Special number field sieve

From Wikipedia, the free encyclopedia

In number theory, a branch of mathematics, the **special number field sieve** (SNFS) is a special-purpose integer factorization algorithm. The general number field sieve (GNFS) was derived from it.

The special number field sieve is efficient for integers of the form $r^e \pm s$, where r and s are small (for instance Mersenne numbers).

Heuristically, its complexity for factoring an integer n is of the form:[1]

$$\exp\left(\left(1+o(1)\right)\left(\tfrac{32}{9}\log n\right)^{1/3}\left(\log\log n\right)^{2/3}\right)=L_n\left[1/3,(32/9)^{1/3}\right]$$

in O and L-notations.

The SNFS has been used extensively by NFSNet (a volunteer distributed computing effort), NFS@Home and others to factorise numbers of the Cunningham project; for some time the records for integer factorisation have been numbers factored by SNFS.

Contents

Overview of method [edit]

The SNFS is based on an idea similar to the much simpler rational sieve; in particular, readers may find it helpful to read about the rational sieve first, before tackling the SNFS.

The SNFS works as follows. Let n be the integer we want to factor. As in the rational sieve, the SNFS can be broken into two steps:

- First, find a large number of multiplicative relations among a *factor base* of elements of $\mathbf{Z}/n\mathbf{Z}$, such that the number of multiplicative relations is larger than the number of elements in the factor base.
- Second, multiply together subsets of these relations in such a way that all the exponents are even, resulting in congruences of the form $a^2 \equiv b^2 \pmod{n}$. These in turn immediately lead to factorizations of n: $n = \gcd(a+b,n) \times \gcd(a-b,n)$. If done right, it is almost certain that at least one such factorization will be nontrivial.

The second step is identical to the case of the rational sieve, and is a straightforward linear algebra problem. The first step, however, is done in a different, more efficient way than the rational sieve, by utilizing number fields.

Details of method [edit]

Let n be the integer we want to factor. We pick an irreducible polynomial f with integer coefficients, and an integer m such that $f(m) \equiv 0 \pmod{n}$ (we will explain how they are chosen in the next section). Let α be a root of f; we can then form the ring $\mathbf{Z}[\alpha]$. There is a unique ring homomorphism φ from $\mathbf{Z}[\alpha]$ to $\mathbf{Z}/n\mathbf{Z}$ that maps α to m. For simplicity, we'll assume that $\mathbf{Z}[\alpha]$ is a unique factorization domain; the algorithm can be modified to work when it isn't, but then there are some additional complications.

Next, we set up two parallel *factor bases*, one in $\mathbf{Z}[\alpha]$ and one in $\mathbf{Z}$. The one in $\mathbf{Z}[\alpha]$ consists of all the prime ideals in $\mathbf{Z}[\alpha]$ whose norm is bounded by a chosen value $N_{\max}$. The factor base in $\mathbf{Z}$, as in the rational sieve case, consists of all prime integers up to some other bound.

We then search for relatively prime pairs of integers (a,b) such that:

- $a+bm$ is smooth with respect to the factor base in $\mathbf{Z}$ (i.e., it is a product of elements in the factor base).

- *a+bα* is smooth with respect to the factor base in **Z**[*α*]; given how we chose the factor base, this is equivalent to the norm of *a+bα* being divisible only by primes less than $N_{\max}$.

These pairs are found through a sieving process, analogous to the Sieve of Eratosthenes; this motivates the name "Number Field Sieve".

For each such pair, we can apply the ring homomorphism φ to the factorization of *a+bα*, and we can apply the canonical ring homomorphism from **Z** to **Z**/n**Z** to the factorization of *a+bm*. Setting these equal gives a multiplicative relation among elements of a bigger factor base in **Z**/n**Z**, and if we find enough pairs we can proceed to combine the relations and factor *n*, as described above.

Choice of parameters [edit]

Not every number is an appropriate choice for the SNFS: you need to know in advance a polynomial *f* of appropriate degree (the optimal degree is conjectured to be $\left(3\frac{\log N}{\log\log N}\right)^{1/3}$, which is 4, 5, or 6 for the sizes of N currently feasible to factorise) with small coefficients, and a value *x* such that $f(x) \equiv 0 \pmod{N}$ where N is the number to factorise. There is an extra condition: *x* must satisfy $ax + b \equiv 0 \pmod{N}$ for a and b no bigger than $N^{1/d}$.

