NTNU
Norwegian University of
Science and Technology

Special number field sieve

Per Reidar Bøhler

Master of Science in Teacher education with Realfag
Submission date: June 2008
Supervisor: Kristian Gjøsteen, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem description

The candidate is to study the special number field sieve and the mathematical background of this integer factorization algorithm.

Acknowledgments

I will most of all thank associate professor Kristian Gjøsteen who has been an excellent supervisor for me on this master thesis. He has given me great advice, help, and motivated me to work. Thank you very much Kristian. I would also like to thank professor Øyvind Solberg who has been a great inspiration and always been available to answer algebra questions in the time preceding this master thesis..

Abstract

Integer factorization is a problem not yet solved for arbitrary integers. Huge integers are therefore widely used for encrypting, e.g. in the RSA encryption scheme. The special number field sieve holds the current factorization record for factoring the number $2^{1039} + 1$. The algorithm depends on arithmetic in an algebraic number field and is a further development from the quadratic sieve factoring algorithm. We therefor present the quadratic sieve as an introduction to the ideas behind the special number field sieve first. Then the special number field is described. The key concepts is evaluated one bye one. Everything is illustrated with the corresponding parts of an example factorization. The running time of the special number field sieve is then evaluated and compared against that of the quadratic sieve. The special number field sieve only applies to integers of a special form, but a generalization has been made, the general number field sieve. It is slower but all estimates suggests it is asymptotically faster than all other existing general purpose algorithms.

Contents

1 Introduction **1**

2 Quadratic Sieve **3**
2.1 The quadratic sieve algorithm 3
2.2 Example 4
2.3 Running time 5

3 Mathematical background **9**

4 Special number field sieve **12**
4.1 Introduction 12
4.2 The special form of n 14
4.3 Factor base 14
4.4 Sieving 17
4.5 Linear algebra 20
4.6 Square roots 21
4.7 Example summary 22
4.8 Running time 23

5 Discussion **25**
5.1 General number field sieve 26

6 Concluding remarks **27**

1 Introduction

Integer factorization is a simple mathematical concept, yet it is still a subject far from solved. That is, anyone can create(with the help of a computer) a composite integer large enough so that not even the brightest mathematicians can factor it. This makes large composite integers very useful in the world of cryptography, e.g. the RSA crypto system depends totally on the fact that it is impossible to factor the number n that encrypts the message.

During the years there has been proposed several algorithms for factoring composite integers in a more or less efficient way. The continued fraction method was introduced by Lehmer and Powers, and later in the 1970's refined for implementing by Brillhart and Morrison [9]. This algorithm allowed complete factorization of integers up to about 50 digits. Carl Pomerance introduced the quadratic sieve algorithm in the early 1980's, this algorithm pushed the limit for integer factorization up above 100 digits. Then in 1988 John Pollard introduced the idea for the number field sieve. This idea was further developed, and in 1990 A.Lenstra, H.Lenstra, Manasse, and Pollard introduced the special number field sieve. The algorithm relies heavily on arithmetic in number fields and traditional sieving techniques, hence the name number field sieve. Then in 1993 Buhler, H.Lenstra, and Pomerance introduced the general number field sieve which generalizes the special number field sieve to all integers.

The above described algorithms are in the group of index calculus algorithms for integer factorization. That is they all use the same idea and approach to achieve a non trivial factorization. They follow a three step procedure to find two integers that will give a factorization. Consider a number n that is composite, and has two factors a and b, then the number n can be written in the following way:

$$n = a \cdot b = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2.$$

Now replace $\frac{a+b}{2}$ by x and $\frac{a-b}{2}$ by y, this gives

$$\begin{aligned} n = x^2 - y^2 &= (x-y)(x+y) \\ x^2 &\equiv y^2 \pmod{n} \\ (x-y)(x+y) &\equiv 0 \pmod{n}. \end{aligned} \tag{1}$$

The last equation implies $n|(x-y)(x+y)$. So either $x \pm y = 1$(and the other is n) or $x \pm y$ contains a non trivial factor of n. If we then use the Euclidean algorithm to compute the greatest common divisor(gcd) of n and $x \pm y$ maybe we can find that non-trivial factor of n.

This means if x and y are known and we compute the gcd, we have two possible results, either the $\gcd(n, x \pm y)$ is 1 and n, or we get a non-trivial factor of n. That is there is approximately a 50% chance to factor n if x and y that satisfy (1) are known.

The three steps the index calculus algorithms use to find the proper integers x and y are, first find a suitable factor base, second use a sieving technique to search for integers of a special form that factors in the factor base. Then the third step is to use linear algebra to find a set of integers, from the set found during the sieving, that can be combined to be on the form of (1)

The difference of the quadratic sieve and the number field sieve is mostly in the sieving step. The sieving of the number field sieve is more effective than the technique of the quadratic sieve when the integer to factor gets large. The special number field is the asymptotically fastest algorithm, but it only applies to integers of a special form. The general number field sieve uses the same ideas as the special number field sieve but it must take several precautions due to the generality of n which causes the running time to be slower. The quadratic sieve also applies to general integers but is again slower than the other two. The record for the general number field sieve is the factorization of the RSA-200 number which has 663 bits . This is the largest integer factored by a general purpose algorithm [3].

