数理解析研究所講究録
第1874巻 2014年 174-182

DETECTING SQUAREFREE NUMBERS

ANDREW R. BOOKER
(JOINT WITH GHAITH A. HIARY AND JON P. KEATING)

These notes are a transcript of my lecture at the conference on Analytic Number Theory at RIMS, describing my joint work with Hiary and Keating on detecting squarefree numbers. For a more formal account of our results, see our paper, `arXiv:1304.6937`.

Motivation. The problem of squarefree testing is no doubt a familiar one, but let us state it here for the record. Given a polynomial f, say with coefficients in some finite field $\mathbb{F}_q$, there is an easy test to determine whether f is squarefree:

$$f \in \mathbb{F}_q[t] \text{ is squarefree} \iff \gcd(f, f') = 1.$$

By way of analogy, one can ask the same question for the rings of integers of the other kind of global field, i.e. number fields, and in particular for ordinary integers. It is often lamented that we know no direct analogue of this fact when $\mathbb{F}_q[t]$ is replaced by $\mathbb{Z}$. Personally I do not feel that this is the right analogy to make in this case, since we also have very good factoring algorithms over $\mathbb{F}_q[t]$, ones that work provably in probabilistic polynomial time (which is arguably the most relevant measure of complexity in actual practice). That is to say that over the function field, the problem of squarefree testing is not very far removed (in terms of complexity) from that of factoring, so why should we expect that to be the case for the integers? Nevertheless, the question remains:

Can one determine whether $N \in \mathbb{Z}$ is squarefree more easily than by factoring it?

In this lecture, I will propose an algorithm for detecting squarefree integers based on L-functions. I make no promises in advance about the features of the algorithm, other than that it seems to be significantly different from the approaches that people have considered in the past.

What do we know already? (theoretical aspects). Before describing our method in detail, let me begin with some background on the problem of squarefree testing, starting with the theoretical aspects.

First, since it is easy to identify squares in $\mathbb{Z}$, to determine whether $N \in \mathbb{Z}_{>0}$ is squarefree it suffices to find all of its prime factors up to the cube root $\sqrt[3]{N}$. Next, there is an algorithm called Pollard–Strassen that finds all prime factors of a given N below a given bound B in time $O(N^{\varepsilon}\sqrt{B})$. Combining these two facts, one gets an algorithm for squarefree testing with running time $O(N^{1/6+\varepsilon})$. By the way, with some improvements to Pollard–Strassen specific to this problem, one can improve this very slightly to $O(N^{1/6-c/\log\log N})$ for some $c > 0$.

What do we know already? (practical aspects). Turning to more pragmatic matters, now that we have computers with very large memory (hundreds of gigabytes), it is possible to implement Pollard–Strassen and realize some gains in speed over trial division. Unfortunately, those gains do not kick in until B is of size at least 10^9, meaning that the basic algorithm described above is only practical for N up to 10^{70} or so. On the other

Date: October 31st, 2012.

hand, any $N \leq 10^{70}$ can, in practice, almost surely be factored with a smartphone in a few minutes. For larger N, finding the complete factorization remains the only viable option.

In conclusion, at least with present algorithms and technology, it is always better to try to factor N. One should not view this too negatively, since it is partly because our factoring algorithms have gotten so good. The best known algorithms (Elliptic Curve Method, Quadratic Sieve, Number Field Sieve, ...) are all *expected* to run in subexponential time $O(\exp[(\log N)^{\theta+o(1)}])$. ($\theta = 1/2$ for ECM and QS, $\theta = 1/3$ for NFS.) We generally cannot prove the running time estimates of these algorithms (although one can describe variants with provable subexponential probablistic running time under GRH). But that does not stop them from working well in practice, in recent years enabling the factorization of RSA moduli with over 200 digits.

Something a little different. Still, we can hold out hope of finding a better algorithm for squarefree testing. In order to give any algorithm that does not rely on factoring, it is clear that we first need a characterization of the squarefree numbers that does not mention their factorization. That is supplied by the following simple observation:

Given $d \in \mathbb{Z}$, $d \equiv 1 \pmod 4$, d is squarefree if and only if it is a fundamental discriminant.

