Implementing the Lagarias-Odlyzko Analytic Algorithm for $\pi(x)$

William F. Galway
Department of Mathematics
University of Illinois at Urbana-Champaign
Urbana, IL 61801
galway@math.uiuc.edu
http://www.math.uiuc.edu/~galway

Three ways to compute $\pi(x)$

- The sieve of Eratosthenes requires $O(x \ln \ln x)$ time and $O(\sqrt{x}/\ln x)$ space.

- The Meissel-Lehmer-Lagarias-Miller-Odlyzko algorithm [LMO85, DR96] requires $O(x^{2/3}/\ln^2 x)$ time and $O(x^{1/3+\epsilon})$ space. Uses sieving, clever use of recursive inclusion-exclusion, $\cdots$.

- The Lagarias-Odlyzko "analytic" algorithm [LO84, LO87] requires $O(x^{1/2+\epsilon})$ time and $O(x^{1/4+\epsilon})$ space. Uses numerical integration, $\cdots$.

Which will win?

To determine the expected "crossover point" at which the Meissel-Lehmer method becomes slower than the analytic method, we approximate the running times by

$$C_1 x^{a_1} \quad \text{for Meissel-Lehmer}$$

and

$$C_2 x^{a_2} \quad \text{for the analytic method.}$$

so the crossover point is

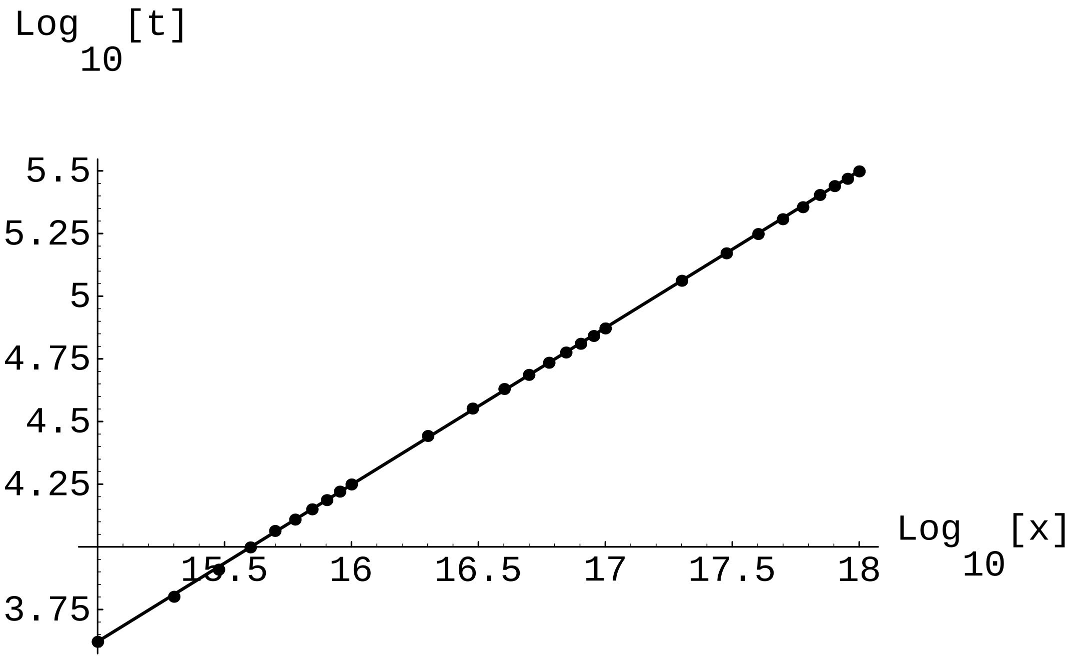
$$x = (C_2/C_1)^{1/(a_1 - a_2)}$$

Ignoring factors of x^ϵ we have $a_1 = 2/3$, $a_2 = 1/2$, and the crossover point is

$$x = (C_2/C_1)^6$$

but $\cdots$

Times for Meissel-Lehmer- ...



Log-Log Fit to Deleglise-Rivat timings

$$\log_{10}(t) = -5.768 + 0.626 \log_{10}(x)$$

(t in seconds.)

Given these experimental timing results perhaps a more accurate estimate is $a_1 = 0.625$ and $a_2 = 0.5$, in which case the crossover point is

$$x = (C_2/C_1)^8$$

$\cdots$ or $a_1 = 0.625$ and $a_2 = 0.525$, in which case the crossover point is

$$x = (C_2/C_1)^{10}$$

in any case, slight improvements in the implementation of either algorithm will greatly affect the crossover point.

