The number-field sieve

Finding small factors of integers

Speed of the number-field sieve

D. J. Bernstein
University of Illinois at Chicago

Prelude: finding denominators

$817/366 \approx 2.23224044$ in $\mathbf{R}$.

Easily compute digits 2.23224044 given $817, 366$.

Can we work backwards: find $817, 366$ given digits 2.23224044?
"2-dim integer-relation finding";
"2-dim lattice-basis reduction";
"half-gcd computation"; etc.

Yes, via continued fractions.

Compute successively

$1/(2.23224044 - 2) \approx 4.3058823;$

$1/(4.3058823 - 4) \approx 3.269231;$

$1/(3.269231 - 3) \approx 3.71428;$

$1/(3.71428 - 3) \approx 1.4000;$

$1/(1.4000 - 1) \approx 2.500;$

$1/(2.500 - 2) \approx 2.00 \approx 2.$

Evidently 2.23224044 is very close to the continued fraction

$$2 + \cfrac{1}{4 + \cfrac{1}{3+\cfrac{1}{3+\cfrac{1}{1+\cfrac{1}{2+\frac{1}{2}}}}}} = \frac{817}{366}.$$

Can obtain y-digit numerator
and y-digit denominator
from $2y$ digits of quotient.

$y(\lg y)^{O(1)}$ bit operations
using fast multiplication,
fast continued fractions.

Analogous polynomial algorithms
find two y-coefficient polynomials
from $2y$ coefficients of their
power-series quotient.

$y(\lg y)^{O(1)}$ coefficient operations
using fast algorithms.

Linear algebra

$y \times y$ matrix M over $\mathbf{F}_2$ specifies linear map $\mathbf{F}_2^y \to \mathbf{F}_2^y$.

e.g. $M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$

specifies $(v_1, v_2, v_3, v_4) \mapsto$

$(0v_1 + 1v_2 + 1v_3 + 1v_4,$
$\ 0v_1 + 1v_2 + 1v_3 + 0v_4,$
$\ 1v_1 + 1v_2 + 1v_3 + 0v_4,$
$\ 1v_1 + 0v_2 + 1v_3 + 1v_4).$

Subroutine in $\mathbf{Q}$ sieve etc., combining smooth congruences to form a square:

"Find linear dependency" $=$ "find nonzero kernel element" $=$ "find nonzero nullspace element": find nonzero $v \in \mathbf{F}_2^y$ with $Mv = 0$.

e.g. previous $M(v_1, v_2, v_3, v_4)$ is 0 only if $(v_1, v_2, v_3, v_4) = 0$, so can't find linear dependency.

"Solve linear equations":
given $w \in \mathbf{F}_2^y$,
find *some* $v \in \mathbf{F}_2^y$ with $Mv = w$.

e.g. given $w = (1, 1, 0, 0)$ and

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}:$$

find (v_1, v_2, v_3, v_4) with

$$\begin{aligned}(&0v_1 + 1v_2 + 1v_3 + 1v_4,\\ &0v_1 + 1v_2 + 1v_3 + 0v_4,\\ &1v_1 + 1v_2 + 1v_3 + 0v_4,\\ &1v_1 + 0v_2 + 1v_3 + 1v_4) = w.\end{aligned}$$

We have fast methods
to solve linear equations.

Easily apply those methods
to find linear dependencies,
if any dependencies exist.

Choose uniform random $r \in \mathbf{F}_2^y$;
compute $w = Mr$;
use linear-equation solver
to find v with $Mv = w$.
This produces uniform random
kernel element, namely $v - r$.
Try again if $v = r$.

"Elimination"
solves linear equations
using $O(y^3)$ bit operations.

"Series denominators"
solve linear equations
using $y^{2+o(1)}$ bit operations
if the equations are sparse.

"Sparse": can evaluate $v \mapsto Mv$
using $y^{1+o(1)}$ bit operations.
Certainly true in $\mathbf{Q}$ sieve
with usual choices of y.

Series denominators

e.g. Given $w = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ and

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}:$$

Have magic equation
$w + M^3 w + M^4 w = 0$ implying
$w = Mv$ for $v = -M^2 w - M^3 w$.