The special number field sieve is constructed to factor large integers on the form $r^e - s$ where r and s are relatively small and e is large. To find integers that satisfy (1) the algorithm uses an irreducible polynomial, number fields, and a homomorphism. The first major and perhaps the most famous factorization by the special number field sieve was the factoring of the ninth Fermat number $2^{512} + 1$ [8]. The current record factorization by the special number field sieve is the factoring of the integer $2^{1039} - 1$ (313 digits), into 3 primes each with 7, 80, and 227 digits [3].

In section 2 there will be an introduction to the quadratic sieve, the algorithm will be described and a heuristic time estimate is analyzed. Then in section 3 the mathematical background of the special number field sieve will be described, before section 4 introduces the idea of the special number field sieve. How to construct the factor base and the sieving techniques is then described with corresponding results from section 3. Then the running time estimates proposed for the special number field sieve are presented. Section 5 discusses different aspects of the algorithm and section 6 concludes with a summary of the thesis.

2 Quadratic Sieve

When the quadratic sieve algorithm was introduced in the early 1980's it was the fastest integer factorization algorithm available. It still is the preferred(and fastest) algorithm to factor numbers up to about 100 digits.

2.1 The quadratic sieve algorithm

The quadratic sieve starts it search for the integers x and y that will satisfy (1) by computing squares of integers and their residues modulo n. Then it hopes that the residues modulo n is composed by relatively small primes, or at least that the residue has few prime factors. If the integers one chooses are close to $\sqrt{kn}$ $k \in \mathbb{Z}$, their residues will be small modulo n when they are squared and then have a better chance of having smaller prime factors. In practice one sets an upper bound and try to find residues that are composed by primes less than this bound. Let's say B is the bound we have set, and we try to find integers on this form

$$x_i^2 \equiv a_i \pmod{n}. \tag{2}$$

In which the a_i's have all its prime factors less than B. When a number has all prime factors less than some integer B we say it is B-smooth, e.g. the quadratic sieve wants to find squares that modulo n are B-smooth.

What we now want is to find several relations like (2) and combine them to get a square on both sides of the congruence, then we could just find the gcd and hopefully it will be a non-trivial factor of n as described in section 1.

The idea is to find a combination of the a_i's that is a square when multiplied together. This means when added up each prime factor of the a_i's appears an even number of times. The other side (x_i^2) is already a square and will of course stay that way when they are multiplied together.

To achieve this combination(a square) we will use theorems from basic linear algebra. If we let β denote the number of primes less than B, that is the number of primes in the factor base, each a_i that are B-smooth are on the form

$$a_i = \prod_{j=1}^{\beta} p_j^{k_j}, \ k \in \mathbb{Z}$$

where $p_j \leq B$ and prime. Our relations can be written as

$$x_i^2 \equiv \prod_{j=1}^{\beta} p_j^{k_{i,j}} \pmod{n}$$

For each integer x_i^2 let $\epsilon_i = (k_{i,1}, k_{i,2}, ..., k_{i,\beta})$. If we now reduce each ϵ_i modulo 2, we only need to find a combination of these vectors that sums to the zero vector in $\mathbb{F}_2^\beta$. This is the same as finding a linear dependent set of vectors in $\mathbb{F}_2^\beta$, and to be guaranteed a linear dependent set we need at least $\beta + 1$ vectors, that is $\beta + 1$ B-smooth residues.

If we can find at least $\beta + 1$ smooth residues we reduce the corresponding ϵ_i's modulo 2, and put them into a matrix. This matrix will have β columns and at least $\beta + 1$ rows. Then it is just to start row reducing until a zero row is obtained. If we now call the set of residues that results in a zero row in the matrix S, the corresponding ϵ vectors for the residues in S will satisfy

$$\sum_{p_j \leq B} \epsilon_i(p_j) \equiv 0 \pmod 2. \tag{3}$$

$\epsilon_i(p_j)$ is the exponent of p_j for the corresponding a_i. Then let e denote the sum of all the $\epsilon_i, i \in S$, that is $e = \sum_{i \in S} \epsilon_i$. And because of (3) the vector $\frac{1}{2}e$ will still have integers on all coordinate positions.

We can now write the product of all relations in the set S as:

$$\prod_{a_i \in S} a_i = \prod_{j=1}^{\beta} p_j^{e_j}$$

And the the integer y in (1) is

$$y = \prod_{j=1}^{\beta} p_j^{\frac{1}{2}e_j}.$$

Then we can combine the values found to achieve

$$x^2 \equiv \prod_{x_i \in S} x_i^2 \equiv \prod_{j=1}^{\beta} p_j^{e_j} \equiv y^2 \pmod n$$

Since we know all the factors of the x_i^2's and the factors p of the a_i's it is straight forward to find the square roots x and y. Then use Euclid's algorithm to find the $\gcd(n, x \pm y)$, which hopefully gives a nontrivial factor of n, if not get another set S and try again.

2.2 Example

Factor the number $n = 260101$ using quadratic sieve.

We chose the factor base to be the primes below 14.

Then we sieved for smooth integers, starting with $\lfloor\sqrt{n}\rfloor + 1$(the least integer greater than $\sqrt{n}$) and the next 40 integers, then $\sqrt{2n}$ and so on. We did Gaussian elimination on the exponent vectors of the smooth integers and found two integers that multiplied together is a square:

$$539^2 \equiv 30240 \equiv 2^2 \cdot 3^2 \cdot 5 \cdot 13^2 \pmod{260101}$$

$$1023^2 \equiv 6125 \equiv 5^3 \cdot 7^2 \pmod{260101}$$

This gives

$$x^2 \equiv (539 \cdot 1023)^2 \equiv 551397^2 \equiv 31195^2 \pmod{260101}$$

and

$$y^2 \equiv (2 \cdot 3 \cdot 5^2 \cdot 7 \cdot 13)^2 \equiv 13650^2 \pmod{260101}$$

$$\gcd(n, x - y) = \gcd(260101, 31195 - 13650) = 29$$

$$\gcd(n, x + y) = \gcd(260101, 31195 + 13650) = 8969$$

$$29 \cdot 8969 = 260101$$

2.3 Running time

There are two major steps that the quadratic sieve running time is affected by.

