High-precision methods for zeta functions
Part 3: fast evaluation of sequences

Fredrik Johansson
INRIA Bordeaux

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

Linearly recurrent sequences

$$\underbrace{c(k+1)}_{\text{vector}} = \underbrace{M(k)}_{r \times r \text{ matrix}} \cdot \underbrace{c(k)}_{\text{vector}}$$

Order-r scalar recurrence

$$c(k+r) + a_{r-1}(k)c(k+r-1) + \ldots + a_0(k)c(k)$$

can be rewritten

$$\begin{bmatrix} c(k+1) \\ \vdots \\ c(k+r) \end{bmatrix} = \begin{bmatrix} & 1 & \\ & \ddots & \\ & & 1 \\ -a_0(k) & \cdots & -a_{r-1}(k) \end{bmatrix} \begin{bmatrix} c(k) \\ \vdots \\ c(k+r-1) \end{bmatrix}$$

How to compute the nth entry

Naively

$$c(1) = M(0)c(0)$$
$$c(2) = M(1)c(1)$$
$$c(3) = M(2)c(2)$$
$$\vdots$$
$$c(n) = M(n-1)c(n-1)$$

Cleverly

$$c(n) = [M(n-1)M(n-2)\cdots M(1)M(0)]c(0)$$

Exploit structure of matrix product!

Example: Fibonacci numbers

$$\begin{bmatrix} F(k+1) \\ F(k+2) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}}_{M(k)} \begin{bmatrix} F(k) \\ F(k+1) \end{bmatrix}$$

M is constant: $M(n-1)M(n-2)\cdots M(0) = M^n$

We can compute M^n in $O(\log(n))$ arithmetic operations using the binary exponentiation algorithm

We can also diagonalize M, giving

$$F(n) = \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{2^n\sqrt{5}}$$

Example: Taylor series of D-finite functions

$$T(k) = \frac{x^k}{k!}, \quad T(k+1) = (x/(k+1))\,T(k)$$

$$S(n) = \sum_{k=0}^{n-1} \frac{x^k}{k!}, \quad S(k+1) = T(k) + S(k)$$

$$\begin{bmatrix} T(k+1) \\ S(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} x/(k+1) & \\ 1 & 1 \end{bmatrix}}_{M(k)} \begin{bmatrix} T(k) \\ S(k) \end{bmatrix}$$

With $n \approx \infty$,

$$\begin{bmatrix} 0 \\ \exp(x) \end{bmatrix} \approx M(n-1)M(n-2)\cdots M(1)M(0) \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Three algorithms

- Binary splitting
  - Growing objects, e.g. $\mathbb{Q}$, $\mathbb{Q}[x]$
- Fast multipoint evaluation
  - Fixed-precision objects, e.g. $\mathbb{R}$, $\mathbb{Z}/p\mathbb{Z}$
- Rectangular splitting
  - Mixed objects, e.g. $\mathbb{Q} + \mathbb{R}$

Binary splitting

Computing $n! = 1 \cdot 2 \cdots (n-2) \cdot (n-1) \cdot n$ using repeated multiplication:

1
2
6
24
120
720
5040
40320
362880
3628800
39916800
479001600
6227020800
87178291200
1307674368000
20922789888000

Height: n
Width: $O(n \log n)$
Bit complexity: $O\tilde{}(n^2)$

Binary splitting

Computing $n! = 1 \cdot 2 \cdots (n-2) \cdot (n-1) \cdot n$ using binary splitting:

```
1  2     3  4     5  6     7  8     9  10     11  12     13  14     15  16
2        12       30       56       90        132        182        240
24                1680               11880              43680
40320                                518918400
20922789888000
```

Height: $O(\log n)$
Width: $O(n \log n)$
Bit complexity: $O^{\sim}(n)$

This idea applies to any linear recurrence over $\mathbb{Q}$ (or an algebraic number field) where the bit length of the entries in $M(k)$ grows slowly, e.g. like $O(\log k)$

High-precision computation of constants

[Machin, 1706]:

$$\frac{\pi}{4} = 4\,\mathrm{atan}\left(\frac{1}{5}\right) - \mathrm{atan}\left(\frac{1}{239}\right), \quad \mathrm{atan}(x) = \sum_{k=0}^{\infty}(-1)^k \frac{x^{2k+1}}{(2k+1)}$$