One set of numbers for which such polynomials exist are the $a^b \pm 1$ numbers from the Cunningham tables; for example, when NFSNET factored 3^479+1, they used the polynomial x^6+3 with x=3^80, since (3^80)^6+3 = 3^480+3, and $3^{480} + 3 \equiv 0 \pmod{3^{479} + 1}$.

Numbers defined by linear recurrences, such as the Fibonacci and Lucas numbers, also have SNFS polynomials, but these are a little more difficult to construct. For example, F_{709} has polynomial $n^5 + 10n^3 + 10n^2 + 10n + 3$, and the value of *x* satisfies $F_{142}x - F_{141} = 0$.[2]

If you already know some factors of a large SNFS-number, you can do the SNFS calculation modulo the remaining part; for the NFSNET example above, 3^479+1 = (4*158071*7167757*7759574882776161031) times a 197-digit composite number (the small factors were removed by ECM), and the SNFS was performed modulo the 197-digit number. The number of relations required by SNFS still depends on the size of the large number, but the individual calculations are quicker modulo the smaller number.

Limitations of algorithm [edit]

This algorithm, as mentioned above, is very efficient for numbers of the form $r^e \pm s$, for *r* and *s* relatively small. It is also efficient for any integers which can be represented as a polynomial with small coefficients. This includes integers of the more general form $a'r^e \pm b's^f$, and also for many integers whose binary representation has low Hamming weight. The reason for this is as follows: The Number Field Sieve performs sieving in two different fields. The first field is usually the rationals. The second is a higher degree field. The efficiency of the algorithm strongly depends on the norms of certain elements in these fields. When an integer can be represented as a polynomial with small coefficients, the norms that arise are much smaller than those that arise when an integer is represented by a general polynomial. The reason is that a general polynomial will have much larger coefficients, and the norms will be correspondingly larger. The algorithm attempts to factor these norms over a fixed set of prime numbers. When the norms are smaller, these numbers are more likely to factor.

See also [edit]

- General number field sieve

References [edit]

1. ^ Pomerance, Carl (December 1996), "A Tale of Two Sieves" (PDF), *Notices of the AMS* **43** (12): 1473–1485
2. ^ Franke, Jens. "Installation notes for ggnfs-lasieve4". MIT Massachusetts Institute of Technology.

Further reading [edit]

- Byrnes, Steven (May 18, 2005), "The Number Field Sieve" (PDF), *Math 129*
- Lenstra, A. K.; Lenstra, H. W., Jr.; Manasse, M. S. & Pollard, J. M. (1993), "The Factorization of the Ninth Fermat Number", *Mathematics of Computation* **61** (203): 319–349, doi:10.1090/S0025-5718-1993-1182953-4
- Lenstra, A. K.; Lenstra, H. W., Jr., eds. (1993), *The Development of the Number Field Sieve*, Lecture Notes in Mathematics **1554**, New York: Springer-Verlag, ISBN 3-540-57013-6

- Silverman, Robert D. (2007), "Optimal Parameterization of SNFS", *J. Mathematical Cryptology* (de Gruyter) **1**: 105–124, doi:10.1515/JMC.2007.007

v · t · e	Number-theoretic algorithms
Primality tests	AKS test · APR test · Baillie–PSW · ECPP test · Elliptic curve · Pocklington · Fermat · Lucas · *Lucas–Lehmer* · *Lucas–Lehmer–Riesel* · *Proth's theorem* · *Pépin's* · Quadratic Frobenius test · Solovay–Strassen · Miller–Rabin
Prime-generating	Sieve of Atkin · Sieve of Eratosthenes · Sieve of Sundaram · Wheel factorization
Integer factorization	Continued fraction (CFRAC) · Dixon's · Lenstra elliptic curve (ECM) · Euler's · Pollard's rho · $p-1$ · $p+1$ · Quadratic sieve (QS) · General number field sieve (GNFS) · ***Special number field sieve (SNFS)*** · Rational sieve · Fermat's · Shanks' square forms · Trial division · Shor's
Multiplication	Ancient Egyptian · Long · Karatsuba · Toom–Cook · Schönhage–Strassen · Fürer's
Discrete logarithm	Baby-step giant-step · Pollard rho · Pollard kangaroo · Pohlig–Hellman · Index calculus · Function field sieve
Greatest common divisor	Binary · Euclidean · Extended Euclidean · Lehmer's
Modular square root	Cipolla · Pocklington's · Tonelli–Shanks
Other algorithms	Chakravala · Cornacchia · Integer relation · Integer square root · Modular exponentiation · Schoof's
Italics indicate that algorithm is for numbers of special forms · Smallcaps indicate a deterministic algorithm	

Categories: Integer factorization algorithms