1. Searching for squares with residues that are B-smooth modulo n
2. Finding a linear dependent set

The running time depends of course on the size of n, but there is nothing we can change about n. The only thing we can change in the algorithm is the size of the factor base. So we will try to find the optimal choice of the factor base depending on the size of n. We will first analyze the time it takes to find the smooth integers, then the running time of the linear algebra. Both will only be rough heuristic estimates where we will make many simplifications.

Sieving

First consider the probability to find a number that is B-smooth. It is customary to denote $\psi(n, B)$ as the number of integers less than n that is only divisible by primes less than B. We are now looking for integers on the form

$$\prod_{i=1}^{\beta} {p_i}^{\alpha_i} \leq n$$

where $p_i \leq B$ is prime. By taking logarithms on both sides we get

$$\sum_{i=1}^{\beta} \alpha_i \cdot \log p_i \leq \log n$$

The primes p_i are in fact not that much smaller than the number B, and their logarithms will be about the same, that is B and the p_i have almost the same number of digits. Therefore we make the simplification to replace $\log p_i$ by $\log B$. And then if we denote the ratio $\frac{\log n}{\log B}$ by u(which gives u a much smaller value than B) we achieve

$$\sum_{i=1}^{\beta} \alpha_i \leq u$$

Then as our last simplification we replace β by B. This means we put in non-trivial terms but they actually cancel out to make no difference. We then find that the number of integers less than n that factors by the primes less than B are roughly the number of integer solutions to the inequality

$$\sum_{i=1}^{B} \alpha_i \leq u$$

This number can be computed from the fact that the number of nonnegative integer B-tuples α_i such that $\sum_{i=1}^{B} \alpha_i \leq u$ is the binomial coefficient $(\frac{[u]+B}{B})$, where [ ] denotes the greatest integer function[7].

So the probability that a number less than n factors by primes less than B is

$$\frac{\psi(n, B)}{n} \approx \frac{\left(\frac{[u]+B}{B}\right)}{n} = \frac{([u]+B)!}{[u]!B!n}$$

If we then take logarithms, replace $\log n$ by $u \log B$ and use the fact that $\log(n!) \approx n \log n - n$ [7], we can estimate the probability to

$\log\left(\frac{([u]+B)!}{[u]!B!n}\right) = \log([u]+B)! - \log([u]!B!n) = ([u]+B)\log([u]+B) - ([u]+B) - [u]\log[u] + [u] - B\log B + B - u\log B$
$\approx [u]\log([u]+B) + B\log([u]+B) - [u]\log[u] - B\log B - u\log B$

If we now replace $[u]$ by u, and because u is assumed to be much smaller than B we replace $\log(u+B)$ by $\log B$ we get
$u\log B + B\log B - u\log u - B\log B - u\log B = -u\log u$

$$\log\left(\frac{\psi(n,B)}{n}\right) \approx -u\log u$$

$$\frac{\psi(n,B)}{n} \approx u^{-u}$$

So to find one B-smooth number we have to check approximately u^u integers on average.

The next question is how much time will be spend on checking each integer? To check one integer we can do trial division to see if the integer is completely factored by the factor base. If the integer is k bits and the integer to divide by is l bits($l \leq \log B$), each division will take O(kl), so the whole test for each integer will take O(klB)

Then to find at least $(\beta+1)$ smooth integers the sieving step will take $u^u(\beta+1)O(klB)$. Using the prime number theorem that $\beta \approx \frac{B}{\log B}$ we get that the running time of the sieving will be

$$O(u^u kB^2)$$

Finding a linear dependent set

The second part to row reduce a matrix, assuming a size of approximately B^2, can be done in at most $O(B^3)$ with Gaussian elimination, but there are faster methods available. We know k is at most $\log n$, combining the relations to get the desired squares can be done in time $O(k^r)$, so the second part will be done in time $O(k^r B^j)$ for appropriate choices r and j.

Best choice of B

The above results give a total running time of

$$O(ku^uB^2 + k^rB^j) = O(k^r u^u B^j) = O(k^r \left(\frac{k}{l}\right)^{\frac{k}{l}} e^{jl}). \tag{4}$$

We would like to find the optimal choice of B, or equivalently the choice of l for which this estimate is minimal. Since k is constant there is nothing we can do about it so we will just ignore it and calculate the derivative of the logarithm of the expression involving l:

$$0 = \frac{d}{dl}\left(\frac{k}{l}\log k - \frac{k}{l}\log l + jl\right) = -\frac{k}{l^2}\log k + \frac{k}{l^2}\log l - \frac{k}{l^2} + j$$

$$0 = -\frac{k}{l^2}(\log\frac{k}{l} + 1) + j \approx -\frac{k}{l^2}(\log\frac{k}{l}) + j \tag{5}$$