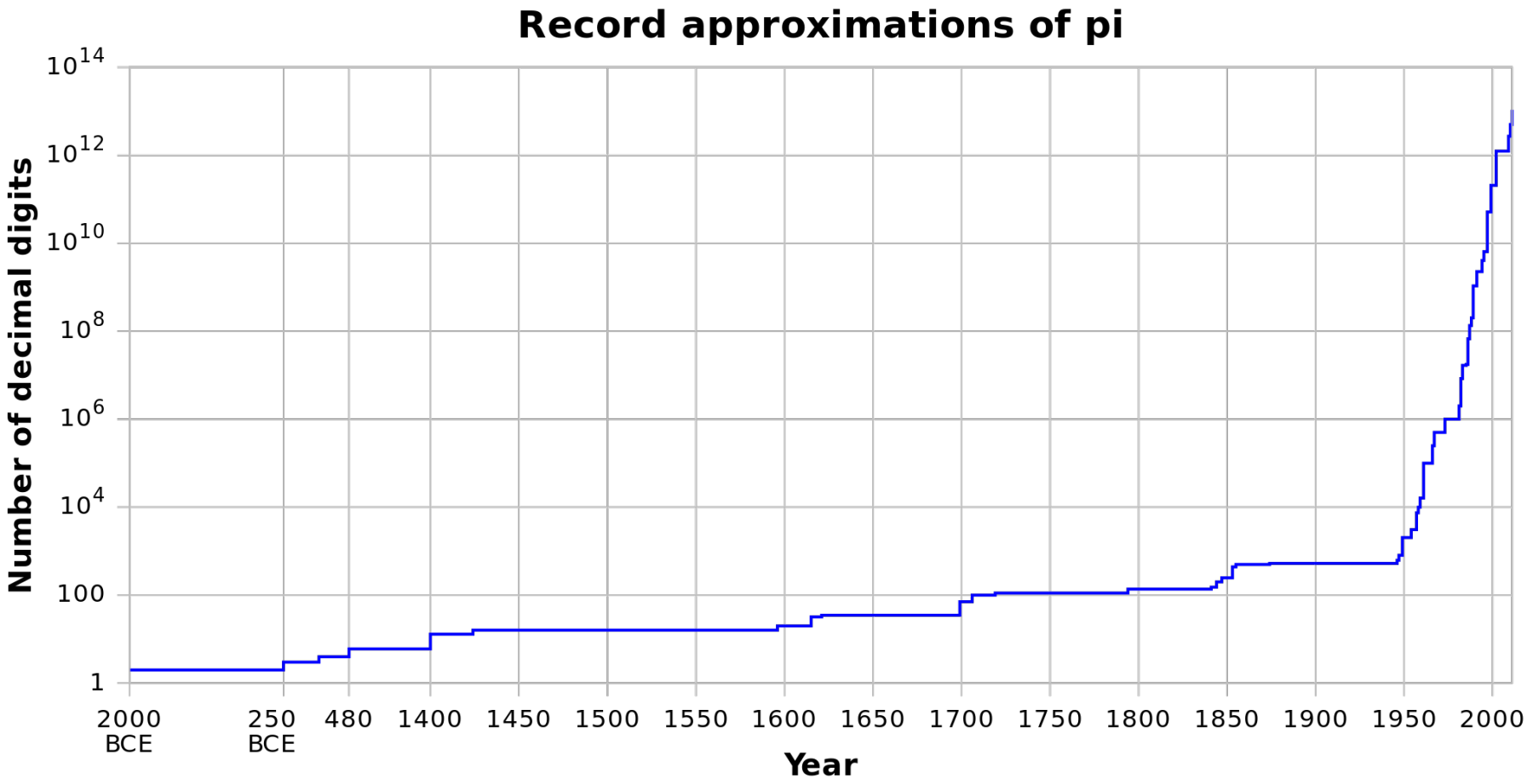
[Ramanujan, 1910] – not proved until [Borwein^2, 1987]:

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801}\sum_{k=0}^{\infty}\frac{(4k)!(1103+26390k)}{(k!)^4 396^{4k}}$$

[Chudnovsky^2, 1989]:

$$\frac{1}{\pi} = 12\sum_{k=0}^{\infty}\frac{(-1)^k(6k)!(13591409+545140134k)}{(3k)!(k!)^3 640320^{3k+3/2}}$$

The current π record is 13 300 000 000 000 digits, set in 2014. The record before 1946 was 707 digits (only 527 correct), accomplished by William Shanks in 1873 after 15 years of work.



