Proceedings of Symposia in Applied Mathematics
Volume 48, 1994

THE NUMBER FIELD SIEVE

CARL POMERANCE*

Dedicated to the memory of D. H. Lehmer

ABSTRACT. The most exciting recent development in the integer factorization problem is the number field sieve. It has had some spectacular successes with integers in certain special forms, most notably the factorization in 1990 of the 155 decimal digit number $2^{512}+1$. For arbitrary hard integers, it now appears to threaten the quadratic sieve as the algorithm of choice. In this paper the number field sieve, and the ideas behind it, are described.

1. Introduction

The problem of factoring integers is a good one to test our mettle as mathematicians. First, it is as fundamental as a problem can be. Second, while having the patina of centuries of history, the problem has taken on a new urgency for its connection with public-key cryptography. Third, it is a very *hard* problem, but not so hard that we do not occasionally gain an insight and make an advance. In this paper I would like to describe the background and main ideas of one of these advances, the number field sieve of John Pollard.

Twenty years ago, at the dawn of the continued fraction factoring algorithm of Brillhart and Morrison, factoring hard 50-digit numbers seemed barely possible, while 100-digit numbers could not even be dreamed about. Ten years ago, when my quadratic sieve factoring algorithm first began to enjoy some success, we indeed did dream of 100-digit numbers, and within a few more years, they were falling regularly. Today, with the number field sieve, our dreams have moved on to the hard 150-digit numbers.

Other advances have enabled us to factor broad classes of formerly unattackable numbers, and in the process have changed our thinking on which integers

1991 *Mathematics Subject Classification.* Primary 11Y05.

Key words and phrases. Number field sieve, quadratic sieve, smooth numbers, exponent vectors.

*Supported in part by an NSF grant.

This paper is in final form and no version of it will be submitted for publication elsewhere.

should be considered as hard to factor. Among these developments are the elliptic curve method of Lenstra and certain optimal cases of the number field sieve itself. One of these optimal cases is the Fermat number $F_9 = 2^{512} + 1$. Despite the fact that it has 155 decimal digits, it is particularly well suited to the number field sieve. It was factored in 1990 into three prime factors – see [15].

To be sure, some part of our success in factoring has little to do with new algorithms and theory, but rather with dramatic improvements in computers and their availability. On the other hand, these advances in computing resources helped to influence and shape the kinds of algorithms that were being developed. For example, almost all of the newer methods are easily distributable to scores or more of unextraordinary computers and most of our greatest successes were accomplished by doing just this.

Several of the key papers on the development of the number field sieve are collected together in the book [13]. Included there is Pollard's original note whose informal distribution started the whole ball rolling. The number field sieve has also been suggested for use in computing discrete logarithms in the multiplicative group of a prime finite field (and perhaps more generally). See [11, 25] for more on this.

Thanks are due to Joe Buhler, Arjen Lenstra, Hendrik Lenstra and Walter Gautschi for their critical comments on an earlier version of this paper.

2. Congruent squares

For which odd numbers $n > 1$ does the congruence $x^2 \equiv y^2 \bmod n$ for integers x, y force us to conclude that we must have $x \equiv \pm y \bmod n$? Since $x^2 \equiv y^2 \bmod n$ is equivalent to n dividing $(x-y)(x+y)$, clearly it is just odd primes n with this property. Perhaps not so interesting as a primality test, this observation can be quite useful in factoring n. For if $x^2 \equiv y^2 \bmod n$ and $x \not\equiv \pm y \bmod n$, then the greatest common divisor of $x - y$ and n is a nontrivial factor of n.

It is a simple matter to compute the greatest common divisor of two given integers by means of Euclid's algorithm. This has us replace the larger integer of the pair with its remainder upon division by the smaller integer. If this remainder is 0, then the smaller integer is the greatest common divisor, while if it is not 0, we may repeat the process.

A wide class of factoring algorithms thus set out to find two squares x^2, y^2 that are congruent mod n. It seems difficult to force the condition $x \not\equiv \pm y$ mod n, which is necessary for splitting n. Thus, the factoring algorithms in this class either use randomness or "pseudorandomness" to count on at least a fair proportion of the congruent pairs of squares produced to lead to a factorization.

Thus, perhaps we should revise the question above and ask for which odd numbers $n > 1$ do we have at least as many pairs of integers x, y with $x^2 \equiv y^2 \bmod n$ and $x \not\equiv \pm y \bmod n$ as we have pairs of integers x, y with $x \equiv \pm y$ mod n. Let us restrict our attention to pairs of integers x, y with $(xy, n) = 1$. The problem is thus reduced to counting solutions to $x^2 \equiv 1 \bmod n$. This congruence has 2^l solutions if n has l distinct prime factors, so that the answer to our revised question is: for odd numbers n divisible by at least two distinct primes.

Hence, finding pairs of congruent squares may not be a good way to factor n if n is a power of a prime. However, perfect powers are easy to factor. For example, to factor a square, we can first find a real approximation to the square root, round to nearby integers and see if one of them squared is our number. The same idea works for higher powers, and for a given number n, we only have to check exponents up to $\log_2 n$, since if $n = m^k$ where $n > m > 1$ are integers, then $k = \log_m n \le \log_2 n$.

In summary, the problem of factoring n for n odd and composite can be reduced to checking if n is a nontrivial power and, if not, finding solutions to $x^2 \equiv y^2 \bmod n$ in a random or pseudorandom fashion.

3. Multiplying congruences

How then are we to find solution pairs x, y to $x^2 \equiv y^2 \bmod n$? We exploit the ideas that being a square is a multiplicative condition and that congruences mod n may be multiplied. Thus, if we have integers s_i, t_i for $i = 1, \dots, k$ where $\prod s_i = x^2$, $\prod t_i = y^2$ and each $s_i \equiv t_i \bmod n$, then $x^2 \equiv y^2 \bmod n$.

Suppose we have a very large set of pairs s_i, t_i where each pair has $s_i \equiv t_i \bmod n$. Is there a nonempty subset $\mathcal{I}$ of subscripts i such that $\prod_{i \in \mathcal{I}} s_i$ and $\prod_{i \in \mathcal{I}} t_i$ are both squares? And if there is such a set $\mathcal{I}$, is there an efficient way to find it?

