma(s), \quad \zeta(s,z) = \sum_{k=0}^{\infty} \frac{1}{(k+z)^s}, \quad \mathrm{Li}_s(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^s}$$

Evaluation supported for $s \in \mathbb{C}[[x]]/\langle x^n \rangle, \quad z \in \mathbb{C}$

Algorithms:

- Euler-Maclaurin summation + functional equations
- Hypergeometric series and other methods for special values
- Some new error bounds + tricks for high precision or large n

Stress test: high-order derivatives of $\zeta(s)$

Keiper/Li: the Riemann hypothesis is equivalent to the statement

$$\lambda_n > 0 \quad \text{for all } n$$

where

$$\log(2\,\xi(\tfrac{x}{x-1})) = \sum_{n=1}^{\infty} \lambda_n x^n, \quad \xi(s) = \frac{s(s-1)}{2\pi^{s/2}}\Gamma(s/2)\zeta(s)$$

Stress test: high-order derivatives of $\zeta(s)$

Keiper/Li: the Riemann hypothesis is equivalent to the statement

$$\lambda_n > 0 \quad \text{for all } n$$

where

$$\log(2\,\xi(\tfrac{x}{x-1})) = \sum_{n=1}^{\infty} \lambda_n x^n, \quad \xi(s) = \frac{s(s-1)}{2\pi^{s/2}}\Gamma(s/2)\zeta(s)$$

Get-rich-quick-scheme:

1. Evaluate $\log(2\,\xi(\tfrac{x}{x-1}))$ in $\mathbb{C}[[x]]/\langle x^{N+1}\rangle$

Stress test: high-order derivatives of $\zeta(s)$

Keiper/Li: the Riemann hypothesis is equivalent to the statement

$$\lambda_n > 0 \quad \text{for all } n$$

where

$$\log(2\,\xi(\tfrac{x}{x-1})) = \sum_{n=1}^{\infty} \lambda_n x^n, \quad \xi(s) = \frac{s(s-1)}{2\pi^{s/2}}\Gamma(s/2)\zeta(s)$$

Get-rich-quick-scheme:

1. Evaluate $\log(2\,\xi(\tfrac{x}{x-1}))$ in $\mathbb{C}[[x]]/\langle x^{N+1}\rangle$
2. Read off the coefficients $\lambda_1, \ldots, \lambda_N$

Stress test: high-order derivatives of $\zeta(s)$

Keiper/Li: the Riemann hypothesis is equivalent to the statement

$$\lambda_n > 0 \quad \text{for all } n$$

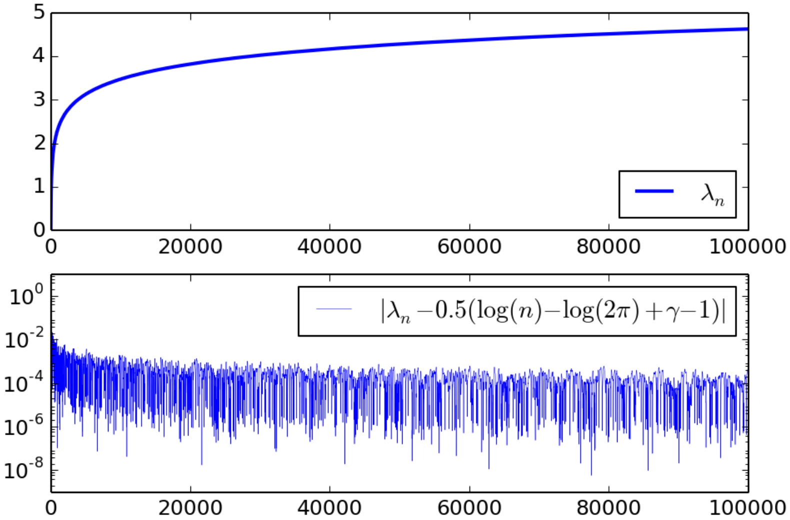
where

$$\log(2\,\xi(\tfrac{x}{x-1})) = \sum_{n=1}^{\infty} \lambda_n x^n, \quad \xi(s) = \frac{s(s-1)}{2\pi^{s/2}}\Gamma(s/2)\zeta(s)$$

Get-rich-quick-scheme:

1. Evaluate $\log(2\,\xi(\tfrac{x}{x-1}))$ in $\mathbb{C}[[x]]/\langle x^{N+1}\rangle$
2. Read off the coefficients $\lambda_1, \ldots, \lambda_N$
3. If any $\lambda_n < 0$, collect the **\$1,000,000 Millennium Prize**.