(Credit: `en.wikipedia.org/wiki/Chronology_of_computation_of_pi`)

Fast logarithms of nearby integers

To evaluate

$$\sum_{k=1}^{N} \frac{1}{k^s} = \sum_{k=1}^{N} \exp(s \log(k))$$

we need the logarithms of consecutive primes $k = 2, 3, 5, 7, 11, \ldots$
We can compute $\log(q)$ from $\log(p)$ using

$$\log(q) = \log(p) + 2 \operatorname{atanh}\left(\frac{q-p}{q+p}\right)$$

where

$$\operatorname{atanh}(x) = \sum_{k=0}^{\infty} \frac{x^{2k+1}}{(2k+1)}$$

Fast computation of some more constants

$$\zeta(3) = \frac{1}{64}\sum_{k=0}^{\infty}(-1)^k(205k^2 + 250k + 77)\frac{(k!)^{10}}{[(2k+1)!]^5}$$

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

Euler's constant

The fastest known algorithm for Euler's constant ($\gamma = 0.577\ldots$) is due to Richard Brent and Edwin McMillan (1980).

$$\gamma = \frac{S_0(2n) - K_0(2n)}{I_0(2n)} - \log(n)$$

$$S_0(x) = \sum_{k=0}^{\infty} \frac{H_k}{(k!)^2} \left(\frac{x}{2}\right)^{2k}, \quad I_0(x) = \sum_{k=0}^{\infty} \frac{1}{(k!)^2} \left(\frac{x}{2}\right)^{2k}$$

$$2xI_0(x)K_0(x) \sim \sum_{k=0}^{\infty} \frac{[(2k)!]^3}{(k!)^4 8^{2k} x^{2k}}$$

If all series are truncated optimally, the error is less than $24e^{-8n}$. This was not proved rigorously until recently [Brent and FJ, *A bound for the error term in the Brent-McMillan algorithm*, 2015]

Fast evaluation of D-finite functions

If $f(z)$ satisfies a linear differential equation with coefficients in $\overline{\mathbb{Q}}[z]$, then for any fixed $c \in \mathbb{C}$, we can compute $r \in \mathbb{Q}[i]$ with

$$|r - f(c)| < 2^{-p}$$

in $O^{\tilde{}}(p)$ bit operations [with some caveats].

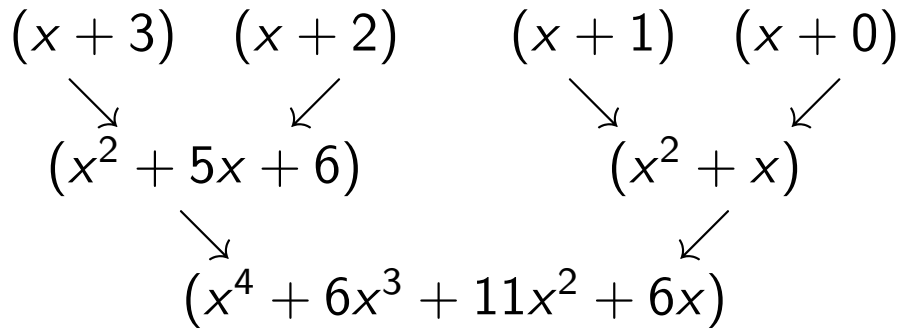
Idea: analytic continuation + binary splitting

[Chudnovsky^2, 1986, van der Hoeven, 2000]