Note that if $N \in \mathbb{Z}$ is odd then $d = (-1)^{\frac{N-1}{2}} N$ satisfies $d \equiv 1 \pmod 4$, so this restriction entails no loss of generality.

That this is a useful characterization rests on two key points. First, given such a d, whether or not d is a fundamental discriminant can be determined from values of the quadratic character

$$\chi_d(n) = \left(\frac{d}{n}\right),$$

(where the right-hand side is the Kronecker symbol) since its conductor will be unexpectedly small if d is not fundamental; I will elaborate on this point in the next section. Second (and also very simple but nonetheless important), we can compute $\chi_d(n)$ for any given n *without knowing the factorization of* d. Thus, there is no circular logic in what I have described (at least not so far). This is thanks to quadratic reciprocity, which is actually built in to the notation of the Kronecker symbol, since we think of it as a character mod d, even though we write the d on top. In particular, if n is a prime number then the Kronecker symbol reduces to the Legendre symbol, which can be evaluated quickly, e.g. by Euler's criterion.

Elaboration of the first key point. We can express d uniquely in the form $\Delta\ell^2$, where Δ is a fundamental discriminant and $\ell \in \mathbb{Z}_{>0}$. We aim to show that d is squarefree, i.e. $\ell = 1$. More generally, it suffices to show that ℓ is small, since at the end of the day we can apply Pollard–Strassen, or even just trial division, to rule out small square divisors of d.

To see how to use the values of the quadratic character to get information about its conductor, consider the following series

$$S_\Delta(x) = \frac{1}{\sqrt{x}} \sum_{n=1}^{\infty} \chi_\Delta(n) \left(\frac{n}{x}\right)^{(1-\chi_\Delta(-1))/2} e^{-\pi(n/x)^2}.$$

The alert reader will object that this expression depends on the unknown value of Δ. However, as in the second key point above, we can compute $\chi_\Delta(n)$ for a given n with

known factorization, even without knowledge of Δ—in fact we have $\chi_\Delta(n) = \chi_d(n)$ unless n has a common factor with ℓ. If we ever encouter such an n then we will have found a square factor of d, answering the original question. We might as well assume that that never happens.

Note that the sum is essentially the θ-function associated to χ_Δ, just with a different normalization of the variable. I like to think of it this way, since if one thinks of the $\chi_\Delta(n)$ as random ± 1s then, thanks to the decay of the Gaussian, the sum is the result of a random walk of length about x, roughly speaking. By the philosophy of the central limit theorem, we should typically expect such a sum to have size on the order of $\sqrt{x}$, so one might expect $S_\Delta(x)$ to oscillate randomly, without growing too large or decaying as $x \to \infty$.

This turns out to be an accurate description for $x \leq \sqrt{|\Delta|}$, but not after that. The reason is Poisson summation, which implies the symmetry

$$S_\Delta(x) = S_\Delta(|\Delta|/x).$$

Thus, whatever random behavior $S_\Delta(x)$ displays up to $\sqrt{|\Delta|}$, it is forced to do the same thing in reverse for larger values of x. (In particular, $S_\Delta(x)$ does decay for very large x, contrary to the above philosophy.) Now, the point is that this symmetry gives us an indication of $|\Delta|$. To prove that Δ is not too small, it suffices to graph $S_\Delta(x)$ and see that it does not "turn around" too early.

This hand-wavy argument can be completed to see that this gives another source of "square root running time", as in Pollard–Strassen, so it essentially reproduces the $O(N^{1/6+\varepsilon})$ time mentioned above. Moreover, if we had a method of computing the θ-function $S_\Delta(x)$ substantially more quickly than the obvious approach by direct, in-order summation, we could break the $1/6$ barrier in the exponent.

