THE MULTIPLE-LATTICE NUMBER FIELD SIEVE

Daniel J. Bernstein

950518 (draft T)

Abstract. We introduce the multiple-lattice number field sieve. The formal relation between the multiple-lattice number field sieve and the number field sieve is the same as the formal relation between the multiple-polynomial quadratic sieve and the quadratic sieve.

1. Introduction

The object of this paper is to show that the *multiple-polynomial variation* [9] of the *quadratic sieve* [8] has an analogue for the *number field sieve* [5].

In section 2 we review the now-standard sieving technique introduced in [8], with emphasis on concepts shared by the quadratic sieve and the number field sieve. In section 3 we explain how the general *multiple-lattice* idea applies to the quadratic sieve and the number field sieve.

2. Sieving

An integer is **smooth** if all its prime divisors are small. In both the quadratic sieve and the number field sieve we search for smooth values of a polynomial on an integer lattice. We choose the polynomial in view of an integer, n, and we hope that by combining enough smooth values we can construct a factor of n.

QS. The quadratic sieve polynomial is $g(x) = (x+r)^2 - n$. Here r is an integer close to $\sqrt{n}$; say $r = \lfloor \sqrt{n} \rfloor$. We want to find smooth values of g on the one-dimensional lattice of integers x.

We pause to establish some terminology. Let p be a positive integer. Then p divides $g(x)$ if and only if p divides $g(x+p)$. Hence the set of x such that p divides $g(x)$ is a finite union of arithmetic progressions. We refer to these progressions as **p-lattices**; each progression is a **sublattice** of our lattice of integers x.

Here is what it means to **sieve** for smooth values of g. Consider a row of boxes labelled by the values of x in some range. Figure out all the p-lattices for various prime powers p. Given a p-lattice L, hop through the boxes for $x \in L$, and record p in each box.

After sieving we zoom through all the boxes. In the box for x we have recorded some prime powers p dividing $g(x)$. If it seems likely, on the basis of a quick

1991 *Mathematics Subject Classification.* Primary 11Y05.

This paper was included in the author's thesis at the University of California at Berkeley. The author was supported in part by a National Science Foundation Graduate Fellowship.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

estimate for $g(x)$, that there are enough p's for $g(x)$ to be smooth, then we spend a few moments dividing $g(x)$ by its known factors and checking whether the quotient z is smooth.

Note that to find z we do not have to calculate $g(x)$ in high precision. When there are many p's it is easier to compute z modulo some integer t coprime to each p. Given bounds on $g(x)$ we may find upper and lower bounds for z, and thus reconstruct z from $z \bmod t$ if t is large enough. We could even compute z modulo several small t's, and reconstruct z with the Chinese Remainder Theorem.

NFS. In the number field sieve we choose an integer m and an integer polynomial f such that $n = f(m)$. Let d be the degree of f; then $N(x,y) = f(x/y)y^d$ is a homogeneous polynomial of degree d in x and y. Now the number field sieve polynomial is $h(x,y) = (x - my)N(x,y)$. We want to find smooth values of h on the two-dimensional lattice of integers x and y. (This is not the most general form of the number field sieve; see, for example, [2, section 12].)

Fix a positive integer p. Then the set of x such that p divides $h(x,y)$ is a finite union of sublattices, which we call **p-lattices**, of our two-dimensional lattice. We use the p-lattices for various prime powers p to sieve over a two-dimensional array of boxes. In practice one can speed this up a bit by paying attention to the structure of the number field sieve polynomial: $h(x,y)$ is smooth if and only if $x - my$ and $N(x,y)$ are both smooth. We may sieve $x - my$ and $N(x,y)$ separately.

3. Multiple lattices

In section 2 we explained how to look for smooth values of a polynomial on an integer lattice. We may use the same procedure to look for smooth values on a *sublattice* of the original lattice. We propose the term **multiple-lattice variation** for the general practice of sieving over many sublattices. In this section we consider the impact of the multiple-lattice variation upon the effectiveness of sieving.

History. In the beginning Carl Pomerance discovered the quadratic sieve [8]. The multiple-lattice variation was first discovered by James A. Davis, then applied by Davis together with Diane B. Holdridge [4], and independently discovered and applied by Peter L. Montgomery [9]. Later John M. Pollard discovered the number field sieve [5] and applied the Davis/Holdridge variation to it [7].