Rigorous computation, $N = 100\,000$ (20 hours, 50 GB memory)



Theta functions, modular forms, elliptic functions

$$\mathrm{agm}(a, b) = \mathrm{agm}(\tfrac{1}{2}(a+b), \sqrt{ab}), \quad a, b \in \mathbb{C}[[x]]/\langle x^n \rangle$$

$$\theta(z, \tau) = \sum_{n=-\infty}^{\infty} \exp(\pi i [n^2 \tau + 2nz]), \quad z \in \mathbb{C}[[x]]/\langle x^n \rangle, \quad \tau \in \mathbb{H}$$

Theta functions, modular forms, elliptic functions

$$\operatorname{agm}(a,b) = \operatorname{agm}\left(\tfrac{1}{2}(a+b), \sqrt{ab}\right), \quad a, b \in \mathbb{C}[[x]]/\langle x^n \rangle$$

$$\theta(z,\tau) = \sum_{n=-\infty}^{\infty} \exp(\pi i [n^2\tau + 2nz]), \quad z \in \mathbb{C}[[x]]/\langle x^n \rangle, \quad \tau \in \mathbb{H}$$

Derived functions:

- Classical modular forms (Dedekind eta function, etc.)
- Weierstrass elliptic functions
- Complete elliptic integrals

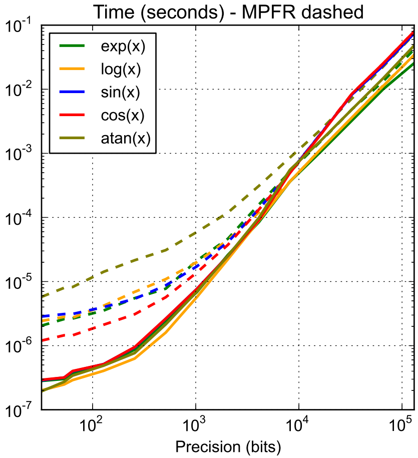
A modular form magnified by a factor 10^{100}

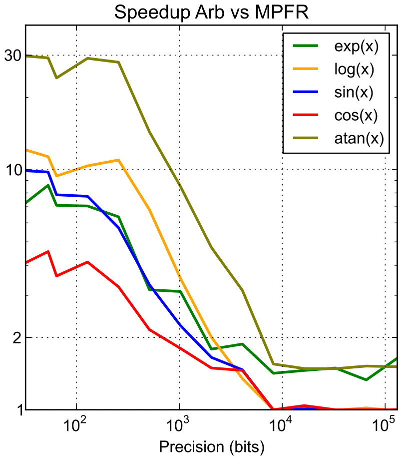
$j(\tau)$ on $[\sqrt{13}, \sqrt{13} + 10^{-101}] + [0, 2.5 \times 10^{-102}]i$

Modular transformations $\tau \mapsto \frac{a\tau+b}{c\tau+d}$ with 100-digit coefficients map τ to the fundamental domain

Rendered using 768-bit arithmetic (5 000 pixels / second)

Performance of elementary functions





Time (microseconds) at quad (113 bits) precision:

	exp	sin	cos	log	atan
MPFR	5.76	7.29	3.42	8.01	21.30
libquadmath	4.51	4.71	4.57	5.39	4.32
QD	0.73	**0.69**	**0.69**	0.82	1.08
Arb	**0.65**	0.81	0.79	**0.61**	**0.68**

Time (microseconds) at quad-double (212 bits) precision:

	exp	sin	cos	log	atan
MPFR	7.87	9.23	5.06	12.60	33.00
QD	6.09	5.77	5.76	20.10	24.90
Arb	**1.29**	**1.49**	**1.49**	**1.26**	**1.23**