The explicit formula. Although that would be a great result (the $1/6$ has not been improved upon since the discovery of Pollard–Strassen over 40 years ago), it is still a far cry from the subexponential time expected of our best factoring algorithms. There is a simple reason for that, which has to do with the fact that we considered a sum over all positive integers n in the above. It is well-understood in problems of this type that one can do better by considering sums over primes, at the expense of having to assume the Generalized Riemann Hypothesis (GRH).

What I am talking about is the "explicit formula" for L-functions. To be precise, let us assume GRH and denote the non-trivial zeros of the Dirichlet L-function $L(s, \chi_\Delta) = \sum_{n=1}^\infty \chi_\Delta(n) n^{-s}$, by $\frac{1}{2} \pm i\gamma_j(\Delta)$ $(j = 1, 2, \ldots)$, where $0 \leq \gamma_1(\Delta) \leq \gamma_2(\Delta) \leq \ldots$. Further, let $g : [0, \infty) \to \mathbb{C}$ be a test function which is continuous of compact support, piecewise smooth, and has Fourier cosine transform $h(t) = 2\int_0^\infty g(x)\cos(tx)\,dx$. Then the explicit formula is the following distributional identity:

$$g(0)\log|\Delta| = 2\sum_{j=1}^\infty h(\gamma_j) + 2\sum_{n=1}^\infty \frac{\Lambda(n)\chi_\Delta(n)}{\sqrt{n}} g(\log n)$$
$$+ g(0)\log(8\pi e^\gamma) - \int_0^\infty \frac{g(0) - g(x)}{2\sinh(x/2)}\,dx + \chi_\Delta(-1)\int_0^\infty \frac{g(x)}{2\cosh(x/2)}\,dx.$$

Here Λ is the von Mangoldt function, so the sum over n is supported on prime powers. The integrals on the second line give, philosophically, the contribution from the "archimedean prime".

What the explicit formula is a formula for is in the eye of the beholder. To Riemann it was a way of getting information about the primes in terms of the zeros of the ζ-function. More recently, people have turned the tables and tried to study statistical properties of zeros using our knowledge of primes. In the formula as presented above it is neither; instead it is a formula for the *conductor* (or rather its logarithm) in terms of primes and zeros. Note that if not for the zero sum, this would be precisely what we seek, i.e. a formula for the conductor in terms of character values. With the zeros in there, we do not get such a nice identity, but let us see what we can do with it anyway.

Using the explicit formula to bound $|\Delta|$ from below. All terms in the explicit formula are easily computed apart from the zeros γ_j. Actually, they can be computed as well, but our methods of doing so rely on the standard technology of sums in additive form, meaning that they would be much too slow to be of any use in our current application.

However, for the purposes of a *lower bound* for $\log|\Delta|$ (remember that's what we want, to show that Δ is close d), we can simply ignore the problem, i.e. choose g so that h is non-negative and ignore the zero contribution:

$$\log|\Delta| \geq 2\sum_{n=1}^{\infty}\frac{\Lambda(n)\chi_\Delta(n)}{\sqrt{n}}g(\log n) + \log(8\pi e^\gamma) - \int_0^\infty \frac{1-g(x)}{2\sinh(x/2)}\,dx + \chi_\Delta(-1)\int_0^\infty \frac{g(x)}{2\cosh(x/2)}\,dx.$$

for any g with $g(0) = 1$ and non-negative cosine transform h.

Naturally, we pay a price for ignoring the zero sum $S = \sum h(\gamma_j)$, in that our estimate for Δ is a factor of e^{2S} too small. However, we can remedy this by applying Pollard–Strassen to rule out ℓ up to e^S, but that takes exponential time $\gg e^{S/2}$ in the size of S. On the other hand, we also face an exponential penalty for increasing the support of the test function (recall that the explicit formula involves $g(\log n)$). If the support of g is too small then, by the uncertainty principle, the zero sum will typically be large.