To make matters a little simpler, let us ask these questions just for one side of the congruence. Given integers $s_1, \dots, s_k$, is there a nonempty set $\mathcal{I} \subset \{1, \dots, k\}$ such that $\prod_{i \in \mathcal{I}} s_i$ is a square? And can we find such a set $\mathcal{I}$?

For any integer $s \neq 0$, consider the "exponent vector" for s. This is the vector $v(s) = (v_{-1}(s), v_2(s), v_3(s), \dots)$ where the subscripts are -1 and the primes, and where

$$s = (-1)^{v_{-1}(s)} \prod_p p^{v_p(s)}.$$

The exponent $v_{-1}(s)$ is taken to be in $\{0, 1\}$. For a given integer s, all but finitely many of the exponents $v_p(s)$ are 0. Note that $\prod_{i \in \mathcal{I}} s_i$ is a square if and only if

$$\sum_{i \in \mathcal{I}} v_{-1}(s_i), \quad \sum_{i \in \mathcal{I}} v_2(s_i), \quad \sum_{i \in \mathcal{I}} v_3(s_i), \dots$$

are all even numbers.

With this observation, we can reduce our question about squares to one on linear algebra over the field $\mathbb{F}_2$ of two numbers. Thus, given $s_1, \dots, s_k$, we first find the exponent vectors $v(s_1), \dots, v(s_k)$, next we reduce them mod 2 so that we have a sequence of vectors $v(s_1) \bmod 2, \dots, v(s_k) \bmod 2$ over $\mathbb{F}_2$. Our problem is to determine if these vectors are linearly dependent and, if so, to find a nontrivial relation. Since there is just one nonzero scalar, a linear dependency takes the form $\sum_{i \in \mathcal{I}} v(s_i) \bmod 2 = 0$. This corresponds exactly to $\prod_{i \in \mathcal{I}} s_i$ being a square.

4. Smooth numbers

We have been a little vague about the *dimension* of the vector space in the preceding discussion. Clearly, the exponent vectors $v(s)$ mod 2, as s runs over the positive integers, span an infinite-dimensional vector space. Why then should we expect $v(s_1) \bmod 2, \dots, v(s_k) \bmod 2$ to be linearly dependent?

In general, we should not expect this to occur. However, if the integers $s_1, \dots, s_k$ are chosen randomly from among the integers of a certain size, and if k is sufficiently large, then it will be highly likely that the vectors $v(s_i)$ mod 2 will be linearly dependent over $\mathbb{F}_2$; that is, that some nonempty set $\mathcal{I}$ exists with $\prod_{i\in\mathcal{I}} s_i$ being a square.

Let us formulate a precise problem. For x, k positive real numbers, say we choose random integers $s_1, s_2, \dots, s_{[k]}$ independently and with the uniform distribution from the interval $[1, x]$. Let $\Pr(x, k)$ denote the probability that some nonempty set $\mathcal{I} \subset \{1, \dots, [k]\}$ exists with $\prod_{i\in\mathcal{I}} s_i$ a square. How large should k be so that $\Pr(x, k)$ is greater than 1/2?

Let us prove the following result.

Proposition 4.1. *For each $\varepsilon > 0$ we have*

$$\lim_{x\to\infty} \Pr(x, \exp((1+\varepsilon)\sqrt{2\log x \log\log x})) = 1.$$

Proof. We shall say a positive integer n is z-smooth if no prime factor of n exceeds z. Let $\psi(x, z)$ denote the number of z-smooth integers n in $[1, x]$. Let

$$z = \exp(\sqrt{(1/2)\log x \log\log x}), \quad k = \exp((1+\varepsilon)\sqrt{2\log x \log\log x}). \tag{4.2}$$

Say we choose $s_1, s_2, \dots, s_{[k]}$ from among the integers in $[1, x]$ independently and with the uniform distribution. Then the expected number of $i \le k$ with s_i being z-smooth is

$$\frac{[k]}{[x]}\psi(x, z). \tag{4.3}$$

From [5], we have that

$$\psi(x, z) = x/u^{(1+o(1))u} \quad \text{as} \quad x \to \infty,$$

where $u = \log x/\log z = \sqrt{2\log x/\log\log x}$. Thus,

$$\psi(x, z) = x \cdot \exp(-(1+o(1))\sqrt{(1/2)\log x \log\log x}) = x/z^{1+o(1)}$$

as $x \to \infty$. (That is, the particular choice of z above satisfies $\psi(x, z) \approx x/z$.)

Thus, the expression in (4.3) exceeds

$$\frac{k}{x} \cdot \frac{x}{z^{1+\varepsilon}} = z^{1+\varepsilon}$$

for all large x. In particular, from the binomial distribution, the probability that there are at least z values of $i \leq k$ with s_i being z-smooth tends to 1 as $x \to \infty$.

So let us assume that there are at least z values of $i \leq k$ with s_i being z-smooth. Clearly, for a z-smooth number s, the coordinates of the exponent vector for s are all 0 for primes larger than z. By truncating these 0's, we have the exponent vectors lying in a $(\pi(z)+1)$-dimensional space, where $\pi(z)$ denotes the number of primes up to z. But we have at least z such vectors and $z > \pi(z)+1$ for all $z > 3$. Thus, the exponent vectors (reduced mod 2) are linearly dependent over $\mathbb{F}_2$ and, as we saw in the preceding section, this leads us to a nonempty set of indices i where the product of the s_i's is a square.

This simple proposition (or rather, its proof) is crucial for understanding the role of smooth numbers in combination of congruences factorization algorithms. We are presented in some manner with random or pseudorandom integers s_i in some range, and we wish to find a nonempty subsequence whose product is a square. Proposition 4.1 not only tells us we are likely to be successful by a certain point, it suggests an efficient way to *find* such a subsequence. First, if the numbers s_i all lie below x, we take z by (4.2) and *discard* any s_i which is not z-smooth. When we have found sufficiently many values of i with s_i being z-smooth, a linear algebra step over $\mathbb{F}_2$ on a system of size about z allows us to find the sought-after subsequence.

I conjecture that Proposition 4.1 is best possible in that if "$1+\varepsilon$" is replaced with "$1-\varepsilon$", the limit in the Proposition is 0.