How did I find magic equation?
First explore its consequences.

Consider the power series

$$S = w + (Mw)t + (M^2w)t^2 + \cdots =$$

$$\begin{pmatrix}1\\1\\0\\0\end{pmatrix} + \begin{pmatrix}1\\1\\0\\1\end{pmatrix} t + \begin{pmatrix}0\\1\\0\\0\end{pmatrix} t^2 +$$

$$\begin{pmatrix}1\\1\\1\\0\end{pmatrix} t^3 + \begin{pmatrix}0\\0\\1\\0\end{pmatrix} t^4 + \begin{pmatrix}1\\1\\1\\1\end{pmatrix} t^5 +$$

$$\begin{pmatrix}1\\0\\1\\1\end{pmatrix} t^6 + \begin{pmatrix}0\\1\\0\\1\end{pmatrix} t^7 + \cdots \text{ in } \mathbf{F}_2^4[[t]].$$

S is rational:

$$S(1+t+t^4) = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} t +$$

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} t^2 + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} t^3.$$

For $n \geq 4$, coefficient of t^n in $(\sum_{i \geq 0} M^i w t^i)(1+t+t^4)$ is $M^n w + M^{n-1} w + M^{n-4} w$ $= M^{n-4}(M^4 w + M^3 w + w) = 0$ by magic equation.

Squeeze S by projecting from $\mathbf{F}_2^4[[t]]$ to $\mathbf{F}_2[[t]]$.

e.g. Define $r = (0\ 0\ 0\ 1)$.

$$rS = rw + rMwt + rM^2wt^2 + \cdots$$
$$= t + t^5 + t^6 + t^7 + t^8 + t^{10} + \cdots.$$

Have $rS(1 + t + t^4) = t + t^2$.

Similar for every $r : \mathbf{F}_2^4 \to \mathbf{F}_2$.
The series $rS \in \mathbf{F}_2[[t]]$ is rational, specifically a poly of degree < 4 divided by $1 + t + t^4$.

Can use continued fractions to quickly find denominator $1 + t + t^4$, and thus to find magic equation.

In general, given $w \in \mathbf{F}_2^y$
and $M : \mathbf{F}_2^y \to \mathbf{F}_2^y$,
find magic equation as follows.

Pick $r : \mathbf{F}_2^y \to \mathbf{F}_2$.
Compute first $2y$ terms of series
$rw + rMwt + rM^2wt^2 + \cdots$ in
$\mathbf{F}_2[[t]]$. Use continued fractions
to find denominator of series.

Repeat for a few random r's,
compute lcm of denominators.
With very high probability
obtain denominator of series
$w + Mwt + M^2wt^2 + \cdots$.

If final denominator is
$p_0 t^y + p_1 t^{y-1} + \cdots + p_y t^0$ then
$p_0 w + p_1 M w + \cdots + p_y M^y w = 0$.

If $p_0 = 1$ then $w = Mv$ where
$v = -p_1 w - \cdots - p_y M^{y-1} w$.

If $p_0 = 0$ then use slightly more complicated algorithm to solve linear equation. But still easy to find linear dependency.

Overall there are
$O(y)$ applications of M.
Total $y^{2+o(1)}$ bit operations
if M is sparse.

Asymptotic cost exponents

Number of bit operations
in number-field sieve,
with theorists' parameters,
is $L^{1.90\ldots+o(1)}$ where $L =$
$\exp((\log n)^{1/3}(\log\log n)^{2/3})$.

What are theorists' parameters?

Choose degree d with
$d/(\log n)^{1/3}(\log\log n)^{-1/3}$
$\in 1.40\ldots+o(1)$.

Choose integer $m \approx n^{1/d}$.
Write n as
$m^d + f_{d-1}m^{d-1} + \cdots + f_1 m + f_0$
with each f_k below $n^{(1+o(1))/d}$.
Choose f with some randomness
in case there are bad f's.

Test smoothness of $i - jm$
for all coprime pairs (i, j)
with $1 \le i, j \le L^{0.95\ldots+o(1)}$,
using primes $\le L^{0.95\ldots+o(1)}$.

$L^{1.90\ldots+o(1)}$ pairs.
Conjecturally $L^{1.65\ldots+o(1)}$
smooth values of $i - jm$.