Recipe for elementary functions

$\exp(x)$	$\sin(x), \cos(x)$	$\log(1+x)$	$\operatorname{atan}(x)$

$$\downarrow$$

Domain reduction using π and $\log(2)$

$$\downarrow$$

$x \in [0, \log(2))$	$x \in [0, \pi/4)$	$x \in [0, 1)$	$x \in [0, 1)$

Recipe for elementary functions

$\exp(x)$ $\sin(x), \cos(x)$ $\log(1+x)$ $\operatorname{atan}(x)$

$\downarrow$

Domain reduction using π and $\log(2)$

$\downarrow$

$x \in [0, \log(2))$ $x \in [0, \pi/4)$ $x \in [0, 1)$ $x \in [0, 1)$

$\downarrow$

Argument-halving $r \approx 8$ times

$$\exp(x) = [\exp(x/2)]^2$$

$$\log(1+x) = 2\log(\sqrt{1+x})$$

$\downarrow$

$x \in [0, 2^{-r})$

$\downarrow$

Taylor series

Better recipe at medium precision

$\exp(x)$ $\sin(x), \cos(x)$ $\log(1+x)$ $\operatorname{atan}(x)$

$\downarrow$

Domain reduction using π and $\log(2)$

$\downarrow$

$x \in [0, \log(2))$ $x \in [0, \pi/4)$ $x \in [0, 1)$ $x \in [0, 1)$

$\downarrow$

Lookup table with $2^r \approx 2^8$ entries

$$\exp(t + x) = \exp(t)\exp(x)$$

$$\log(1 + t + x) = \log(1 + t) + \log(1 + x/(1 + t))$$

$\downarrow$

$x \in [0, 2^{-r})$

$\downarrow$

Taylor series

Optimizing lookup tables

$m = 2$ tables with $2^5 + 2^5$ entries gives same reduction as
$m = 1$ table with 2^{10} entries

Function	Precision	m	r	Entries	Size (KiB)
exp	≤ 512	1	8	178	11.125
exp	≤ 4608	2	5	23+32	30.9375
sin	≤ 512	1	8	203	12.6875
sin	≤ 4608	2	5	26+32	32.625
cos	≤ 512	1	8	203	12.6875
cos	≤ 4608	2	5	26+32	32.625
log	≤ 512	2	7	128+128	16
log	≤ 4608	2	5	32+32	36
atan	≤ 512	1	8	256	16
atan	≤ 4608	2	5	32+32	36
Total					236.6875

Evaluating Taylor series using rectangular splitting

Paterson and Stockmeyer, 1973:

$\sum_{i=0}^{n} \square x^i$ in $O(n)$ cheap steps $+$ $O(n^{1/2})$ expensive steps

Evaluating Taylor series using rectangular splitting

Paterson and Stockmeyer, 1973:

$\sum_{i=0}^{n} \square x^i$ in $O(n)$ cheap steps + $O(n^{1/2})$ expensive steps

$$
\begin{array}{lllllllllll}
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^4 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^8 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^{12} &
\end{array}
$$

Evaluating Taylor series using rectangular splitting

Paterson and Stockmeyer, 1973:

$\sum_{i=0}^{n} \square x^i$ in $O(n)$ cheap steps $+$ $O(n^{1/2})$ expensive steps

$$
\begin{array}{lllllllllll}
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^4 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^8 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^{12} &
\end{array}
$$

- Smith, 1989: elementary and hypergeometric functions

Evaluating Taylor series using rectangular splitting

Paterson and Stockmeyer, 1973:

$\sum_{i=0}^{n} \square x^i$ in $O(n)$ cheap steps + $O(n^{1/2})$ expensive steps

$$
\begin{array}{llllllllllll}
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^4 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^8 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^{12} &
\end{array}
$$

- Smith, 1989: elementary and hypergeometric functions
- Brent & Zimmermann, 2010: improvements to Smith

Evaluating Taylor series using rectangular splitting

Paterson and Stockmeyer, 1973:

$\sum_{i=0}^{n} \square x^i$ in $O(n)$ cheap steps $+$ $O(n^{1/2})$ expensive steps

$$
\begin{array}{lllllllll}
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^4 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^8 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^{12} &
\end{array}
$$

- Smith, 1989: elementary and hypergeometric functions
- Brent & Zimmermann, 2010: improvements to Smith
- FJ, 2014: generalization to holonomic functions

Evaluating Taylor series using rectangular splitting

Paterson and Stockmeyer, 1973:

$\sum_{i=0}^{n} \square x^i$ in $O(n)$ cheap steps $+$ $O(n^{1/2})$ expensive steps

$$
\begin{array}{llllllllllll}
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^4 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^8 & + \\
(& \square & + & \square x & + & \square x^2 & + & \square x^3 &) & x^{12} &
\end{array}
$$

- Smith, 1989: elementary and hypergeometric functions
- Brent & Zimmermann, 2010: improvements to Smith
- FJ, 2014: generalization to holonomic functions
- FJ, 2015: optimized algorithm for elementary functions

Logarithmic series

$$x + \frac{1}{2}x^2 + x^3\left\{\frac{1}{3} + \frac{1}{4}x + \frac{1}{5}x^2 + x^3\left\{\frac{1}{6} + \frac{1}{7}x + \frac{1}{8}x^2\right\}\right\}$$