If we then choose l such that lj is equal to $\frac{k}{l}\log\frac{k}{l}$, that is the two factors of $\left(\frac{k}{l}\right)^{\frac{k}{l}} e^{jl}$ are approximately equal. j is still constant so we can take l^2 to have the same order of magnitude as $k(\log k - \log l)$, which means l has a value between $\sqrt{k}$ and $\sqrt{k \cdot \log k}$. That gives $\log l \approx \frac{1}{2}\log k$. Which gives us the optimal value of B,

$$B \approx e^{\sqrt{\log n}} \tag{6}$$

Substituting this value for $\log l$ into (5) gives

$$j = \frac{k}{2l^2}\log k$$

$$l \approx \sqrt{\frac{k}{2j}\log k}$$

Using this value of l, and the approximation that $e^{jl} \approx \frac{k}{l}log\frac{k}{l}$, our time estimate from (4) comes to $O(e^{2jl}) = O(e^{2j\sqrt{\frac{k}{2j}log\ k}}) = O(e^{\sqrt{2jk\cdot log\ k}})$

Substituting the constant $\sqrt{2j}$ with C gives us the final running time approximation for the quadratic sieve to be

$$O(e^{C\sqrt{k\log k}})$$

Remarks

This estimate for the running time of the quadratic sieve was a very rough one, there are several aspects in the above argument which can be improved.

The sieving part assumes all integers has the same probability of being smooth, that is the smooth integers are uniformly spread out in the interval [1,Y], and that one has to check all integers in no special order. The sieving

step is done more effectively by only checking values that are close to $\sqrt{kn}$ for small integer values of k as described in section 2.2. These integers will have small residues modulo n and then a bigger chance of being B-smooth. One will then in practice having to check far less than the calculated u^u integers to find a smooth integer. This will play a significant role in the running time of the algorithm as it will find the relations much faster than the time estimate suggests.

As for the linear algebra part, there are faster algorithms than the method of Gaussian elimination, e.g. the conjugate gradient method and the Lanczos method which will speed up things at this stage of the algorithm [2]

The choice of factor base is what we can adjust here, and the consequences of a to small/large factor base are as follows: A small factor base will make it very hard to find the relations in the first step, it might not even find enough of them. But when it does the second step of linear algebra will be very fast as the matrix will be small. A large factor base will make sure the relations will be easy to find, you'll need more of them but as they come in fast you'll probably get enough of them. But the linear algebra will be time consuming since the matrix will be very large. So the time gained in step one might be lost in step two.

We found something that seemed like a reasonable choice for B in our time estimate but other aspects should be considered to. E.g. the choice should also be made according to what resources one has available. The linear algebra will require more fast memory on a computer than the sieving part. So if the resources says less memory, more hard disk space one can to avoid a bottleneck in step two, shift some of the workload over to step 1 by choosing the factor base smaller, or vice versa.

3 Mathematical background

Before we describe the special number field sieve we will go through some mathematical results that the special number field sieve rely upon.

Let $f(x) = x^d - t$ be an irreducible polynomial over $\mathbb{Q}$, $t \in \mathbb{Z}$. Let α be such that $f(\alpha) = 0$. $K = \mathbb{Q}(\alpha)$ is then an algebraic extension, a number field, and $\mathbb{Z}[\alpha]$ is a subring of K

Norm

The elements in $\mathbb{Q}(\alpha)$ can be expressed as $\sum_{i=0}^{d-1} q_i \alpha$, where $q_i \in \mathbb{Q}$. If we associate each element with a vector consisting of the d components

$q_0, q_1, ..., q_{d-1}$, then addition and subtraction is just vector addition and vector subtraction. From the fact that $\alpha^d = z, z \in \mathbb{Z}$ we can interpret multiplying an element in the field by an element $\delta = \sum_{i=0}^{d-1} q_i \alpha$ as multiplying the column vector by a $d \times d$ matrix Δ with rational entries, corresponding to δ. See [8, 4.1] for an example of this. The norm $\mathbf{N}(\delta)$ is then defined to be the determinant of this matrix Δ. We can then easy see that the norm is multiplicative, let $\kappa \in \mathbb{Q}(\alpha)$, $\mathbf{N}(\delta\kappa) = \mathbf{N}(\delta)\mathbf{N}(\kappa)$ since $\delta\kappa$ is the product of two matrices belonging to δ and κ.

If we use the fact that the norm is multiplicative, set $\delta = \kappa^{-1}$ and the fact that $\mathbf{N}(1) = 1$ we see that $\mathbf{N}(\delta) \neq 0$ whenever $\delta \neq 0$

Considering an element δ in $\mathbb{Z}[\alpha]$, the associated matrix Δ will have integer entries, and the determinant will be an integer as well. Note also that the $|\mathbf{N}(\delta)|$ can be interpreted as the index of the subgroup $\delta\mathbb{Z}[\alpha] = \{\delta\rho : \rho \in \mathbb{Z}[\alpha]\}$ in $\mathbb{Z}[\alpha]$ which is finite when $\delta \neq 0$. That is

$$|\mathbf{N}(\delta)| = \#(\mathbb{Z}[\alpha]/\delta\mathbb{Z}[\alpha]). \tag{7}$$

This is for $\delta \in \mathbb{Z}[\alpha]$ and $\delta \neq 0$. For further reference on this result see [8, 4].
Definition 1. The norm of an ideal $\mathbf{I} \subset \mathbb{Z}[\alpha]$, $\mathbf{I} \neq (0)$ is the positive integer $\#(\mathbb{Z}[\alpha]/\mathbf{I})$.