Use $L^{0.12\ldots+o(1)}$ number fields.

For each (i, j)
with smooth $i - jm$,
test smoothness of $i - j\alpha$
and $i - j\beta$ and so on,
using primes $\le L^{0.82\ldots+o(1)}$.

$L^{1.77\ldots+o(1)}$ tests.
Each $|j^d f(i/j)| \le m^{2.86\ldots+o(1)}$.
Conjecturally $L^{0.95\ldots+o(1)}$
smooth congruences.

$L^{0.95\ldots+o(1)}$ components
in the exponent vectors.

Three sizes of numbers here:

$(\log n)^{1/3}(\log\log n)^{2/3}$ bits:
y, i, j.

$(\log n)^{2/3}(\log\log n)^{1/3}$ bits:
m, $i - jm$, $j^d f(i/j)$.

$\log n$ bits: n.

Unavoidably $1/3$ in exponent:
usual smoothness optimization
forces $(\log y)^2 \approx \log m$;
balancing norms with m
forces $d \log y \approx \log m$;
and $d \log m \approx \log n$.

The number-field sieve
is asymptotically much faster
than the quadratic sieve
and the elliptic-curve method.

Also works well in practice.

Latest record: NFS found
two prime factors $\approx 2^{332}$
of "RSA-200" challenge, using
$\approx 5 \cdot 10^{18}$ Opteron cycles.

Batch NFS

The number-field sieve used $L^{1.90\ldots+o(1)}$ bit operations finding smooth $i - jm$; only $L^{1.77\ldots+o(1)}$ bit operations finding smooth $j^d f(i/j)$.

Many n's can share one m; $L^{1.90\ldots+o(1)}$ bit operations to find squares for *all* n's.

Oops, linear algebra hurts; fix by reducing y.
But still end up factoring batch in much less time than factoring each n separately.

Polynomial selection

Many choices of NFS polynomial.
Which choices are best?

Consider, e.g., poly degree $d = 5$.
Select integer $m \in [n^{1/6}, n^{1/5}]$;
find integers $f_5, f_4, \ldots, f_0$
with $n = f_5 m^5 + f_4 m^4 + \cdots + f_0$;
for various integers i, j inspect
$(i - jm)(f_5 i^5 + f_4 i^4 j + \cdots + f_0 j^5)$.

Practically every choice of m
will succeed in factoring n.
For speed want smallest possible
$(i - jm)(f_5 i^5 + f_4 i^4 j + \cdots + f_0 j^5)$.

e.g. $n = 314159265358979323$:
Can choose $m = 1000$,
$f_5 = 314$, $f_4 = 159$, $f_3 = 265$,
$f_2 = 358$, $f_1 = 979$, $f_0 = 323$.

NFS succeeds in factoring n
by inspecting congruences
$(i - 1000j)(314i^5 + \cdots + 323j^5)$
for various integer pairs (i, j).

But NFS succeeds more quickly
using $m = 1370$, inspecting
$(i - 1370j)(65i^5 + 130i^4 j +$
$38i^3 j^2 + 377i^2 j^3 + 127ij^4 + 33j^5)$.

Consider, e.g.,

2^{45} possible choices of m.

Quickly identify, e.g.,

2^{25} attractive candidates.

Will choose one m later.

If $|i| \le SR$ and $|j| \le S^{-1}R$ then $\left|(i - jm)(f_5 i^5 + \cdots + f_0 j^5)\right| \le \mu(m, S)R^6$ where $\mu(m, S) = (mS^{-1}+S)(\left|f_5 S^5\right| + \cdots + \left|f_0 S^{-5}\right|)$.

Attractive m, S: small $\mu(m, S)$.

Choosing one typical $m \approx n^{1/6}$ produces $\mu(m, 1) \approx n^{2/6}$.

Question: How much time do we need to save factor of B—to find m, S with $\mu(m, S) \approx B^{-1} n^{2/6}$?

This has as much impact as chopping $\approx 3 \lg B$ bits out of n.

Searching for good values of m takes noticeable fraction of total time of optimized NFS.
(If not, consider more m's!)
End up with rather large B.