Logarithmic series

$$x + \frac{1}{2}x^2 + x^3\left\{\frac{1}{3} + \frac{1}{4}x + \frac{1}{5}x^2 + x^3\left\{\frac{1}{6} + \frac{1}{7}x + \frac{1}{8}x^2\right\}\right\}$$

Logarithmic series (fewer divisions)

$$x+\frac{1}{60}\left[30x^2+x^3\left\{20+15x+12x^2+x^3\left\{10+\frac{1}{56}\left[60\left[8x+7x^2\right]\right]\right\}\right\}\right]$$

Logarithmic series

$$x + \frac{1}{2}x^2 + x^3\left\{\frac{1}{3} + \frac{1}{4}x + \frac{1}{5}x^2 + x^3\left\{\frac{1}{6} + \frac{1}{7}x + \frac{1}{8}x^2\right\}\right\}$$

Logarithmic series (fewer divisions)

$$x+\frac{1}{60}\left[30x^2+x^3\left\{20+15x+12x^2+x^3\left\{10+\frac{1}{56}\left[60\left[8x+7x^2\right]\right]\right\}\right\}\right]$$

Exponential series

$$1+x+\frac{1}{2}\left[x^2+\frac{1}{3}x^3\left\{1+\frac{1}{4}\left[x+\frac{1}{5}\left[x^2+\frac{1}{6}x^3\left\{1+\frac{1}{7}\left[x+\frac{1}{8}x^2\right]\right\}\right]\right]\right\}\right]$$

Logarithmic series

$$x + \frac{1}{2}x^2 + x^3\left\{\frac{1}{3} + \frac{1}{4}x + \frac{1}{5}x^2 + x^3\left\{\frac{1}{6} + \frac{1}{7}x + \frac{1}{8}x^2\right\}\right\}$$

Logarithmic series (fewer divisions)

$$x+\frac{1}{60}\left[30x^2+x^3\left\{20+15x+12x^2+x^3\left\{10+\frac{1}{56}\left[60\left[8x+7x^2\right]\right]\right\}\right\}\right]$$

Exponential series

$$1+x+\frac{1}{2}\left[x^2+\frac{1}{3}x^3\left\{1+\frac{1}{4}\left[x+\frac{1}{5}\left[x^2+\frac{1}{6}x^3\left\{1+\frac{1}{7}\left[x+\frac{1}{8}x^2\right]\right\}\right]\right]\right\}\right]$$

Exponential series (fewer divisions)

$$1+x+\frac{1}{24}\left[12x^2+x^3\left\{4+1\left[x+\frac{1}{30}\left[6x^2+x^3\left\{1+\frac{1}{56}\left[8x+x^2\right]\right\}\right]\right]\right\}\right]$$

Implementation

For each Taylor series (exp, sinh, cosh, sin, cos, atanh),
a **near-optimal evaluation sequence is precomputed**

Implementation

For each Taylor series (exp, sinh, cosh, sin, cos, atanh), a **near-optimal evaluation sequence is precomputed**

The evaluation is done in **fixed-point arithmetic** with a low-level representation (n words store $0 \le x < 1$ with $64n$-bit precision)

Implementation

For each Taylor series (exp, sinh, cosh, sin, cos, atanh), a **near-optimal evaluation sequence is precomputed**

The evaluation is done in **fixed-point arithmetic** with a low-level representation (n words store $0 \leq x < 1$ with $64n$-bit precision)

An exhaustive precomputation is used to **prove correctness**

- Error bounds
- No overflows possible

Final remarks

For **special functions**, we can simultaneously achieve:

- **Rigorous error bounds** everywhere
- **High performance** (as fast as previous arbitrary-precision software, or faster)

Final remarks

For **special functions**, we can simultaneously achieve:

- **Rigorous error bounds** everywhere
- **High performance** (as fast as previous arbitrary-precision software, or faster)

Important special functions not yet implemented include:

- Analytic continuation of ${}_2F_1$, ${}_3F_2$, ...
- Incomplete elliptic integrals
- Holonomic functions (see work by Marc Mezzarobba)
- More general L-functions and modular forms

Final remarks

For **special functions**, we can simultaneously achieve:

- **Rigorous error bounds** everywhere
- **High performance** (as fast as previous arbitrary-precision software, or faster)

Important special functions not yet implemented include:

- Analytic continuation of ${}_2F_1$, ${}_3F_2$, ...
- Incomplete elliptic integrals
- Holonomic functions (see work by Marc Mezzarobba)
- More general L-functions and modular forms

Thank you!