Binary splitting for polynomials

Example: rising factorials $x(x+1)\cdots(x+n-1)$

$M(k) = (x+k), \qquad n = 4$

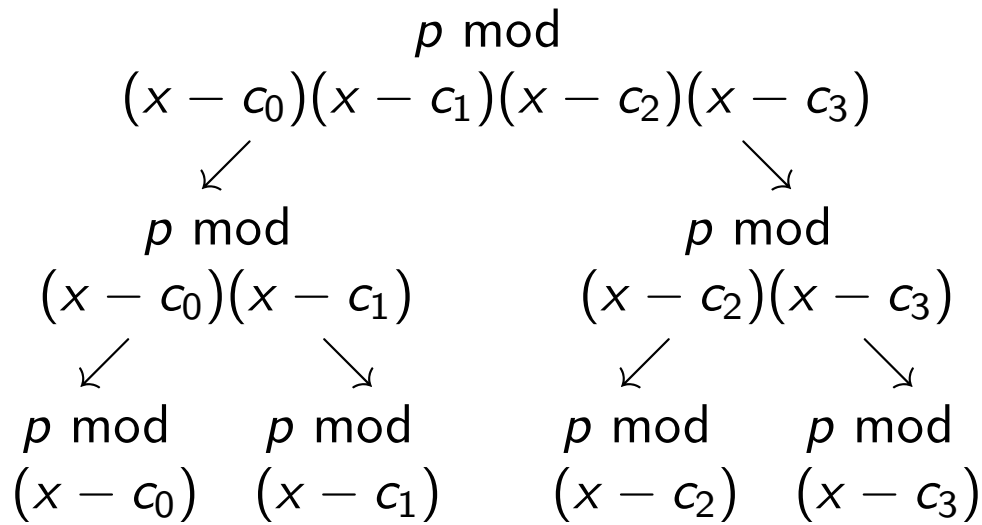


Assuming that the entries in $M(k)$ have bounded degree, this costs $O\tilde{}(n)$ arithmetic operations

Assuming that the entries in $M(k)$ have slowly growing coefficients (e.g. $O(\log k)$ bits), this costs $O\tilde{}(n^2)$ bit operations

Fast multipoint evaluation

A polynomial of degree n can be evaluated at n points using $O(\mathrm{M}(n) \log n) = O\tilde{}(n)$ arithmetic operations



Fast multipoint evaluation applied to sequences

Assume that $M(k)$ is a matrix of polynomials in k

Example: $M(k) = (k+1), \qquad n = 9$

$$P = (k+3)(k+2)(k+1) = k^3 + 6k^2 + 11k + 6$$

binary splitting

$$[P(6), P(3), P(0)] = [504, 120, 6]$$

fast multipoint evaluation

$$P(6)P(3)P(0) = 362880$$

repeated multiplication

This costs $O(\mathrm{M}(n^{1/2}) \log n) = O\tilde{}(n^{1/2})$ arithmetic operations

Application to primes

Wilson's theorem: an integer $n > 1$ is a prime iff $(n-1)! \equiv 1 \bmod n$.

This gives a terrible $O\tilde{}(n)$ primality test

Fast multipoint evaluation gives a no less terrible $O\tilde{}(n^{1/2})$ primality test

The same idea is used in Strassen's deterministic algorithm to factor an integer $N = pq$ with $p < q$ in time $O\tilde{}(N^{1/4})$.

Idea: let $m = \lfloor N^{1/2} \rfloor$. Then $\gcd(m! \bmod N, N) = p$

Fast evaluation of D-finite functions (2)

If $f(z)$ satisfies a linear differential equation with coefficients in $\mathbb{C}[z]$, then for any fixed $c \in \mathbb{C}$, we can compute $r \in \mathbb{Q}[i]$ with

$$|r - f(c)| < 2^{-p}$$

in $O^{\sim}(p^{1.5})$ bit operations [with some caveats].

Idea: analytic continuation + fast multipoint evaluation

Rectangular splitting

Assume that the recurrence matrix $M(k)$ contains polynomials of a parameter x

Polynomial coefficients (cheap): $c = 42$

Parameter (expensive): $x = 3.14159265358979323846264338327950288\ldots$

Distinguish between operations

Coefficient	GOOD	$c + c, \quad c \cdot c$
Scalar	OK	$x + x, \quad c \cdot x$
Nonscalar	BAD	$x \cdot x$

Rectangular splitting

Evaluate $\sum_{i=0}^{n} \square x^i$ using $O(n)$ scalar and $O(n^{1/2})$ nonscalar operations

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

Polynomials: Paterson and Stockmeyer, 1973
Composition of power series: Brent and Kung, 1978
Elementary / hypergeometric functions: Smith, 1989
Rising factorials (special case): Smith, 2001
General matrix polynomial recurrences: FJ, 2014
Optimized algorithm for elementary functions: FJ, 2015

The gamma function revisited

Argument reduction:

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

The Stirling series:

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

- For $s \to s + x \in \mathbb{C}[[x]]/\langle x^n \rangle$, binary splitting speeds up both steps
- For $s \in \mathbb{C}$, fast multipoint evaluation or rectangular splitting speeds up the argument reduction
- For $s \in \overline{\mathbb{Q}}$, binary splitting speeds up the argument reduction

Gamma function without Bernoulli numbers

$$\Gamma(s) \approx \int_0^N t^{s-1} e^{-t} dt \approx \frac{N^s e^{-N}}{s} \sum_{k=0}^{n} \frac{N^k}{(1+s)_k}$$

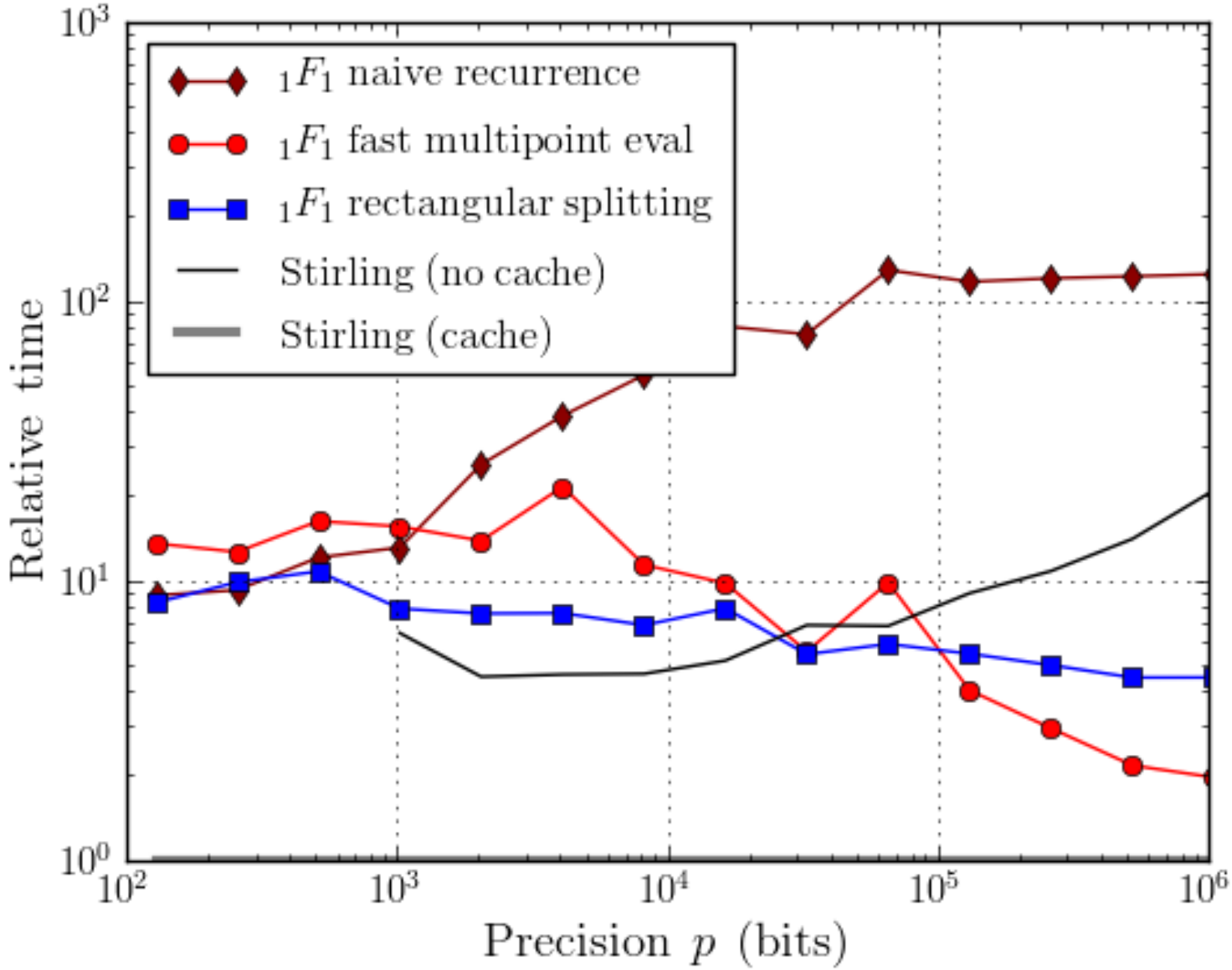
Partial sums satisfy order-2 recurrence with

$$M(k) = \frac{1}{1+k+s} \begin{pmatrix} 1+k+s & 1+k+s \\ 0 & N \end{pmatrix}$$

For p-bit precision: $n \sim N \sim p$

For $s \in \mathbb{C}$, bit complexity is $O\tilde{}(p^{1.5})$ with fast multipoint evaluation
For $s \in \overline{\mathbb{Q}}$, bit complexity is $O\tilde{}(p)$ with binary splitting