From the above it follows that this is finite, and that the norm of an ideal is equal to the norm of the element generating the ideal.

We recall that an ideal $\mathbf{I}$ is a prime ideal if

$$\forall xy \in \mathbf{I} \Leftrightarrow x \in \mathbf{I} \cup y \in \mathbf{I}. \tag{8}$$

Definition 2. A first degree prime ideal, is a prime ideal $\mathbf{I}$ with norm p, where p is prime.

Let $\mathbf{I}$ be a first degree prime ideal of norm p, then it is possible to construct a ring homomorphism

$$\theta : \mathbb{Z}[\alpha]/\mathbf{I} \to \mathbb{Z}/p\mathbb{Z} \tag{9}$$
$$\alpha \to c \bmod p$$

Where c is a root of $f(x)$ in $\mathbb{Z}/p\mathbb{Z}$, equivalently $f(c) \equiv 0 \pmod p$

The pair $(p, c \bmod p)$ corresponds to the first degree prime ideal $\mathbf{I}$, in fact the first degree prime ideals and the pairs $(p, c \bmod p)$ are in bijective correspondence with each other. This follows from two lemmas.
Lemma 1. *Let p be a prime integer. If $f(c) \equiv 0 \pmod p \Rightarrow \boldsymbol{I} = \langle p, c - \alpha \rangle$ is a first degree prime ideal*

Proof. Let ϕ be a mapping defined as follows

$$\phi : \mathbb{Z}[\alpha]/\mathbf{I} \to \mathbb{Z}/p\mathbb{Z}$$

$$\alpha \to c$$

$$z \to z \pmod{p}$$

for $\alpha \in \mathbb{Z}[\alpha]$, $z \in \mathbb{Z}$, and $c \in \mathbb{Z}/p\mathbb{Z}$ such that $f(c) \equiv 0 \pmod{p}$

We now want to show that ϕ is an isomorphism, which will prove the lemma. First let $x, y \in \mathbb{Z}[\alpha]/\mathbf{I}$. Then we see that $\phi(x+y) = (x+y)+\mathbf{p} = (x+\mathbf{p})+(y+\mathbf{p}) = \phi(x)+\phi(y)$ and $\phi(xy) = xy+\mathbf{p} = (x+\mathbf{p})(y+\mathbf{p}) = \phi(x)\phi(y)$. So we have that ϕ is a ring homomorphism

Now let $y \in \mathbb{Z}/p\mathbb{Z}$, we want to find an $x \in \mathbb{Z}[\alpha]/\mathbf{I}$ such that $\phi(x) = y$. We know that $1 \in \mathbb{Z}[\alpha]/\mathbf{I}$ and since ϕ is a homomorphism $\phi(1) = 1$. Then $\phi(y) = \phi(1 \cdot y) = y\phi(1) = y$ so $x = y$ will do the job and ϕ is onto.

Again let $x \in \mathbb{Z}[\alpha]/\mathbf{I}$ but assume that $\phi(x) = 0$. Since $x \in \mathbb{Z}[\alpha]/\mathbf{I}$ it is of the form $a_0 + a_1\alpha + a_2\alpha^2 + \ldots + a_n\alpha^n, a_i \in \mathbb{Z}$. If we then take an element $x' = a_0 + a_1 c + a_2 c^2 + \ldots + a_n c^n \in \mathbb{Z}[\alpha]/\mathbf{I}$, add and subtract it from x we see that $x = x - x' + x' = k_1 p + k_2(c - \alpha)$ for $k_1, k_2 \in \mathbb{Z}[\alpha]$ is an element of $\mathbf{I}$, and we can conclude that ϕ is 1-1.

We have then proved that ϕ is a homomorphism, onto, and 1-1, hence ϕ is an isomorphism □

Lemma 2. *Let p be a prime integer. If* $\boldsymbol{I} = \langle p, c - \alpha\rangle$ *and* $\boldsymbol{I'} = \langle p, c' - \alpha\rangle \Rightarrow$ *either* $\boldsymbol{I} \neq \boldsymbol{I'}$*(and* $c \not\equiv c' \pmod{p}$*) or* $\boldsymbol{I} = \boldsymbol{I'}$*(and* $c \equiv c' \pmod{p}$*)*

Proof. Consider $(c - \alpha) - (c' - \alpha) = c - c'$, this has two options, either $c - c' = 0$ or $c - c' \neq 0$. First case $c - c' = 0$ gives $c = c'$ and then of course $\mathbf{I} = \mathbf{I'}$ and $c \equiv c' \pmod{p}$.

Second case $c - c' \neq 0$ gives $c \neq c'$. Then we can evaluate $\gcd(c - c', p)$. Since p is prime this has to be equal to 1 or p.

First let $\gcd(c - c', p) = 1$ and assume $\mathbf{I} = \mathbf{I'}$. Then $p, c - \alpha, c' - \alpha$ are all in the same ideal which also means 1 is in the ideal. This is a contradiction since $\mathbf{I}$ would then be the whole ring. We can then conclude $\mathbf{I} \neq \mathbf{I'}$ and $c \not\equiv c' \pmod{p}$ when $c \neq c'$ and $\gcd(c - c', p) = 1$.