The analytic algorithm – background

A "kernel" function $f(u)$ and its Mellin transform $F(s)$ are related by

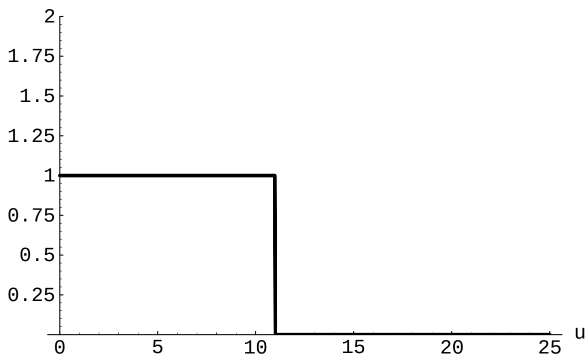
$$F(s) = \int_0^\infty f(u)u^{s-1}\,du$$

$$f(u) = \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} F(s)u^{-s}\,ds$$

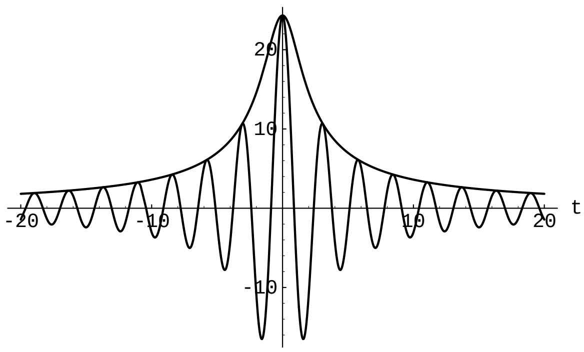
from which we see

$$\frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} F(s)\sum_{n\geq 1} a_n\, n^{-s}\,ds = \sum_{n\geq 1} a_n\, f(n)$$

A kernel function $f_1(u) = \chi_{[0,x]}(u)$



and its Mellin transform $F_1(s) = x^s/s$



From the Euler product formula for $\zeta(s)$ we have

$$\ln \zeta(s) = \sum_{n \geq 1} a_n \, n^{-s}$$

where

$$a_n = \begin{cases} 1/m & n = p^m \\ 0 & \text{otherwise} \end{cases}$$

and so (as noted by Riemann)

$$\begin{aligned} \pi^*(x) &= {\sum_{p^m \leq x}}' 1/m \\ &= \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} F_1(s) \ln \zeta(s) \, ds \\ &= \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \frac{x^s}{s} \ln \zeta(s) \, ds \end{aligned}$$

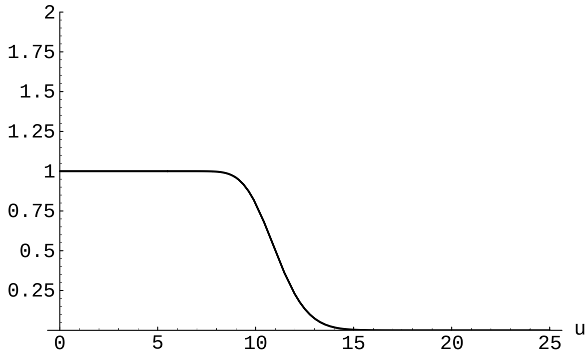
$\pi^*(x)$ is related to $\pi(x)$ by

$$\begin{aligned}
\pi^*(x) &= \sum_{m\geq 1}{}' \frac{1}{m}\pi(x^{1/m}) \\
\pi(x) &= \pi^*(x) - \sum_{m\geq 2} \frac{1}{m}\pi(x^{1/m}) \\
&= \sum_{m\geq 1} \frac{1}{m}\mu(m)\pi^*(x^{1/m})
\end{aligned}$$

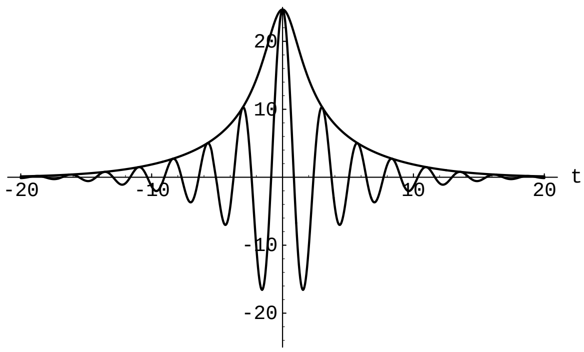
so $\pi(x)$ can easily be computed given a method for finding $\pi^*(x)$. Riemann's integral won't do since it converges far too slowly.