Our first result is that there is a "best" test function to use for each X, and thus an optimal tradeoff between these two exponential penalties:

Theorem. *Let $\mathcal{C}(X)$ be the class of functions $g : [0,\infty) \to \mathbb{R}$ that are continuous, supported on $[0,X]$, have non-negative cosine transform h, and satisfy $g(0) = 1$. For $g \in \mathcal{C}(X)$, let $\Lambda(g)$ denote our lower bound. Then for any $X > 0$, $\exists g_X \in \mathcal{C}(X)$ such that $\Lambda(g_X) \geq \Lambda(g)\ \forall g \in \mathcal{C}(X)$.*

This is only an existence result. To actually find a good g in practice, we let $g = f * f$ where f has many parameters, e.g.

$$f(x) = \sum_{n=-M}^{M} a_n \mathbf{1}_{(-1/2,1/2)}\left(\frac{2M+1}{X}x - n\right), \quad a_{-n} = a_n \in \mathbb{R}.$$

(This is perhaps better stated in words: f is a step function, with steps on a fine grid of points between $-X/2$ and $X/2$, and with the height of the steps treated as variables.) Then $\sum h(\gamma_j)$ is a positive-definite quadratic form in the parameters a_n, and we can compute the whole matrix of the form directly from the explicit formula. The condition $g(0) = 1$ amounts to an L^2 normalization, so we are asking for the *smallest eigenvalue* of the quadratic form, which is readily computed using standard software for linear algebra.

It is not hard to see that this family of test functions comes arbitrarily close to the optimal g as $M \to \infty$. However, the optimal g might be highly oscillatory, meaning that we would have to take M very large.

Twisting. Another strategy is to "twist" our given quadratic character χ_d by χ_q for a known fundamental discriminant q. In other words, if we run out of luck with our given d then we can multiply by q and ask if the product is fundamental. This is related to the first strategy since, by Fourier analysis, varying the test function amounts to considering combinations of the twists by n^{it} for various t.

On the surface this might seem like a bad idea, since we are making the number large by multiplying by q; in fact to get a bound for $\log|\Delta|$ rather than $\log|q\Delta|$, we have to subtract $\log|q|$:

$$\log|\Delta| \geq 2\sum_{n=1}^{\infty} \frac{\Lambda(n)\chi_{q\Delta}(n)}{\sqrt{n}} g(\log n) + \log\left|\frac{8\pi e^{\gamma}}{q}\right| - \int_0^{\infty} \frac{1-g(x)}{2\sinh(x/2)}\,dx + \chi_{q\Delta}(-1)\int_0^{\infty} \frac{g(x)}{2\cosh(x/2)}\,dx.$$

However, we do gain something by this approach, as we discuss next.

When is this likely to work? Either approach (varying the test function or using twists) has two opposing goals: We want $\sum h(\gamma_j(q\Delta))$ to be small, but with the support of g also not too large. Typically (for carelessly chosen g of small support), the zero sum will dominate the right-hand side of the explicit formula, since the left-hand side just reflects the average zero density and, by the uncertainty principle, h must be relatively "wide".

However, if there is a region where the zeros are unusually sparse then we can hope to achieve both goals. What we are relying on is the (empirical) observation that the zeros do not rigidly line up according to their average density; sometimes they are much closer together than average and sometimes much farther apart. The simplest idea is to look for a large gap between zeros:

Theorem. *Suppose that $L(s,\chi_{q\Delta})$ has no zeros with imaginary part in $(-\delta,\delta)$. Set $X = 2\delta^{-1}(A + \log\log|q\Delta|)$ for some $A \geq 0$. Then there is an explicit $g \in C(X)$ whose cosine transform h satisfies*

$$\sum_{j=1}^{\infty} h(\gamma_j(q\Delta)) \ll \frac{e^{-A}X}{(\log\log|q\Delta|)^{3/2}}.$$

In other words, there is an explicit test with support of size inversely proportional to the size of the zero gap for which the zero sum is relatively small. (Note that it is enough for the zero sum to be less than X, since then the work done in computing the terms of the explicit formula for all primes up to e^X is already enough to rule out small values of ℓ and complete the proof that d is squarefree.) The parameter A is present to show that one can defeat even a lazily-derived implied constant at the expense of only a modest increase in the size of X.