Improvement for the gamma function



The Hurwitz zeta function revisited

The power sum:

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

The tail:

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

- For $s \to s + x \in \mathbb{C}[[x]]/\langle x^n \rangle$, binary splitting speeds up the tail
- For $s \to s + x \in \mathbb{C}[[x]]/\langle x^n \rangle$, a *transposed version* of fast multipoint evaluation speeds up the power sum

Fast power series power sum

$$S = \sum_{k=0}^{N} \frac{1}{(a+k)^{s+x}} = \sum_{i=0}^{N} \left(\sum_{k=0}^{N} \frac{(-1)^i \log^i(a+k)}{i!(a+k)^s} \right) x^i$$

$$V = \begin{bmatrix} 1 & \log(a+0) & \cdots & \log^N(a+0) \\ 1 & \log(a+1) & \cdots & \log^N(a+1) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \log(a+N) & \cdots & \log^N(a+N) \end{bmatrix}$$

$$Y = \begin{bmatrix} (a+0)^{-s} & (a+1)^{-s} & \dots & (a+N)^{-s} \end{bmatrix}^T$$

- We want the vector $V^T Y$
- VY is *multipoint evaluation*: $\sum_i Y_i X^i$ at $X = \log(a), \dots, \log(a+N)$
- A fast ($O\tilde{}(N)$ arithmetic operations) algorithm for $V^T Y$ exists by the *transposition principle*

A few more references

- Borwein, Bradley & Crandall, *Computational strategies for the Riemann zeta function*, J. Comp. Appl. Math,, 2000
- Bernstein, *Fast multiplication and its applications*, `http://cr.yp.to/lineartime/multapps-20080515.pdf`
- Borwein & Borwein, *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*, Wiley, 1987
- Gourdon & Sebah, *Numbers, constants and computation*, `http://numbers.computation.free.fr/Constants/constants.html`
- Brent & Zimmermann, *Modern Computer Arithmetic*, CUP 2010, `http://www.loria.fr/~zimmerma/mca/pub226.html`
- FJ, *Fast and rigorous computation of special functions to high precision*, PhD thesis, 2014, `http://fredrikj.net/thesis/`

Zeta acceleration

Many slowly converging sums and products can be rewritten as more rapidly converging sums or products taken over zeta values

Example: the prime zeta function

$$P(s) = \sum_{p} \frac{1}{p^s}$$

$$\log(\zeta(s)) = \sum_{p \geq 2} -\log(1 - p^{-s}) = \sum_{p \geq 2} \sum_{k=1}^{\infty} \frac{p^{-ks}}{k} = \sum_{k=1}^{\infty} \frac{P(ks)}{k}$$

By Möbius inversion,

$$P(s) = \sum_{k=1}^{\infty} \mu(k) \frac{\log(\zeta(ks))}{k}$$

The twin prime constant

$$C_2 = \prod_{p \geq 3} \left(1 - \frac{1}{(p-1)^2}\right) = 0.66016\ldots$$

A zeta-accelerated representation is:

$$C_2 = \prod_{n=2}^{\infty} (\zeta(n)(1 - 2^{-n}))^{-I_n}, \qquad I_n = \frac{1}{n} \sum_{d|n} \mu(d) 2^{n/d}$$

Further results of this kind can be found in [Flajolet and Vardi, *Zeta function expansions of classical constants*, 1996 – `http://algo.inria.fr/flajolet/Publications/landau.ps`]