Several versions of the multiple-lattice quadratic sieve and the multiple-lattice number field sieve have their own names. The Davis/Holdridge version is the *special-q variation*; the Montgomery version is the *multiple-polynomial variation*; and the Pollard version is the *lattice sieve*.

Note that [3] is an unrelated method, which involves multiple polynomials in a much more fundamental way than the multiple-polynomial variation. To prevent confusion one could refer to [3] as the **multiple number field sieve**.

Overhead. Sieving time per box is approximately constant. But this approximation breaks down if we search through too few boxes, because we always spend some extra time figuring out p-lattices for each p. To keep overhead down we must sieve through at least, say, B boxes at a time: an interval of length B for the quadratic sieve, or some sort of rectangular blob of area B for the number field sieve. In the following discussion we will treat B as a constant.

Expansion. Our goal is to find as many smooth polynomial values as possible. Since small values are more likely to be smooth than large values, we should sieve over regions where our polynomial is as small as possible.

Both the quadratic sieve polynomial g and the number field sieve polynomial h are designed to be reasonably small near the origin. Say we sieve over B boxes centered around the origin. For the quadratic sieve, $g(x)$ is typically as large as $B\sqrt{n}$. For the number field sieve, $x - my$ is typically as large as $(m/2)\sqrt{B}$, and $N(x,y)$ is typically as large as $cB^{d/2}$ where c reflects the coefficients of N, so $h(x,y)$ is typically proportional to $B^{(d+1)/2}$.

The primary disadvantage of sieving over a sublattice is that we move rapidly away from the origin. B boxes within a sublattice of determinant q cover the same range as Bq boxes within the original lattice. Hence, for the quadratic sieve, $g(x)$ expands by a factor of q. For the number field sieve, $x - my$ expands by a factor of $\sqrt{q}$ and $N(x,y)$ expands by a factor of $q^{d/2}$.

Note that one can reduce the expansion slightly by choosing sublattices that interact well with the coefficients of the polynomial. For example, as pointed out in [9], polynomial values in the *multiple-polynomial variation* are typically a factor of $\sqrt{8}$ smaller than polynomial values in the *special-q variation*.

Note, furthermore, that the same disadvantage occurs if we sieve over more than B boxes in the original lattice. If we sieve over kB boxes, we see a k expansion for the quadratic sieve, and an overall $k^{(d+1)/2}$ expansion for the number field sieve.

Squeezing. The primary advantage of the multiple-lattice variation is that for our sublattice we may select a determinant-q q-lattice. Thus we force every value of the polynomial to be divisible by q.

For the quadratic sieve we squeeze a factor of q out of $g(x)$, exactly balancing the expansion by q mentioned above. We thus have a nearly infinite supply of equally useful sublattices. It is much better to systematically use these sublattices than to sieve farther away from the origin in the original lattice.

We briefly survey examples of the multiple-lattice quadratic sieve. The *special-q variation* uses small primes q, in the same range as allowable factors of smooth numbers. The *multiple-polynomial variation* uses larger q's—preferably squares, which are useful for factoring even though they are not smooth. (We can use any q at the expense of a single smooth value on the q-lattice. This is worthwhile if we find several smooth values.) The *self-initialization procedure* reduces overhead by simultaneously considering several q's built up from subsets of a single small set of primes.

Next we consider the multiple-lattice number field sieve. After squeezing a q factor out of $h(x,y)$, we are left with an overall expansion of $q^{(d-1)/2}$. Since $q^{(d-1)/2}$ grows as q grows, it may be better to sieve more than B boxes for a single q than to sieve B boxes for several q's.

We may quantify this effect as follows. If we sieve B boxes for q_0, we suffer an overall expansion of $q_0^{(d-1)/2}$. If we sieve kB boxes for $q < q_0$, we suffer an overall expansion of $k^{(d+1)/2}q^{(d-1)/2}$. To make these expansions match we should take $k \approx (q_0/q)^{(d-1)/(d+1)}$. For example, if $d = 5$, the number of boxes we sieve for q should be proportional to $q^{-2/3}$. Sieving $10B$ boxes for a single $q \approx 10^7$ is about as productive as sieving B boxes for ten values of $q \approx 3 \cdot 10^8$.