5. How to recognize smooth numbers

If we are presented with many randomly chosen integers $s \leq x$, and if we choose z by (4.2), then only a few such numbers s will be z-smooth. In fact, as we saw in the last section, the probability that s is z-smooth is about $1/z$. If we are to use the above ideas in an efficient factorization algorithm, we shall need to have an efficient "smoothness test," that is, an algorithm that can quickly recognize the z-smooth numbers among many random (or pseudorandom) integers.

The first method that comes to mind is trial division, taking about z steps to recognize whether s is z-smooth. This is the smoothness test used in the continued fraction factoring algorithm. The "early abort strategy" (see [21]) does not complete the trial division on a particular number s if at increasing points a factored portion of s does not exceed an increasing threshold. On average, this method takes about $z^{1/2}$ steps as a smoothness test, though some smooth numbers may be lost. If trial division in the early abort strategy is replaced with the "fast factorials" method of Pollard and Strassen (see [21]), the combined method takes about $z^{1/4}$ steps on average.

The elliptic curve method factors smooth numbers much sooner than numbers with several large prime factors. Used as a smoothness test, it takes z^{ε} steps to recognize a z-smooth number. More precisely, the elliptic curve method is expected to take $\exp((1+o(1))\sqrt{2\log z \log\log z})$ steps as $z \to \infty$. This is what we conjecture. What can be proved is that it recognizes most z-smooth numbers in $\exp((\log z)^{5/6+\varepsilon})$ steps (see [18, 23]). Recently, Lenstra, Pila and I showed that a certain *hyperelliptic* curve method is expected to factor any z-smooth number in $\exp((\log z)^{2/3+\varepsilon})$ steps (see [17] for the beginning of the proof).

The elliptic and hyperelliptic curve methods are not practical subroutines in current combination of congruences algorithms, though they are of interest in a rigorous treatment of factoring. It is not inconceivable, though, that the elliptic curve method, perhaps augmented with an early abort strategy, could be of practical use in a combination of congruences algorithm. And perhaps the algorithm of [6] is a candidate.

If the numbers s are truly random, then we know of no better smoothness tests. This is true even for certain pseudorandom sequences, such as in the continued fraction factoring algorithm. However, there are other pseudorandom sequences which quite nicely lend themselves to a very simple smoothness test.

The sieve of Eratosthenes is well understood as an efficient method of finding all the primes to some point. In this sieve composite numbers are crossed off until only primes remain. Let us look at these numbers that are crossed off. They are marked as many times as they have prime factors used in the sieve. Thus, smooth numbers will be crossed off many times over, and we might use this phenomenon as a way of recognizing them.

To make this idea a little more precise, one should not count the number of times a number is crossed off, which treats each prime factor of the number with equal weight, but rather higher primes (below z) should carry more weight since they make up a larger portion of the numbers that they divide. We weight each prime in the sieve (the primes up to z) with its logarithm (say to base 2 and rounded to the nearest integer). Marking a location with a prime amounts to adding the approximate logarithm of the prime to a counter at the location. When the count at a particular location exceeds some threshold, the number corresponding to this location is reported as a candidate z-smooth number. Since reports are few, we can afford to use a slow method, such as trial division, to check the candidates.

This is all well and good if the numbers s are consecutive integers up to some point. But what about more general sequences? A moment's reflection reveals that we can use this sieve idea whenever the numbers s are consecutive values of a polynomial with integer coefficients. If we consider at least z consecutive values of this polynomial, then the time spent sieving is only about $O(\log\log z)$ steps per value, the O-constant depending on the choice of polynomial. (For z tending to infinity, this O-constant tends to the number of distinct irreducible factors of the polynomial.) Moreover, as we have seen, these steps are particularly easy computer instructions such as adding low precision integers.

The comparison between z steps per candidate with trial division and $\log\log z$ steps with sieving is striking. Our problem has us searching for z-smooth needles in a huge haystack. Trial division would have us examine every straw by hand to see if it is a needle. In contrast, when sieving is appropriate, it is as if we could pass a strong magnet over the haystack fed over a rapidly moving conveyor belt, with the needles jumping to the magnet as they pass by.

6. The quadratic sieve

In §3 we described a general factoring scheme where one has at hand many congruences $s_i \equiv t_i \bmod n$ and somehow finds a nonempty set $\mathcal{I}$ of indices i where $\prod_{i\in\mathcal{I}} s_i$ and $\prod_{i\in\mathcal{I}} t_i$ are both squares, say x^2 and y^2. Then $x^2 \equiv y^2 \bmod n$ and, as we saw in §2, this may lead to a nontrivial factorization of n via the greatest common divisor of $x - y$ and n.

Consider the congruences $t^2 \equiv t^2 - n \bmod n$, where t runs over consecutive integers. Since one side of these congruences is already a square, the problem is reduced to finding a set $\mathcal{T}$ of values of t where $\prod_{t\in\mathcal{T}}(t^2 - n)$ is a square. The search for $\mathcal{T}$ is reduced, via the ideas of §4, to a search for values of t with $t^2 - n$ smooth. Since this function of t is a polynomial with integer coefficients, we can use a sieve, as described in §5, to locate rapidly those numbers t with $t^2 - n$ smooth.

This simple algorithm is the quadratic sieve factorization algorithm. As we saw in §4, the number k of values of t that must be examined depends on x, a bound for the size of the numbers $t^2 - n$ that we hope are smooth. By taking t in an interval near $\sqrt{n}$, the numbers $t^2 - n$ are $O(n^{1/2+\epsilon})$, so that the number k in §4 is about $\exp(\sqrt{\log n \log\log n})$.

This expression is in fact the approximate heuristic complexity of the quadratic sieve algorithm. To be sure, one must also account for the complexity of finding a linear dependency among the exponent vectors mod 2. By using the sparse matrix methods in [7, 8, 12, 20, 24, 26], this complexity is the same as the sieving time; that is, about $\exp(\sqrt{\log n \log\log n})$.

A problem with the quadratic sieve is that the numbers $t^2 - n$ grow as t moves away from $\sqrt{n}$. In particular, if T values of t are considered, all we can say about the size of the numbers $t^2 - n$ is that they are $O(T\sqrt{n})$. The penalty for large values of $t^2 - n$ is that they are less likely to be smooth.