Khinchin's constant

For almost all real numbers x, the continued fraction coefficients

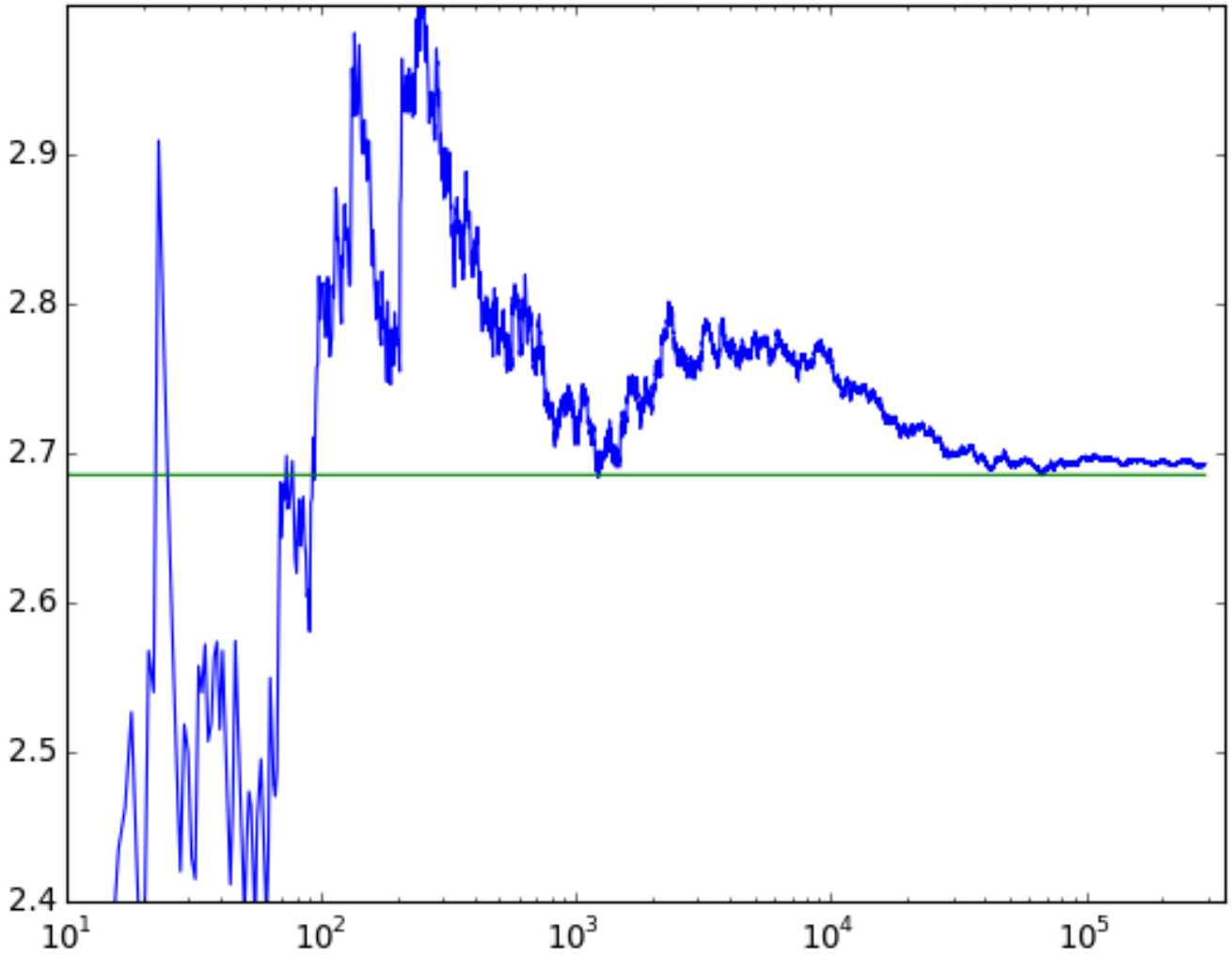
$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots}}}}$$

satisfy

$$\lim_{n\to\infty} (a_1 a_2 \ldots a_n)^{1/n} = \prod_{k=1}^{\infty} \left(1 + \frac{1}{k(k+2)}\right)^{\log_2(k)} \equiv K \approx 2.685452\ldots$$

A zeta-accelerated representation is:

$$\log(K) = \frac{1}{\log 2} \sum_{n=1}^{\infty} \frac{\zeta(2n) - 1}{n} \sum_{k=1}^{2n-1} \frac{(-1)^{k+1}}{k}$$



Convergence(?) to K for the ordinate 14.1347251417... of the first nontrivial zero of $\zeta(s)$