Conjectured time $B^{7.5+o(1)}$:
Enumerate many possibilities
for m near $B^{0.25}n^{1/6}$.

Have $f_5 \approx B^{-1.25}n^{1/6}$.
f_4, f_3, f_2, f_1, f_0 could be
as large as $B^{0.25}n^{1/6}$.
Hope that they are smaller,
on scale of $B^{-1.25}n^{1/6}$,
so $\mu(m, 1) \approx B^{-1}n^{2/6}$.

Conjecturally this happens
within roughly $B^{7.5}$ trials.

Conjectured time $B^{6+o(1)}$:
Skip through m's with small f_4.

Say $n = f_5 m^5 + f_4 m^4 + \cdots + f_0$.

Choose integer $k \approx f_4/5f_5$.
Write n in base $m + k$:

$$n = f_5(m+k)^5 + (f_4 - 5kf_5)(m+k)^4 + \cdots.$$

Now degree-4 coefficient
is on same scale as f_5.

Hope for small f_3, f_2, f_1, f_0.

Conjectured time $B^{4.5+o(1)}$:
Increase S.

Enumerate many possibilities for m near $Bn^{1/6}$.

Have $f_5 \approx B^{-5}n^{1/6}$.
f_4, f_3, f_2, f_1, f_0 could be as large as $Bn^{1/6}$.

Force small f_4. Hope for f_3 on scale of $B^{-2}n^{1/6}$, f_2 on scale of $B^{-0.5}n^{1/6}$.
Then $\mu(m, B^{0.75}) \approx B^{-1}n^{2/6}$.

Conjectured time $B^{3.5+o(1)}$:
Partly control f_3.

Say $n = f_5 m^5 + f_4 m^4 + \cdots + f_0$.

Choose integer $k \approx f_4/5f_5$
and integer $\ell \approx m/5f_5$.

Find all short vectors
in lattice generated by
$(m/B^3, 0, 0, 10f_5k^2 - 4f_4k + f_3)$,
$(0, m/B^4, 0, 20f_5k\ell - 4f_4\ell)$,
$(0, 0, m/B^5, 10f_5\ell^2)$,
$(0, 0, 0 \qquad\quad, m)$.

Hope for v below B^1

with $(10f_5k^2 - 4f_4k + f_3)$

$\quad + (20f_5k\ell - 4f_4\ell)v$

$\quad + (10f_5\ell^2)v^2$

below m/B^3 modulo m.

Write n in base $m + k + v\ell$.

Obtain degree-5 coefficient

on scale of $B^{-5}n^{1/6}$;

degree-4 coefficient

on scale of $B^{-4}n^{1/6}$;

degree-3 coefficient

on scale of $B^{-2}n^{1/6}$.

Hope for good degree 2.

After selecting attractive m's,
how to identify best (m, y)?

Could check smoothness of
some congruences for each m
to estimate smoothness chance.

But this is expensive:
smooth congruences are rare;
need quite a few of them
before estimate is reliable.

Want something faster,
to test more (m, y)'s.

Quickly and accurately estimate
number of small congruences
by numerically approximating
a “superelliptic integral.”

Quickly and accurately estimate
congruence smoothness chance
by approximating distribution of
a “Dirichlet series.”

So can estimate cost of
finding more smooth congruences
than exponent-vector length.
In practice: Fewer required.
Open: Estimate how many.

Given $H, m, f_5, \ldots, f_0$:
How many congruences survive initial selection of small congruences?

Consider integer pairs (i, j) with $i\mathbf{Z} + j\mathbf{Z} = \mathbf{Z}$ and $j > 0$.
How many congruences $(i - jm)(f_5 i^5 + \cdots + f_0 j^5)$ are in $[-H, H]$?

μ bound is quite crude.
Can instead enumerate j's, count i's for each j.

Faster: Numerically approximate the area of $\{(i,j) \in \mathbf{R} \times \mathbf{R} : \cdots \in [-H, H]\}$.

Number of qualifying pairs is extremely close to $(3/\pi^2)H^{2/6} \int_{-\infty}^{\infty} dx/(F(x)^2)^{1/6}$ where $F(x) = (x - m)(f_5 x^5 + \cdots + f_0)$.