This problem is neatly mitigated by an idea of Peter Montgomery (see [22]). He has us replace t above with $at + b$, where a, b are integers satisfying $b^2 \equiv n \bmod a$, $|b| \le a/2$. Then the polynomial $(at + b)^2 - n$ has all of its values divisible by a. If also a is a square, then we are only concerned with whether

$$f_{a,b}(t) = \frac{1}{a}((at + b)^2 - n)$$

has smooth values. The advantage is that there are many choices for a, b, and thus many polynomials that one may use. If one wishes to sieve $2M$ values per polynomial, one chooses a near $\sqrt{2n}/M$ and sieves over an interval centered at 0. A simple computation shows that $f_{a,b}(t) = O(M\sqrt{n})$, so that the numbers we hope are smooth do not grow as we sieve over more and more numbers. We just take more and more polynomials $f_{a,b}(t)$. This is the multiple polynomial variation of the quadratic sieve. Another version of this idea was proposed earlier by Davis and Holdridge [10].

The size of the numbers we wish to find smooth in the basic quadratic sieve is about $\sqrt{n}\exp(\sqrt{\log n \log\log n})$. This drops to about $\sqrt{n}\exp\left(\frac{1}{2}\sqrt{\log n \log\log n}\right)$ in the multiple polynomial version. Both expressions are of the form $n^{1/2+\varepsilon}$, so both versions of the quadratic sieve have about the same heuristic complexity. However, the drop to a smaller "ε" makes a big difference in practice.

This thought underscores an important principle in combination of congruences factorization algorithms: The smaller are the auxiliary numbers that you hope to find smooth, the faster the algorithm will be. As we will see, in the number field sieve, the auxiliary numbers are about $\exp((\log n)^{2/3})$; that is, they are much smaller than $\sqrt{n}$. This is why the method is so exciting and has so much potential.

7. The number field sieve

Suppose $f(t)$ is a monic polynomial irreducible over $\mathbb{Z}$, suppose m is an integer with $f(m) \equiv 0 \bmod n$ and $\alpha \in \mathbb{C}$ is a root of f. There is thus a natural homomorphism φ from the ring of algebraic integers $\mathbb{Z}[\alpha]$ to $\mathbb{Z}/n\mathbb{Z}$, where $\varphi(\alpha) = m \bmod n$. That is, if $g(t)$ is any polynomial over $\mathbb{Z}$, then $\varphi(g(\alpha)) = g(m) \bmod n$.

Of what interest to us are f, α, m and φ? Well, suppose we could find a set $\mathcal{S}$ of polynomials g over $\mathbb{Z}$ such that $\prod_{g\in\mathcal{S}} g(\alpha)$ is a square, say β^2, in $\mathbb{Z}[\alpha]$ and $\prod_{g\in\mathcal{S}} g(m)$ is a square, say y^2, in $\mathbb{Z}$. Let x be an integer such that $\varphi(\beta) = x \bmod n$. Then

$$x^2 \equiv \varphi(\beta)^2 \equiv \varphi(\beta^2) \equiv \varphi\left(\prod_{g\in\mathcal{S}} g(\alpha)\right) \equiv \prod_{g\in\mathcal{S}} g(m) \equiv y^2 \bmod n.$$

That is, we have found a pair of squares that are congruent mod n and we may attempt to factor n by computing $(x-y, n)$.

The reader is an expert now and may be a bit suspicious. This scenario will be promising only if we can find choices for $g \in \mathbb{Z}[t]$ such that $g(m)$ is small and so likely to be smooth. And how in the world are we to combine various numbers $g(\alpha)$ to get a square in $\mathbb{Z}[\alpha]$? Do we have a notion of smoothness in $\mathbb{Z}[\alpha]$, and can we factor $g(\alpha)$ into primes to get an exponent vector?

Let us first attack the question on $g(m)$, for it is much easier. Suppose we wish our polynomial $f(t)$ to have degree d. The size of d as a function of n will be discussed momentarily, but for now we assume that $d > 1$ and $n > 2^{d^2}$. Before the polynomial f is chosen, we do the somewhat surprising thing of first choosing the integer m. We let $m = \lfloor n^{1/d} \rfloor$. We write n in the base m, so we find integers $c_0, c_1, \ldots, c_d$ in $\{0, 1, \ldots, m-1\}$ with

$$n = c_d m^d + c_{d-1} m^{d-1} + \cdots + c_0.$$

It is easy to show we must have $c_d = 1$. Let

$$f(t) = t^d + c_{d-1}t^{d-1} + \cdots + c_0.$$

Then f is a monic polynomial in $\mathbb{Z}[t]$ and $f(m) = n$. Is f irreducible? It probably is since most primitive polynomials over $\mathbb{Z}$ are. In fact, though, if f is not irreducible then we are in luck. Using the algorithm of [14], we can factor f into irreducibles in time polynomial in $\log n$, and if the factorization is nontrivial, by substituting m for t, we get a nontrivial factorization of n (see [3]). In this event we cancel plans to spend a lot of effort splitting n, for we have just succeeded!

So let us assume that f is irreducible over $\mathbb{Z}$. Suppose we take for our polynomials $g(t)$ the linear polynomials $a - bt$, where a, b run over small coprime numbers with $b > 0$, say $0 < b \le B$, $0 \le |a| \le B$. Since m is about $n^{1/d}$, we see that the integers $a - bm$ can be rather small compared with n, especially if we do not have to take B large and we can choose d with some size. Further, it is easy to see how a sieve might be used to pick out pairs a, b with $a - bm$ being smooth: we fix b and sieve over a, then choose the next b and sieve again, continuing until we exhaust the choices for b.

Well, I have put off the questions on how to produce squares in $\mathbb{Z}[\alpha]$ long enough. If $\mathbb{Z}[\alpha]$ happens to be a unique factorization domain, we can get a glimmer of what it means for $a - b\alpha$ to be a smooth element, namely it factors into "small" primes. But we cannot count on any special properties of $\mathbb{Z}[\alpha]$.