The proposition also suggests another strategy for finding good test functions: Fix a choice of h, anticipating a gap of a certain size, and twist by various choices of q. Once a suitable q has been found, we can refine our choice of test function using the quadratic form approach with a relatively small matrix.

Zero gaps and running time. The previous theorem tells us that it is sufficient to find a large zero gap. To make this precise and to understand how quickly we can expect the method to work, we define

$$M_\Delta(\theta) = \max\Big\{\gamma_1(q\Delta) : q \text{ fund. disc.}, |q| \le \exp\big((\log|\Delta|)^\theta\big)\Big\},$$
$$\eta_\Delta(\theta) = -\frac{\log M_\Delta(\theta)}{\log\log|\Delta|}, \quad \eta_\infty(\theta) = \limsup_{|\Delta|\to\infty} \eta_\Delta(\theta),$$
$$\theta^* = \inf\{\theta > 0 : \eta_\infty(\theta) \le \theta\}.$$

In other words, we consider the largest zero gap that we can find among the twists by all values of q up to $e^{(\log|\Delta|)^\theta}$ (anticipating a subexponential running time comparable to that of the factoring algorithms), and renormalize it on a logarthmic scale. Recall that the average gap between zeros is proportional to $1/\log|\Delta|$, which corresponds to $\eta = 1$. We might think of a large gap as having size on the order of $1/\sqrt{\log|\Delta|}$, say, which would correspond to $\eta = 1/2$. Finally, since our only information about $|\Delta|$ *a priori* is that it is large, we consider the worst case by taking the lim sup. Then our precise result is the following:

Theorem. *Assume GRH for quadratic Dirichlet L-functions. There is an algorithm that takes as input a positive integer N and outputs either a non-trivial square factor of N or a proof that N is squarefree. If N is squarefree then the algorithm runs in time $O\big(\exp[(\log N)^{\theta^*+o(1)}]\big)$.*

This looks great of course. It is a proof of deterministic subexponential running time, but there is a catch—the result is only non-trivial if it turns out that $\theta^* < 1$. That is probably true (more on that in a bit), but seems to be beyond our technology to prove at present, even assuming the GRH. (We can prove that $\theta^* \in [0,1]$ by a simple computation of the 1-level density.)

Random matrix theory. This does not stop us from making a reasonable conjecture of the value of θ^*. The tool of choice for this is *random matrix theory*, which we explain as follows.

Any $N \times N$ unitary matrix A is conjugate to a unique matrix of the form

$$\begin{pmatrix} e^{i\theta_1} & & \\ & \ddots & \\ & & e^{i\theta_N} \end{pmatrix},$$

where $0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_N < 2\pi$. The angles $\theta_1, \dots, \theta_N$ are called the *eigenphases* of A. On the other hand, assuming GRH, an interval of length 2π along the critical line of any L-function contains roughly log(conductor) zeros, which one may also regard as eigenphases of some unitary matrix. The philosophy of random matrix theory asserts that properly phrased statistical questions regarding zeros of families of L-functions can be modeled by analogous questions about matrices chosen randomly from an ensemble (such as $U(N)$, $Sp(2N)$, $O(2N)$, ...) according to Haar measure.

There is by now a lot of empirical evidence for this philosophy. I think it is fair to say that nobody really has as inkling for *why* it is true, though as noted above, that does not stop us from putting it to good use. For the particular family of quadratic twists, it is expected (from comparison of 1-level densities) that the right ensemble to consider is symplectic, i.e. $Sp(2N)$, with $2N \approx \log|\Delta|$. To conjecture a value for θ^*, we have the following result:

Theorem. *Fix $\beta \in (0,2)$, and define*

$$M_\beta(N) := \lfloor \exp((2N)^\beta) \rfloor, \quad s^\pm_{\varepsilon,\beta}(N) := (4 \pm \varepsilon)(2N)^{\beta/2-1}.$$

Let $A_1, \dots, A_{M_\beta(N)} \in Sp(2N)$ be chosen independently and uniformly with respect to Haar measure. Then, for fixed $\varepsilon > 0$, as $N \to \infty$ we have

$$\mathbb{P}_{USp(2N)}\left(s^-_{\varepsilon,\beta}(N) < \max_{1 \le m \le M_\beta(N)} \theta_1(A_m) \le s^+_{\varepsilon,\beta}(N) \right) \to 1.$$

In other words, $(2N)^{1-\beta/2} \max_{1 \le m \le M_\beta(N)} \theta_1(A_m)$ converges in distribution to 4.