Now assume $\gcd(c - c', p) = p$, then $c - c' = (c - \alpha) - (c' - \alpha) = kp$, k$\in \mathbb{Z}$. This gives $c - \alpha = c' - \alpha + kp$, and since $c' - \alpha + kp \in \mathbf{I'}$, $c - \alpha$ is also in $\mathbf{I'}$ and $\mathbf{I} \subseteq \mathbf{I'}$. Equivalently $c' - \alpha = c - \alpha + kp$, and since $c - \alpha + kp \in \mathbf{I}$, $c' - \alpha \in \mathbf{I}$ and $\mathbf{I'} \subseteq \mathbf{I}$. We can then conclude that $\mathbf{I} = \mathbf{I'}$ and $c \equiv c' \pmod{p}$ if $c - c' \neq 0$ and $\gcd(c - c', p) = p$. □

It is now clear that the pairs $(p, c \bmod p)$ are in bijective correspondence with the first degree prime ideals $\mathbf{I}$, and $\mathbf{I}$ is generated by the elements $(p, c - \alpha)$.

We can also use the map θ to check whether a given element in $\mathbb{Z}[\alpha]$ is contained in a first degree prime ideal $\mathbf{I} = \langle p, c - \alpha \rangle$. This is because

$$\sum_{i=0}^{d-1} a_i \alpha^i \in \mathbf{I} \Leftrightarrow \sum_{i=0}^{d-1} a_i c^i \equiv 0 \pmod{p} \tag{10}$$

It should now be clear that 'an element $\pi_p = \sum_{i=0}^{d-1} a_i \alpha^i \ a_i \in \mathbb{Z}$ of $\mathbb{Z}[\alpha]$ generates a first degree prime ideal corresponding to a pair $(p, c \bmod p)$ if and only if $\mathbf{N}(\pi_p) = \pm p$ and $\sum_{i=0}^{d-1} a_i c^i \equiv 0 \bmod p$ [1, 3]'

To describe the factorization of $a + b\alpha$ in $\mathbb{Z}[\alpha]$ we have the following lemma

Lemma 3. *Let $a,b \in \mathbb{Z}$, $\gcd(a, b) = 1$. Then all prime ideals $\mathbf{p}$ that occur in $a + b\alpha$ are first degree prime ideals.*

Proof. Assume that $\mathbf{p}$ occurs in $a + b\alpha$, and let $\mathbf{p}$ be the kernel of a ring homomorphism $\psi : \mathbb{Z}[\alpha] \to \mathbb{F}$, where $\mathbb{F}$ is a finite field. Suppose also that the characteristic of $\mathbb{F}$ is p, such that the field $\mathbb{F}_p$ is a subfield of $\mathbb{F}$.

Since $(a + b\alpha) \in \mathbf{p} \Rightarrow \psi(a + b\alpha) = 0$ which again gives

$$\psi(a) = -\psi(b)\psi(\alpha)$$

Now it is easy to see that since both $a, b \in \mathbb{Z} \Rightarrow \psi(a) \in \mathbb{F}_p$ and $\psi(b) \in \mathbb{F}_p$

Suppose that $\psi(b) = 0$, that means $\psi(a) = 0$ also, which means $p|a$ and $p|b$, this implies that $p|\gcd(a, b)$ which is a contradiction since $\gcd(a, b) = 1$. So $\psi(b) \neq 0$

We can then conclude that the element $\psi(\alpha) = \frac{-\psi(a)}{\psi(b)} \in \mathbb{F}_p$

This shows that the ring homomorphism ψ maps all elements from $\mathbb{Z}[\alpha]$ to $\mathbb{F}_p$, and $\mathbf{p}$ is the kernel of ψ which by definition means it is a first degree prime ideal. □

4 Special number field sieve

4.1 Introduction

The special number field sieve algorithm is roughly described as follows. Given a large composite integer $n = r^e - s$ that is not a power of a prime, start by finding a monic polynomial f in such a way that it is irreducible

over $\mathbb{Q}$ and that there exist some $m \in \mathbb{Z}$ such that $f(m) \equiv 0 \pmod{n}$. This polynomial can be constructed in the following way, decide the degree d of the polynomial, let k be the least positive integer such that $kd \geq e$. Then let $t = s \cdot r^{k \cdot d - e}$. Now $f(x) = x^d - t$, and $m = r^k$ satisfies $f(m) \equiv 0 \pmod{n}$. Find an element α such that $f(\alpha) = 0$, and set up an homomorphism ϕ

$$\phi : \mathbb{Z}[\alpha] \to \mathbb{Z}/n\mathbb{Z}$$

$$\alpha \to m \pmod{n}$$

Then decide on a factor bound B_1 such that the factor base B' consists of primes less than B_1. A second bound B_2 such that the factor base B'' consists of the primes less than B_2 that together with an integer c correspond to a first degree prime ideal in $\mathbb{Z}[\alpha]$ as proved by (Lemma1) and (Lemma2).

The idea is to get a relation like (1) by creating a square on one side of the congruence by means of combining B_1-smooth integers. The other side is to consist of elements in $\mathbb{Z}[\alpha]$, and we want to find a square composed of generators of first degree prime ideals.