Consider the norm map N from the field $\mathbb{Q}(\alpha)$ to $\mathbb{Q}$. If $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_d$ are the conjugates in $\mathbb{C}$ of α, and $g \in \mathbb{Q}[t]$, then $N(g(\alpha)) = \prod_{i=1}^{d} g(\alpha_i)$. The norm map is multiplicative and it sends algebraic integers to $\mathbb{Z}$. In particular, $N(\mathbb{Z}[\alpha]) \subset \mathbb{Z}$. We thus have the following simple result: If $\mathcal{S}$ is a set of coprime pairs (a, b) of integers and $\prod_{(a,b)\in\mathcal{S}}(a - b\alpha)$ is a square in $\mathbb{Z}[\alpha]$, then $\prod_{(a,b)\in\mathcal{S}} N(a - b\alpha)$ is a square in $\mathbb{Z}$.

Can we at least arrange for this necessary condition to hold? Note that for any numbers $a, b \in \mathbb{Q}$ with $b \ne 0$, we have

$$\begin{aligned} N(a - b\alpha) &= \prod_{i=1}^{d}(a - b\alpha_i) = b^d \prod_{i=1}^{d}\left(\frac{a}{b} - \alpha_i\right) = b^d f\left(\frac{a}{b}\right) \\ &= a^d + c_{d-1}a^{d-1}b + \cdots + c_1ab^{d-1} + c_0b^d. \end{aligned}$$

Thus, the norm of $a - b\alpha$ is a polynomial with integral coefficients in the two variables a, b.

For integers a, b, let us say that $a - b\alpha$ is z-smooth if $N(a - b\alpha)$ is z-smooth. So now it should be clear how we may find a set $\mathcal{S}$ of coprime integer pairs (a, b) such that $\prod_{(a,b)\in\mathcal{S}} N(a - b\alpha)$ is a square in $\mathbb{Z}$. Namely, we use a sieve to detect pairs a, b where $a - b\alpha$ is z-smooth, create exponent vectors from the prime factorizations of the corresponding norms, and use linear algebra over $\mathbb{F}_2$ to create a square.

But is it good enough for our purposes that the product of the norms is a square? Consider the following example. In $\mathbb{Z}[i]$ we have the elements $2 \pm i$. They both have norm 5, so their product has norm 5^2. However, their product is 5, which is not a square in $\mathbb{Z}[i]$. It should be clear what it is we are throwing away when we look at norms. A particular prime number p in $\mathbb{Z}$ may factor into several prime ideals in an algebraic extension, each having norm a power of p. The norm map lumps all of these prime ideals together. Is there any way for us to tease them apart?

Let us look a little more closely at how we would sieve the integers $N(a-b\alpha)$ as a, b run over integers. If we are trying to detect pairs a, b where $a - b\alpha$ is z-smooth, then we only consider prime numbers p with $p \leq z$. Let us denote by $R(p)$ the set of integers $r \in \{0, 1, \ldots, p-1\}$ with $f(r) \equiv 0 \bmod p$. Then for a, b coprime, p is a divisor of $N(a - b\alpha)$ if and only if $a \equiv br \bmod p$ for some $r \in R(p)$ and $b \not\equiv 0 \bmod p$. Thus, the prime factor p of $N(a - b\alpha)$ comes with a unique *signature*, a particular number r of $R(p)$.

This idea suggests that we may create somewhat more complicated exponent vectors when $a - b\alpha$ is z-smooth and a, b are coprime. For each prime $p \leq z$ and each $r \in R(p)$, define the function $v_{p,r}(a - b\alpha)$ to be the exponent on p in the prime factorization of $N(a - b\alpha)$ if $a \equiv br \bmod p$, and define $v_{p,r}(a - b\alpha)$ to be 0 otherwise. We also include a coordinate for the sign of $N(a - b\alpha)$, so our exponent vectors $v(a - b\alpha)$ will have $1 + \sum_{p \leq z} \#R(p)$ coordinates.

Is it true that if $\prod_{(a,b)\in\mathcal{S}}(a - b\alpha)$ is a square in $\mathbb{Z}[\alpha]$, then $\sum_{(a,b)\in\mathcal{S}} v(a - b\alpha) \bmod 2$ is the zero vector over $\mathbb{F}_2$? Yes it is true, but the proof is not trivial. With some effort, it is shown in [4] that the exponent vector map is well defined on the subgroup $\mathcal{H}$ of $\mathbb{Q}(\alpha)^*$ generated by elements $a - b\alpha$ with a, b coprime rational integers, and in fact this map is a homomorphism. Thus squares have even exponent vectors.

Is the converse true? That is, if $\sum_{(a,b)\in\mathcal{S}} v(a - b\alpha) \bmod 2$ is the zero vector, must $\prod_{(a,b)\in\mathcal{S}}(a - b\alpha)$ be a square in $\mathbb{Z}[\alpha]$? Unfortunately no, as the following two examples show. In $\mathbb{Z}[i]$, consider the associate elements $2 + i$ and $-1 + 2i$. Since they both have positive norm, they have the same exponent vectors: all 0, except $v_{5,3} = 1$. The sum of their exponent vectors has all even coordinates, but $(2 + i)(-1 + 2i) = i(2 + i)^2$ is not a square in $\mathbb{Z}[i]$. Or consider $\mathbb{Z}[5i]$. If the element $5i$ is written as $a - b \cdot 5i$, then $a = 0$, $b = -1$. Thus $0 \in R(5)$ and $v_{5,0}(5i) = 2$ (since $N(5i) = 25 = 5^2$). But $5i$ is not a square in $\mathbb{Z}[5i]$ (nor is it the associate of a square).

Although the answer to our last question was no, in some sense it should not be considered an emphatic no. The condition that $\sum_{(a,b)\in\mathcal{S}} v(a - b\alpha) \bmod 2$ is the zero vector takes us a long way towards concluding that $\prod_{(a,b)\in\mathcal{S}}(a - b\alpha)$ is a square in $\mathbb{Z}[\alpha]$. There are just a few small obstructions keeping us from this promised land, and in the next section we shall show how a certain simple device allows us to overcome these obstructions. For now though, let us assume that this problem is solved and try to put the above ideas together to form a factorization algorithm.