This *suggests* that $\eta_\infty(\theta) = 1 - \theta/2$ for $\theta \in [0,1]$ (for $\theta > 1$, the q-aspect takes over, so presumably $\eta_\infty(\theta)$ is constant after that), leading to the conjecture $\theta^* = 2/3$.

More general twists. We considered twists of $L(s, \chi_\Delta)$ by χ_q. More generally, we may take any family of automorphic L-functions and consider their twists by the given character χ_Δ. The explicit formula for the twisted family may again be used to produce a lower bound for $\log|\Delta|$. Natural examples of twist families include:

- Elliptic curve L-functions. Can we take advantage of the occurrence of high-order zeros (under BSD) to force zero repulsion?
- Dedekind ζ-functions. Can we make use of the existence of towers of number fields of bounded root discriminant?
- Rankin–Selberg L-functions. Can we make use of the algebraic structure of the coefficients of L-functions implied by the Langlands conjectures to find correlating twists quickly?

All of these are ideas for future investigations. Nevertheless, it would be a fair criticism at this point that the method seems to do worse than existing factoring algorithms, since $2/3$ is larger than the expected complexity exponents for factoring. There are a couple of saving graces. First, our method at least has the virtue of relying on a different set of heuristics (random matrix theory as opposed to distributional questions about smooth numbers), and it is reasonable to hope that we will one day be able to substantiate enough of those heuristics to give a proof of determinstic subexponential running time. Second, and more tangibly, our method gives partial information that one does not obtain from a failed attempt at factoring, as illustrated in the following.

Sample numerical result.