Evaluate superelliptic integral by standard techniques: partition, use series expansions.

What is chance that a uniform random integer in $[1, H]$ is, e.g., 1000000-smooth?

Define S as the set of 1000000-smooth integers $n \geq 1$.

The Dirichlet series for S is $\sum [n \in S] x^{\lg n} =$

$$(1 + x^{\lg 2} + x^{2 \lg 2} + x^{3 \lg 2} + \cdots)$$
$$(1 + x^{\lg 3} + x^{2 \lg 3} + x^{3 \lg 3} + \cdots)$$
$$(1 + x^{\lg 5} + x^{2 \lg 5} + x^{3 \lg 5} + \cdots)$$
$$\cdots$$
$$(1 + x^{\lg 999983} + x^{2 \lg 999983} + \cdots).$$

Replace primes $2, 3, 5, \ldots, 999983$ with slightly larger real numbers $\overline{2} = 1.1^8$, $\overline{3} = 1.1^{12}$, $\overline{5} = 1.1^{17}$, $\ldots$, $\overline{999983} = 1.1^{145}$.

Replace each $2^a 3^b \cdots$ in S with $\overline{2}^a \overline{3}^b \cdots$, obtaining multiset $\bar{S}$.

The Dirichlet series for $\bar{S}$ is $\sum [n \in \bar{S}] x^{\lg n} =$

$$(1 + x^{\lg \overline{2}} + x^{2 \lg \overline{2}} + x^{3 \lg \overline{2}} + \cdots)$$
$$(1 + x^{\lg \overline{3}} + x^{2 \lg \overline{3}} + x^{3 \lg \overline{3}} + \cdots)$$
$$(1 + x^{\lg \overline{5}} + x^{2 \lg \overline{5}} + x^{3 \lg \overline{5}} + \cdots)$$
$$\ldots$$
$$(1 + x^{\lg \overline{999983}} + x^{2 \lg \overline{999983}} + \cdots).$$

This is simply a power series
$c_0 y^0 + c_1 y^1 + \cdots =$
$(1 + y^8 + y^{2\cdot 8} + y^{3\cdot 8} + \cdots)$
$(1 + y^{12} + y^{2\cdot 12} + y^{3\cdot 12} + \cdots)$
$(1 + y^{17} + y^{2\cdot 17} + y^{3\cdot 17} + \cdots)$
$\cdots (1 + y^{145} + y^{2\cdot 145} + \cdots)$
in the variable $y = x^{\lg 1.1}$.

Compute series mod (e.g.) y^{2910};
i.e., compute $c_0, c_1, \ldots, c_{2909}$.
$\bar{S}$ has $c_0 + \cdots + c_{2909}$ elements
$\le 1.1^{2909} < 2^{400}$, so S has
at least that many elements
$< 2^{400}$.

Can modify Dirichlet series to modify notion of smoothness.

Use $1 + x^{\lg \overline{999983}}$ instead of $(1 + x^{\lg \overline{999983}} + x^{2\lg \overline{999983}} + \cdots)$ to throw away n's having more than one factor 999983.

Multiply $c_0 y^0 + \cdots + c_{2909} y^{2909}$ by $x^{\lg \overline{1000003}} + \cdots + x^{\lg \overline{999999937}}$ to allow n's that are 1000000-smooth integers $< 2^{400}$ times one prime in $[10^6, 10^9]$.

Number-field smoothness: replace
$1 + x^{\lg p} + x^{2 \lg p} + \cdots$ with
$1 + x^{\lg N(P)} + x^{2 \lg N(P)} + \cdots$
where P is ideal, N is norm.

In all of these situations,
can compute an upper bound
on number of smooth values
to check tightness of lower bound.

If looser than desired,
move 1.1 closer to 1.
Achieve any desired accuracy.

Smoothness chance for $i - j\alpha$
in $\mathbf{Q}(\alpha)$ is, conjecturally,
very close to smoothness chance
for ideals of the same size.

Same for $(i - jm, i - j\alpha)$
in $\mathbf{Q} \times \mathbf{Q}(\alpha)$.

Integrate size distribution
of $(i - jm)(i - j\alpha)$ against
smoothness distribution of ideals.