Recall that we wish to construct *two* squares, not just one. We accomplish this with still more complicated exponent vectors. For a coprime pair a, b with *both* $a - bm$ and $a - b\alpha$ being z-smooth, we consider the vector $v(a, b)$ which has the usual exponent vector $v(a - bm)$ in its first $1 + \pi(z)$ coordinates and the exponent vector $v(a - b\alpha)$ in its next $1 + \sum_{p \le z} \#R(p)$ coordinates. If we find a set $\mathcal{S}$ of coprime integer pairs (a, b) with $\sum_{(a,b)\in\mathcal{S}} v(a, b) \bmod 2$ being the zero vector, then both $\prod_{(a,b)\in\mathcal{S}}(a - bm)$ will be a square in $\mathbb{Z}$ and $\prod_{(a,b)\in\mathcal{S}}(a - b\alpha)$ will be a square in $\mathbb{Z}[\alpha]$ (modulo the few obstructions still to be discussed), and we have fulfilled our main goal.

What is the size of the numbers we wish to find smooth? These numbers are the integers $(a - bm)N(a - b\alpha)$ where a, b run over coprime integer pairs with $0 < b \le B$, $|a| \le B$. We have already seen that we would like to have d, the degree of our polynomial, be large to make m small and so make $|a - bm|$ small. The norm form $N(a - b\alpha)$ contains the coefficients of our polynomial f, which are bounded by m, so it may seem that we should like to make d large to keep $N(a-b\alpha)$ small as well. However, $N(a-b\alpha)$ is a homogeneous polynomial in the variables a, b of degree d. Thus, when a, b are at the upper end of their range, $N(a - b\alpha)$ may well be as large as $(d + 1)B^d n^{1/d}$. When multiplied by $a - bm$, the absolute value of the product may well be as large as $(d + 1)B^{d+1}n^{2/d}$. This shows that we should not choose d *too* large.

There is a delicate aspect to our problem concerning the parameter B. We will be sieving over approximately B^2 pairs (a, b), so surely this will be a lower bound for the running time. Thus, we would like to take B small. But if we do so, we may not have found enough pairs (a, b) with $(a - bm)N(a - b\alpha)$ being z-smooth. Further, the larger we take B the larger the numbers $(a-bm)N(a-b\alpha)$ get, and so the less likely they are to be z-smooth.

Nevertheless, it is more or less routine to solve this optimization problem. Let $\varepsilon > 0$ be arbitrary but fixed, let $\beta = (8/9)^{1/3} + \varepsilon$ and let

$$B = \exp(\beta(\log n)^{1/3}(\log\log n)^{2/3}), \quad d = [(2/\beta)^{1/2}(\log n/\log\log n)^{1/3}].$$

Then the expression $x = (d + 1)B^{d+1}n^{2/d}$, which is a bound for the numbers we wish to find smooth, is less than $\exp((8\beta + \varepsilon)^{1/2}(\log n)^{2/3}(\log\log n)^{1/3})$. By Proposition 4.1, it will likely suffice if we have $\exp((1 + \delta)\sqrt{2\log x \log\log x})$ such auxiliary numbers, for some fixed $\delta > 0$. But this expression is less than $\exp((128\beta/9 + 2\varepsilon)^{1/4}(\log n)^{1/3}(\log\log n)^{2/3})$, if we choose δ as an appropriate function of ε. Note that $(128\beta/9 + 2\varepsilon)^{1/4} < 2\beta$, so the B^2 auxiliary numbers we have should indeed be sufficient for the task at hand. Thus, an upper bound for the heuristic running time of the algorithm is about B^2, that is, about $\exp((64/9)^{1/3}(\log n)^{1/3}(\log\log n)^{2/3})$. (Note that the number $(8/9)^{1/3}$ in the expression for β is chosen as the real root v of $(128v/9)^{1/4} = 2v$ and that d is chosen so that $B^d \approx n^{2/d}$. The smoothness bound z is that given by (4.2); it is about B. Some of the inequalities given in the above argument hold only when

n is sufficiently large depending on the choice of ε. For a fuller treatment of the heuristic complexity of the number field sieve, see [4].)

It should be noted that the exponent $(64/9)^{1/3}$ in the heuristic complexity of the number field sieve can be reduced using an idea of Coppersmith [6]. However, this improvement is not likely to prove practical until the numbers factored become *very* large.

8. Obstructions

As we just saw, in the number field sieve we hope that numbers $(a-bm)N(a-b\alpha)$ are smooth, where a, b run over small integers. A bound x for the size of these numbers is about $\exp((\log n)^{2/3})$, where n is the number we are trying to factor. The smallness of this bound, when compared to the analogous bound of about $\sqrt{n}$ with the quadratic sieve algorithm, is so encouraging that it gives us great motivation to overcome any remaining obstructions.

To overcome these obstructions, however, we should at least have an idea of what they are. One problem comes from the fact that the ring $\mathbb{Z}[\alpha]$ may not be the full ring $\mathcal{O}$ of integers in $\mathbb{Q}(\alpha)$. Thus, even if $\sum_{(a,b)\in\mathcal{S}} v(a-b\alpha) \bmod 2$ is the zero vector, we may not have $\prod_{(a,b)\in\mathcal{S}}(a-b\alpha)\mathcal{O}$ being the square of an ideal in $\mathcal{O}$. Even if it is the square of an ideal I in $\mathcal{O}$, it may be that $\mathcal{O}$ is not a principal ideal domain, so I may not be principal. Even if $I=(\gamma)$ is principal, because $\mathcal{O}$ may have a complicated unit group, it may be that $\prod_{(a,b)\in\mathcal{S}}(a-b\alpha) \neq \gamma^2$. There are two more problems, but let us deal with these three first.

These three obstructions deal with three algebraic objects: $\mathcal{O}/\mathbb{Z}[\alpha]$, the class group of $\mathcal{O}$, and the unit group of $\mathcal{O}$. These three groups are each related to a different aspect of the "obstruction group" $\mathcal{G}$. To describe $\mathcal{G}$, first take the subgroup $\mathcal{H}$ of $\mathbb{Q}(\alpha)^*$ mentioned above. Then $\mathcal{G}$ is the subgroup of $\mathcal{H}$ of elements with even exponent vectors modulo the group of squares in $\mathcal{H}$. In the obstruction group every element other than the identity has order 2, and so it may be thought of as an $\mathbb{F}_2$ vector space. In [4] it is shown that the dimension of this vector space is less than $\log n/\log 2$. Surely, such a low-dimensional vector space should not be hard to deal with!