Theorem. *Assume GRH for quadratic Dirichlet L-functions. Then*

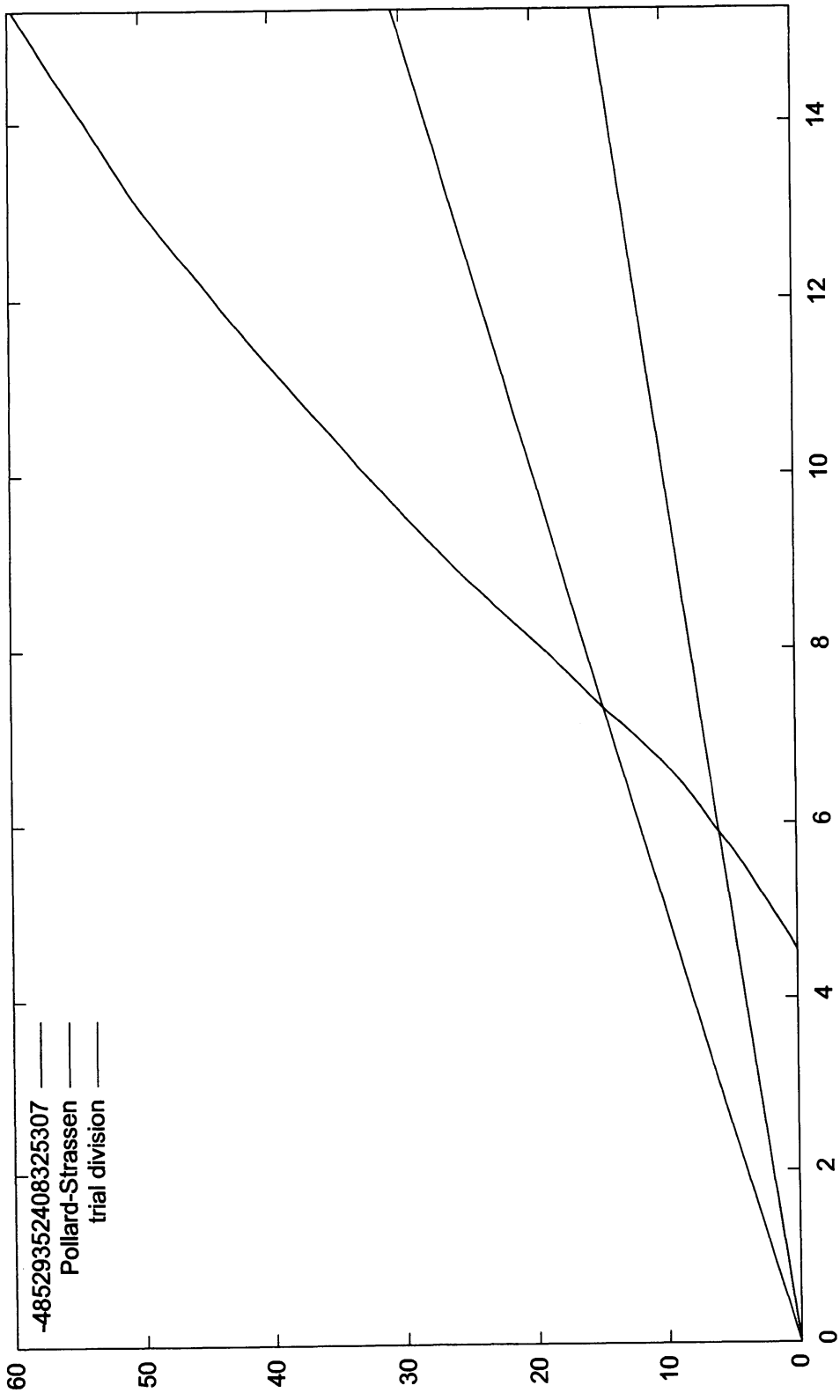
$$\begin{aligned} \texttt{RSA-210} = {} & 2452466449002782119765176635730880184670267878783327597434144517150616008300385872169522083993320715491036268271916798640797767232430056005920356312465612184658179041001318592996199338170121493350348758705510 67 \end{aligned}$$

is not squarefull, i.e. it has at least one simple prime factor.

The challenge number `RSA-210` is significant because, at 210 digits, it is the smallest that has not yet been factored. Certainly the technology to factor it exists, and in fact the slightly longer `RSA-704` (212 digits) and `RSA-768` (232 digits) were factored in 2012 and 2009, respectively. However, it remains prohibitively expensive to perform such factorizations routinely. By constrast, the proof of the theorem above could be obtained

using a desktop PC in a couple of months. To my knowledge, this is the first statement to be proven about the factorization of a composite integer without exhibiting any of its factors. (The alert reader might protest that, by construction, RSA-210 must be a product of two distinct primes and therefore squarefree. The point is that we have no proof of that fact, since its prime factors were never recorded.)

The proof of the theorem is contained in the following graph, which shows our lower bound for $|\Delta|$ (i.e. RSA-210 divided by its largest square factor) as a function of the size of the prime sum, for the particular twist $q = -933460208865458027728 3$. The graph is on a log-log scale, i.e. using the primes up to 10^x in the explicit formula, we get a lower bound of 10^y. Also shown is the lower bound that one obtains from trial division using the same primes (we checked that RSA-210 is not a perfect square, so it must have at least one prime factor of odd multiplicity; ruling out small prime factors thus gives the lower bound $y = x$) and that obtained from a theoretically perfect implementation of Pollard–Strassen (which runs in square root time, corresponding to the bound $y = 2x$; in reality, practical implementations of Pollard–Strassen only surpass trial division by a small margin near the end of the graph). Although for a fixed choice of q our method initially does worse (since we make things harder when multiplying by q), in the end we get nearly a four-fold improvement in the lower bound (i.e. 10^{60} as opposed to 10^{15}).


-48529352408325307
Pollard-Strassen
trial division
0
10
20
30
40
50
60
0
2
4
6
8
10
12
14