We sieve for smooth integers of the form $a + bm$ and $a + b\alpha$, call this set of smooth pairs (a, b) T. Then we need to find a set of the pairs $(a, b) \in T$ that when multiplied together becomes a square of integers $(a+bm)$ and a square of elements in $\mathbb{Z}[\alpha]$ $(a + b\alpha)$, call this set S. The set S is found just as in the quadratic sieve, put the exponents of the pairs $(a, b) \in T$ in a vector ϵ(here we include exponents of both $a + bm$ and $a + b\alpha$) and then make a matrix of all the ϵ vectors. It is then straight forward to row reduce the matrix to find a zero row, that is a linear dependent set of the vectors. We then get two equalities

$$\prod_{(a,b)\in S} (a_i + b_i m) = x^2 \in \mathbb{Z}$$

$$\prod_{(a,b)\in S} (a_i + b_i \alpha) = y^2 \in \mathbb{Z}[\alpha].$$

When the set S is found we need to find the elements x and y, which means we need to find the square root of the two elements x^2 and y^2. When that is done we use the homomorphism ϕ. Since both $a + bm$ and $a + b\alpha$ have the same image under ϕ in $\mathbb{Z}/n\mathbb{Z}$, we achieve the wanted equality like (1) as

$$\phi(x)^2 \equiv \prod_{(a,b)\in S} \phi(a_i + b_i m) \equiv \prod_{(a,b)\in S} \phi(a_i + b_i \alpha) \equiv \phi(y)^2 \pmod{n}$$

Then find $\gcd(\phi(x) \pm \phi(y), n)$, hopefully a non-trivial factor of n.

In the next subsections we will go into the details concerning the different parts of the algorithm. We will discuss how the factor base is constructed, how the sieving is performed, and how to find the square roots of an element in $\mathbb{Z}[\alpha]$. The descriptions will be illustrated with the corresponding parts for factoring the 6 digit integer $n = 510^2 + 1 = 260101$.

In this description partial relations has not been taken into consideration. There is also assumed that the number field is a principal ideal domain. No concerns is made about the fact that the number field might not even be a unique factorization domain. These are problems that occur but for the simplifications it does they have not been taken into consideration when describing the algorithm. Comments will be made in the parts below where these assumptions are needed.

4.2 The special form of n

Why does the number field sieve require the integer to factor to be of the special form $n = r^e - s$? The answer is quit simply the procedure to find the polynomial and consequently the size of the number field. As described in section 4.1 there is a simple way to construct an irreducible polynomial given $n = r^e - s$, and subsequently the number field, the number field is small and then easier to control.

Example

For the number $n = 510^2 + 1 = 260101$ we chose the polynomial to have degree 2, that is $d = 2$. n has $r = 510$, $e = 2$, $s = -1$ which gives the values $k = 1$, $t = -1$, $f(x) = x^2 + 1$, and $m = 510$. A root α of f(x) is then the complex number $\sqrt{-1} = i$, the number field K is $\mathbb{Q}[i]$, where the subring $\mathbb{Z}[i]$ is the ring of integers of $\mathbb{Q}[i]$. Notice $\mathbb{Z}[i]$ is in fact a principal ideal domain. The norm of an element $a + bi \in \mathbb{Z}[i]$ is the positive integer $a^2 + b^2$. We set up the homomorphism ϕ

$$\phi : \mathbb{Z}[i] \rightarrow \mathbb{Z}/260101\mathbb{Z}$$

$$i \rightarrow 510$$

4.3 Factor base

The factor base is to consist of three parts, the first part is all the prime numbers up to some limit. The second and third part of the factor base is to consist of generators for all the first degree prime ideals that have norm less than some chosen bound, and a set of generators for the group of units

of $\mathbb{Z}[\alpha]$. Since the subring $\mathbb{Z}[\alpha]$ is assumed to be a principal ideal domain we know there exists a generator for all the ideals in $\mathbb{Z}[\alpha]$.

The first part of the factor base is practically the same as the factor base the quadratic sieve uses, set a bound B_1 and we want to find B_1-smooth integers $(a + bm)$.

$$a + bm = \prod_{p_j \leq B_1} p_j^{\epsilon(p_j)}$$

where $\epsilon(p_j) \in \mathbb{Z}_{\geq 0}$ is the corresponding exponent of each $p_j \leq B_1$

The second and third part works just as the first part in $\mathbb{Z}[\alpha]$, it factorizes elements in the extension. It does this by means of the first degree prime ideals that occur in the ideal generated by the element. Call the set of generators for the first degree prime ideals of norm $\leq B_2$, G, and the set of generators for the units U. Then we will look for elements that factors by G and U, that is elements on the form:

$$a + b\alpha = \prod_{u_i \in U} u_i^{\epsilon(u_i)} \prod_{g_i \in G} g_i^{\epsilon(g_i)}.$$

Finding generators

To find the set G start by making a list of all first degree prime ideals of norm less than the preset bound B_2. As proved in section 3 this amounts to making a list of pairs $(p, c \bmod p)$. p (prime) is the norm of the ideal and $c \in \mathbb{Z}$ such that $f(c) \equiv 0 \bmod p$. To find the pairs, as described in [1], a probabilistic root finder for polynomials over finite fields can be used [5, 4.6.2]

In practice the search for both U and G are performed at the same time. The search can be carried out like this. Fix a multiplier bound M and a search bound C, see [1, 3.6] for references on how to best determine the bounds. For all the ideals $\mathbf{I}$ we want to find generators, fix the number $m(\mathbf{I}) = M+1$. This number is holding the status of $\mathbf{I}$, if no generator is found then $m(\mathbf{I}) > M$, if a generator $\bar{\pi}_p$ with $\mathbf{N}(\bar{\pi}_p) = \pm m(\mathbf{I})\mathbf{I}$ has been found then $m(\mathbf{I}) < M$. And the first degree prime ideal generated by $\bar{\pi}_p$ is $\mathbf{I}$ times an ideal of norm $m(\mathbf{I})$