In [1], Adleman suggested a particularly neat device for dealing with this vector space. Suppose l is large compared with $\log n/\log 2$, say $l \approx 3\log n/\log 2$. And say we have l "random" multiplicative maps $\chi_1, \ldots, \chi_l$ from the z-smooth members of $\mathbb{Z}[\alpha]$ to $\{1,-1\}$. A necessary condition for $\prod_{(a,b)\in\mathcal{S}}(a-b\alpha)$ to be the square of an element in $\mathbb{Z}[\alpha]$ is that each $\prod_{(a,b)\in\mathcal{S}}\chi_i(a-b\alpha)$ is equal to 1 for $i=1,\ldots,l$. If we write $v_i(a-b\alpha)$ for the member of $\{0,1\}$ with $\chi_i(a-b\alpha) = (-1)^{v_i(a-b\alpha)}$, then we can enlarge our exponent vectors by these l coordinates and get an even stronger necessary condition for $\prod_{(a,b)\in\mathcal{S}}(a-b\alpha)$ to be a square. Since the three obstructions described above have dimension so much smaller than l, and since the maps $\chi_1, \ldots, \chi_l$ are random, it should be highly likely that we overcome these three obstructions.

So where are we to find these multiplicative maps $\chi_1, \ldots, \chi_l$? We can take these as quadratic characters modulo primes exceeding our smoothness bound z. In particular, if q_i is a prime exceeding z, and s_i is an integer satisfying $f(s_i) \equiv 0 \bmod q_i$, $f'(s_i) \not\equiv 0 \bmod q_i$, then we may take for $\chi_i(a - b\alpha)$ the Legendre symbol $\left(\frac{a-bs_i}{q_i}\right)$ (provided that $a \not\equiv bs_i \bmod q_i$, which will be the case if $a - b\alpha$ is z-smooth). We may consider these choices of χ_i as pseudorandom, and in fact with these characters we can overcome the three mentioned obstructions.

There are just two problems left, one minor and one major. The minor problem is that even if $\prod_{(a,b)\in\mathcal{S}}(a-b\alpha) = \beta^2$ for some $\beta \in \mathcal{O}$, it may be that $\beta \notin \mathbb{Z}[\alpha]$. We solve this by multiplying the square $\prod_{(a,b)\in\mathcal{S}}(a-bm)$ with the square $f'(m)^2$ and multiplying β^2 by $f'(\alpha)^2$. It is not hard to show that for any $\beta \in \mathcal{O}$, we have $f'(\alpha)\beta \in \mathbb{Z}[\alpha]$, so $f'(\alpha)^2\beta^2$ is indeed the square of an element of $\mathbb{Z}[\alpha]$.

So now assume that $f'(\alpha)^2 \prod_{(a,b)\in\mathcal{S}}(a-b\alpha) = \gamma^2$ for some $\gamma \in \mathbb{Z}[\alpha]$. We have worked long and hard to get to this point. But we cannot attempt to factor n unless we can actually find the element γ of $\mathbb{Z}[\alpha]$. This is the major problem mentioned above. Note that this problem also occurs with the quadratic sieve when we locate a set of integers $\mathcal{T}$ with $\prod_{t\in\mathcal{T}}(t^2 - n)$ being a square. But in that case, our exponent vectors give us the prime factorization of the square and so the prime factorization of the square root. Moreover, the square root need only be computed modulo n, so this is not a difficult problem.

We still have exponent vectors for our numbers $a - b\alpha$, but unless we are working in a unique factorization domain it is not so easy to use them. A brute-force method is to multiply out the huge product and somehow take its square root. More subtle methods for solving the "square root problem" are discussed in [4, 9, 19]. Suffice it to say that a subroutine for doing this can be fashioned so that asymptotically this phase of the algorithm takes a negligible amount of time compared with sieving and the linear algebra with the exponent vectors. In practice though, this problem is not trivial, though it should not be considered as a barrier to running the algorithm – see [2].

9. The special number field sieve

For some numbers n it may be possible to find a polynomial $f(t)$ for use in the number field sieve where f has especially small coefficients. For example, in the case $n = F_9 = 2^{512} + 1$, we may take $f(t) = t^5 + 8$ and $m = 2^{103}$. Then $f(m) = 8n \equiv 0 \bmod n$.

When f has such small coefficients, the complexity of the algorithm is reduced. In particular, when the coefficients of f have negligible size, the bound $(d+1)B^{d+1}n^{2/d}$ on the numbers we wish to find smooth, where d is the degree of f and B^2 is the size of the sieve, is reduced to $(d+1)B^{d+1}n^{1/d}$. This in turn reduces the complexity of the number field sieve to $\exp((32/9)^{1/3}(\log n)^{1/3}(\log\log n)^{2/3})$.

There can be other advantages to working with special polynomials such as t^5+8. In particular, if we are working in a unique factorization domain, explicit factorizations of the auxiliary numbers $a - b\alpha$ into prime elements and units can allow us to overcome the obstructions of the previous section without characters.

In addition, such factorizations allow for a simple solution of the square root problem.

It is naturally with the special number field sieve that we have seen the most spectacular factorizations. For more on the experience with this, see [16].

One lesson that the special number field sieve has for the general case is that it is very desirable to find a polynomial with small coefficients. An averaging argument shows that we cannot hope to do very much better than the "base m method" of §7, but that there is room for improvement. One of the simpler improvements we have is to choose m as $\lceil n^{1/(d+1)} \rceil$ and write n in the base m as before. The resulting polynomial is not monic, but this is not a big problem to overcome. No one has yet come up with a substantially better way of choosing a polynomial in the general case.

10. Conclusion

It seems clear that there is a correlation between algorithmic developments and the evolution of computer hardware. Though Lehmer and Powers in 1931 had almost all of the ingredients of the continued fraction factoring algorithm, they did not make the leap to smooth numbers and exponent vectors reduced modulo 2 as did Brillhart and Morrison, almost surely because these ideas are not well suited for the hand computations that Lehmer and Powers did. Similarly, Kraitchik, 50 years before the introduction of the quadratic sieve, had suggested finding a set of integers $\mathcal{T}$ with $\prod_{t \in \mathcal{T}}(t^2 - n)$ a square as a way of factoring n. Because he did not have a large computer to work on, the idea of sieving to discover smooth values of $t^2 - n$ and then combining them via linear algebra mod 2 did not occur to him.