To analyze the multiple-lattice number field sieve more accurately we would have to take into account the structure of $h(x,y)$. We have a choice: we may squeeze our

q factor out of either $x - my$ or $N(x, y)$. Probably squeezing q out of $x - my$, as in the *lattice sieve*, is best for special numbers, while squeezing q out of $N(x, y)$ is best for general numbers. In the first case the total effect of squeezing and expansion is to gain $\sqrt{q}$ for $x - my$ and lose $q^{d/2}$ for $N(x, y)$. In the second case we lose $\sqrt{q}$ for $x - my$ and lose $q^{d/2-1}$ for $N(x, y)$. The resulting smoothness probabilities will depend on the relative sizes of q, $x - my$, $N(x, y)$, and the bound for smooth primes.

Collisions. Multiple lattices generally intersect. One difficulty of sieving over several lattices is that any smooth value at the intersection of two lattices—i.e., any smooth value divisible by two of our q's—will be found twice. Davis and Holdridge call this event a **collision**. It is not difficult to weed out duplicates (or triplicates!), but any collision indicates that we have wasted some sieving time.

When every q has a large prime factor, there is no problem: we insist that values be smooth (apart from q), and hence not divisible by any other q.

In the *special-q method* it is traditional to ignore p-sublattices of q-lattices for $p > q$, on the theory that polynomial values divisible by p will be found when $q = p$. This conveniently papers over the duplication problem without solving it. For the number field sieve, that theory is not even close to correct, and we suggest breaking with tradition.

Any precise analysis of the multiple-lattice variation will have to take collisions into account.

Avoiding redundancy. Here is a different use of the multiple-lattice number field sieve.

The quadratic sieve polynomial $q(x) = (x + r)^2 - n$ takes the same value for x and for $-x - 2r$. If one is smooth then the other is too. But we cannot double our money this way, since these two smooth values together will eventually produce a trivial factorization of n.

In the number field sieve, (x, y) and $(-x, -y)$ carry the same information. So we insist that y be positive. Furthermore, given (x, y) it is redundant to consider (dx, dy) for $d > 1$. So we insist that x and y be coprime.

Now the number of useful inputs with a given y, and hence the number of smooth values, is correlated with $\phi(y)/y$, the chance that x is coprime to y. So to save time we can simply throw away all y divisible by 6. Arjen K. Lenstra independently proposed considering the three coprime possibilities for $(x \bmod 2, y \bmod 2)$. In general we may consider the coprime possibilities for $(x \bmod s, y \bmod s)$. Each possibility defines a sublattice, and we sieve separately over each sublattice.

References

1. Thomas Beth et al. (editors), *Advances in Cryptology—Proceedings of EUROCRYPT 84*, Lecture Notes in Computer Science 209, Springer-Verlag, Berlin, 1985.
2. Joseph P. Buhler, Hendrik W. Lenstra, Jr., Carl Pomerance, *Factoring integers with the number field sieve*, in [5], 50–94.
3. Don Coppersmith, *Modifications to the number field sieve*, to appear, Journal of Cryptology.
4. James A. Davis, Diane B. Holdridge, *Factorization using the quadratic sieve algorithm*, Sandia Report SAND 83–1346, Sandia National Laboratories, Albuquerque, 1983.
5. Arjen K. Lenstra, Hendrik W. Lenstra, Jr. (editors), *The development of the number field sieve*, Lecture Notes in Mathematics 1554, Springer-Verlag, Berlin, 1993.
6. Hendrik W. Lenstra, Jr., R. Tijdeman (editors), *Computational methods in number theory*, Mathematical Centre Tracts 154/155, Mathematisch Centrum, Amsterdam, 1982.

7. John M. Pollard, *The lattice sieve*, in [5], 43–49.
8. Carl Pomerance, *Analysis and comparison of some integer factoring algorithms*, in [6], 89–139.
9. Carl Pomerance, *The quadratic sieve factoring algorithm*, in [1], 169–182.

5 Brewster Lane, Bellport, NY 11713