Getting a fast algorithm

Another kernel function $f_2(u)$

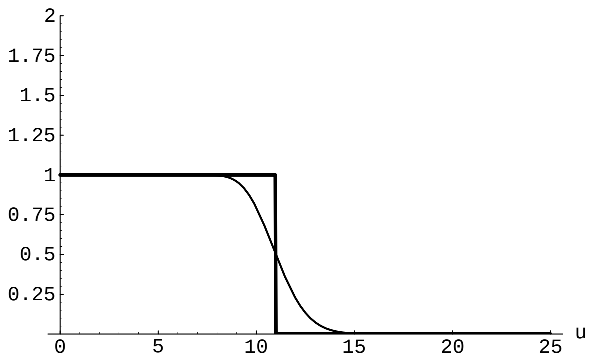


and its Mellin transform $F_2(s)$

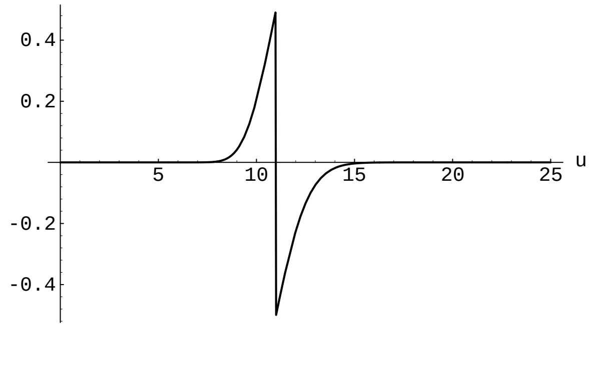


DRAFT Sun Apr 5 19:13:05 CDT 1998

Both kernel functions



and $f_1(u) - f_2(u)$



To compute $\pi^*(x)$ Lagarias and Odlyzko noted that

$$
\begin{aligned}
\pi^*(x) &= \sum_{n\geq 1} a_n\, f_1(n) \\
&= \sum_{n\geq 1} a_n\, f_2(n) + \sum_{n\geq 1} a_n\,(f_1(n) - f_2(n)) \\
&= \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} F_2(s)\ln\zeta(s)\,ds \\
&\quad + \sum_{n\geq 1} a_n\,(f_1(n) - f_2(n))
\end{aligned}
$$

the integral and the sum can be truncated — to $\int_{c-iT}^{c+iT}\cdots$ and $\sum_{n\in[u_0,u_1]}\cdots$ respectively. Ideally the kernel $f_2(u)$ should closely approximate a step function, with a Mellin transform $F_2(s)$ that damps out quickly as $t\to\pm\infty$, $t = \mathrm{Im}(s)$.

Things we need

For a practical implementation of the Lagarias-Odlyzko analytic algorithm we need:

- A good kernel function
- Optimal truncation points, based on good tail estimates
- A good quadrature algorithm for $\int_{c-iT}^{c+iT} \cdots$
- A fast and accurate way to compute $\zeta(s)$
- A fast primality test for

$$\sum_{n \in [u_0, u_1]} \cdots = \sum_{p^m \in [u_0, u_1]} \cdots$$

- Optimal choice of parameters (path of integration, relative width of truncation intervals, $\cdots$)
- Explicit bounds on all errors
- Computer code, $\cdots$.

Kernel function $\cdots$

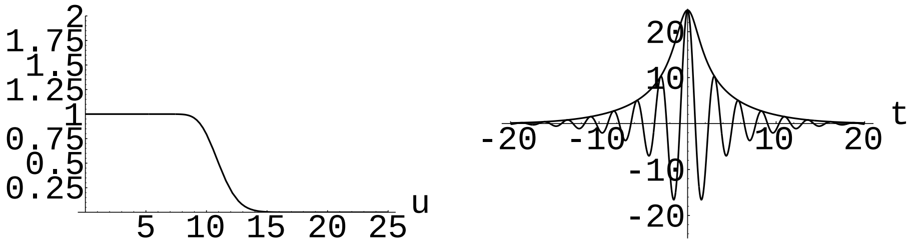
We use the Mellin transform pair

$$\begin{aligned} f_2(u) &= f_2(u; x, \lambda) \\ &= \frac{1}{2}\operatorname{erfc}\left(\frac{\ln(u/x)}{\sqrt{2}x^{\lambda-1}}\right) \\ F_2(s) &= F_2(s; x, \lambda) \\ &= \frac{x^s}{s}\exp\left(\frac{s^2}{2x^{2-2\lambda}}\right) \end{aligned}$$