In some sense the number field sieve for general integers may be a little ahead of its time. Although heuristically it is the *asymptotic* champion, it has not yet factored the largest composite number of no special form and with no small prime factor ever factored. This honor still belongs to the quadratic sieve. We believe we are close to the crossover point now, namely 120 to 130 decimal digits. But to factor such numbers takes an enormous amount of computing power, despite our clever algorithms. Given perhaps an order of magnitude improvement of the speed (and size) of computers, I believe the number field sieve would emerge as the clear method of choice for the hardest numbers.

It is perhaps a little too self-serving, though, to say that others should first improve their aspects of the problem. Rather than sitting on our hands, let us instead find ways to improve the number field sieve now, or perhaps even find a wholly new factorization method. At one point some suggested that $\exp(\sqrt{\log n \log\log n})$ might be the true complexity of factoring, only because we had several different methods with this complexity and none faster. Now we have the number field sieve and it has heuristic complexity of the shape $\exp(c(\log n)^{1/3}(\log\log n)^{2/3})$. Is this now the limit of what we can do? It may be, but it is unlikely an advance will be made by people who think they cannot succeed.

REFERENCES

1. L. M. Adleman, *Factoring numbers using singular integers*, Proc. 23rd Ann. ACM Symp. on Theory of Computing (STOC) (1991), 64-71.
2. D. J. Bernstein and A. K. Lenstra, *A general number field sieve implementation*, in [13], 103-126.
3. J. Brillhart, M. Filaseta and A. Odlyzko, *On an irreducibility theorem of A. Cohn*, Canad. J. Math. **33** (1981), 1055-1059.
4. J. P. Buhler, H. W. Lenstra, Jr. and C. Pomerance, *Factoring integers with the number field sieve*, in [13], 50-94.
5. E. R. Canfield, P. Erdős and C. Pomerance, *On a problem of Oppenheim concerning "factorisatio numerorum"*, J. Number Theory **17** (1983), 1-28.
6. D. Coppersmith, *Modifications to the number field sieve*, J. Cryptology (to appear); also see IBM Research Report #RC 16264, Yorktown Heights, NY, 16 pages, November 1990.
7. ______, *Solving linear equations over $GF(2)$: block Lanczos algorithm*, Linear Algebra and its Applications **192** (1993), 33-60; also see IBM Research Report #RC 16997, July 1991, and see Proceedings of Essen Workshop on Linear Algebra, July 1992.
8. ______, *Solving homogeneous linear equations over $GF(2)$ via block Wiedemann algorithm*, Math. Comp. **62** (1994), 333-350.
9. J.-M. Couveignes, *Computing a square root for the number field sieve*, in [13], 95-102.
10. J. A. Davis and D. B. Holdridge, *Factorization using the quadratic sieve algorithm*, Sandia Report Sand 83-1346, Sandia National Laboratories, Albuquerque, NM (1983).
11. D. Gordon, *Discrete logarithms in $GF(p)$ using the number field sieve*, SIAM J. Discrete Math. **6** (1993), 124-138.
12. B. A. LaMacchia and A. M. Odlyzko, *Solving large sparse linear systems over finite fields*, in Advances in Cryptology – CRYPTO 90 (A. J. Menezes and S. A. Vanstone, eds.), Lecture Notes in Computer Science **537**, Springer-Verlag, Berlin, 1991, 109-133.
13. A. K. Lenstra and H. W. Lenstra, Jr. (eds.), *The development of the number field sieve*, Lecture Notes in Mathematics **1554**, Springer-Verlag, Berlin, 1993.
14. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), 515-534.
15. A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse and J. M. Pollard, *The factorization of the ninth Fermat number*, Math. Comp. **61** (1993), 319-349.
16. ______, *The number field sieve*, in [13], 11-42, Extended abstract: Proc. 22nd Ann. ACM Symp. on Theory of Computing (STOC) (1990), 564-572.
17. H. W. Lenstra, Jr., J. Pila and C. Pomerance, *A hyperelliptic smoothness test*, I, Phil. Trans. Royal Soc. Ser. A (to appear).
18. H. W. Lenstra, Jr. and C. Pomerance, *A rigorous time bound for factoring integers*, J. Amer. Math. Soc. **5** (1992), 483-516.
19. P. L. Montgomery, *Square roots of products of algebraic numbers*, preprint, August 6, 1993.
20. A. M. Odlyzko, *Discrete logarithms in finite fields and their cryptographic significance*, in Advances in Cryptology – Proceedings of EUROCRYPT 84 (T. Beth et al., eds.), Lecture Notes in Computer Science **209**, Springer-Verlag, Berlin, 1985, 224-314.
21. C. Pomerance, *Analysis and comparison of some integer factoring algorithms*, in Computational methods in number theory (H. W. Lenstra, Jr. and R. Tijdeman, eds.), Mathematical Centre Tracts **154/155**, Mathematisch Centrum, Amsterdam, 1982, 89-139.
22. ______, *The quadratic sieve factoring algorithm*, in Advances in Cryptology – Proceedings of EUROCRYPT 84 (T. Beth et al., eds.), Lecture Notes in Computer Science **209**, Springer-Verlag, Berlin, 1985, 169-182.
23. ______, *Fast, rigorous factorization and discrete logarithm algorithms*, in Discrete algorithms and complexity (D. S. Johnson et al., eds.), Academic Press, Orlando, 1987, 119-143.

24. C. Pomerance and J. W. Smith, *Reduction of huge, sparse matrices over a finite field via created catastrophes*, Experimental Math. **1** (1992), 90-94.
25. O. Schirokauer, *On pro-finite groups and on discrete logarithms*, Ph.D. thesis, University of California, Berkeley, May 1992.
26. D. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory **32** (1986), 54-62.

Department of Mathematics
University of Georgia
Athens, Georgia 30602 U.S.A.
e-mail: carl@ada.math.uga.edu