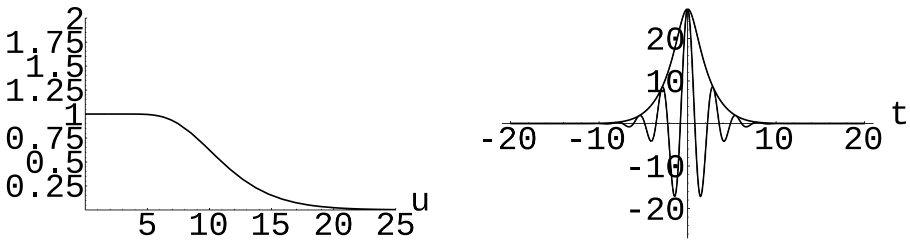
where $-\infty < \lambda < \infty$ is a parameter to be chosen later, and where

$$\operatorname{erfc}(z) = \frac{2}{\sqrt{\pi}}\int_z^\infty e^{-r^2}\,dr$$

$\cdots$ truncation points



$f_2(u; x, \lambda)$ and $F_2(s; x, \lambda)$
$x = 11$, $\lambda = 0.1$



$f_2(u; x, \lambda)$ and $F_2(s; x, \lambda)$
$x = 11$, $\lambda = 0.5$

With this pair, we may truncate at

$$\begin{aligned} u_1 - u_0 &= O(x^{\lambda}\sqrt{\ln x}) \\ T &= O(x^{1-\lambda}\sqrt{\ln x}) \end{aligned}$$

Quadrature algorithm

The trapezoidal rule works well, since $F_2(s)$ damps out quickly as $t =\!\!\longrightarrow \pm\infty$. Assuming that $1 < c < 2$, and using a stepsize h, the quadrature error E is of order

$$E = O(xe^{-2\pi(c-1)/h})$$

and we may use

$$h = O_E\left(\frac{c-1}{\ln x}\right)$$

to compute

$$\int_{c-iT}^{c+iT} F_2(s) \ln \zeta(s)\, ds\ .$$

Computing $\zeta(s)$

The method of Odlyzko and A. Schönhage [OS88] may be used to compute $\zeta(s)$ at points spaced regularly up to $s = c + iT$, using $O((\ln T)^{2+\epsilon})$ time per point, and $O(T^{1/2+\epsilon})$ space.

The Odlyzko-Schönhage algorithm depends on the Riemann-Siegel formula, for which we need carefully estimated error bounds. This has been done by Gabcke [Gab79] in the case $\mathrm{Re}(s) = 1/2$, but no similar bounds have been worked out (yet) for general s.

Primality testing

Sieving is very fast, but requires $O(\sqrt{x}/\ln x)$ space (too much).

Instead, we can use a partial sieve, eliminating "boring" multiples of primes $p \leq Y$, where Y grows slowly with x, perhaps $Y = O(x^{1/4})$.

The remaining numbers n are "Y-interesting". These n may be subjected to a probable primality test. If

$$2^{n-1} \equiv 1 \bmod n$$

then n is *very likely* to be prime. Finally, to preclude the chance that n is pseudoprime, we compare it against a pre-computed table of Y-interesting pseudoprimes in the interval of interest.

This method of finding primes appears to be very fast, is not probabilistic, but it appears very hard to prove anything about the time and space requirements.

DRAFT Sun Apr 5 19:13:05 CDT 1998

Choosing parameters

In computing $\int_{c-iT}^{c+iT} \cdots$, recall that a larger value of c allows a larger stepsize. However, the integrand is of order $O(x^c)$, so a larger c also requires greater precision $\cdots$.

To minimize the running time we must choose the parameter λ to balance the time spent in quadrature against the time spent computing the sum over prime powers $\cdots$.

References

[DR96] M. Deléglise and J. Rivat. Computing $\pi(x)$: the Meissel, Lehmer, Lagarias, Miller, Odlyzko method. *Mathematics of Computation*, 65(213):235–245, January 1996.

[Gab79] Wolfgang Gabcke. *Neue Herleitung und Explizite Restabschätzung der Riemann-Siegel-Formel*. PhD thesis, Georg-August-Universität zu Göttingen, 1979.

[LMO85] J. C. Lagarias, V. S. Miller, and A. M. Odlyzko. Computing $\pi(x)$: The Meissel-Lehmer method. *Mathematics of Computation*, 44(170):537–560, April 1985.

[LO84] J. C. Lagarias and A. M. Odlyzko. *New Algorithms for Computing $\pi(x)$*, pages 176–

193. Lecture Notes in Mathematics 1052. Springer-Verlag, New York, 1984.

[LO87] J. C. Lagarias and A. M. Odlyzko. Computing $\pi(x)$: an analytic method. *Journal of Algorithms*, 8:173–191, 1987.

[OS88] Andrew M. Odlyzko and A. Schönhage. Fast algorithms for multiple evaluations of the Riemann zeta function. *Trans. Amer. Math. Soc.*, 309:797–809, 1988.