The search goes like this. For all elements $\lambda = \sum_{i=0}^{d-1} a_i \alpha^i \in \mathbb{Z}[\alpha]$ that satisfy $\sum_{i=0}^{d-1} a_i^2 |\zeta|^{2i} < C$ compute the norm $\mathbf{N}(\lambda)$ to check if the norm is of the form kp for some p from the list of pairs $(p, c \bmod p)$, k a non-zero integer with absolute value $\leq M$. The number $|\zeta|$ denotes the real number $|t|^{\frac{1}{d}}$

For all the elements λ where $\mathbf{N}(\lambda) = kp$ identify the first degree prime ideal, the pair $(p, c \bmod p)$ that λ correspond to by checking if $\sum_{i=0}^{d-1} a_i c^i \equiv$

$0 \bmod p$. Then if $|k| < m(\mathbf{I})$ update the data for $\mathbf{I}$ by setting $\bar{\pi}_p = \lambda$ and $m(\mathbf{I}) = |k|$.

If M and C has been chosen properly one should have generators for all the ideals after searching through all λ's. For all first degree prime ideals where $m(\mathbf{I}) = 1$ set $\pi_p = \bar{\pi}_p$, where $m(\mathbf{I}) > 1$ find the generator π_p by dividing $\bar{\pi}_p$ by a generator of the appropriate ideal of norm $m(\mathbf{I})$. This requires the computation of the generators and their inverses of the ideals of norm$< M$. But there are not many of these ideals and one may hope to encounter them during the search described above[1, 3.1]

While searching for the generators π_p, all units that are encountered can be stored as well. These are of course the elements λ where $\mathbf{N}(\lambda) = \pm 1$ but also quotients of elements with the same absolute value of their norm, and generate the same ideal. The set of units we are left with will then very often be the set U of generators for the group of units we are looking for[1, 3.1]

Remark

In this part of the algorithm the assumption of $\mathbb{Z}[\alpha]$ being a principal ideal domain guarantees the existence of the generators for the ideals. However this is not so easy in a general case where the number field might not be a principal ideal domain.

Example

In our example we chose the limit $B_1 = 40$ this resulted in the factor base

$$B' = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37\}$$

For the ideals we chose the first degree prime ideals with norm less than $B_2 = 55$. We performed the above described search with $M = 5, C = 55$, and found generators for all the first degree prime ideals, that is the value m($\mathbf{I}$) was equal to 1 for all the ideals when the search was done. The generators are presented with the corresponding pair $(p, c \bmod p)$ for the first degree prime ideal that they generate.

$(p, c \bmod p)$	Generator	$(p, c \bmod p)$	Generator
$(2, 1 \bmod 2)$	$(1+i)$	$(29, 12 \bmod 29)$	$(5+2i)$
$(5, 2 \bmod 5)$	$(1+2i)$	$(37, 6 \bmod 37)$	$(1+6i)$
$(13, 5 \bmod 13)$	$(3+2i)$	$(41, 9 \bmod 41)$	$(5+4i)$
$(17, 4 \bmod 17)$	$(1+4i)$	$(53, 23 \bmod 53)$	$(7+2i)$

The units of $\mathbb{Z}[i]$ are the set $\{i, -i, 1, -1\}$, a generator for this set is i

4.4 Sieving

When sieving for relations, we want to find integers of the form $a + bm$ and $a + b\alpha$, $a, b \in \mathbb{Z}$, α and m roots of f in $\mathbb{Z}[\alpha]$ and $\mathbb{Z}/n\mathbb{Z}$. The pairs (a, b) we are looking for need to satisfy three conditions to be smooth.

1. $\gcd(a, b) = 1$
2. $|a + bm|$ is B_1-smooth
3. $a + b\alpha$ is B_2-smooth

This is for suitable bounds B_1 and B_2. The second condition ensures that $(a + bm)$ is B_1-smooth in the same way as the factor base in quadratic sieve does.

The first and third conditions will give us elements that has only generators of first degree prime ideals with norm less than B_2 as factors(and perhaps a unit). In the same way as we search for smooth integers, we search for smooth elements, that is elements that is completely factored by U and G.

Condition one ensures that all ideals containing $a + b\alpha$ are first degree prime ideals, as proved by Lemma(3)

We do not have to use the generators of each first degree prime ideal to check if the elements is B_2-smooth. The idea is as follows, due to (10) we know exactly when $a + b\alpha$ is contained in the first degree prime ideal corresponding to $(p, c \pmod p)$. Since the extension is assumed to be a principal ideal domain and since the norm is multiplicative this implies that the prime ideal factorization of $\langle a + b\alpha \rangle$ corresponds to the factorization of the norm of $\langle a + b\alpha \rangle$. That is each first degree prime ideal that is a 'factor' of $\langle a + b\alpha \rangle$ corresponds to a factor of the norm. So one can simply check the factorization of the norm $\mathbf{N}(a + b\alpha) = a^d - t(-b)^d$ [1] to see if the element $a + b\alpha$ factors in the factor base. This means that the second and third part of our factor base(U and G) will during the sieving be replaced by all primes less than a limit B_2 that together with an integer c have a corresponding first degree prime ideal in $\mathbb{Z}$. This means in the context of condition 3 that an element $a + b\alpha$ is B_2-smooth if the norm $\mathbf{N}(a + b\alpha)$ only has prime factors less than B_2

Remark

In the sieving step partial relations play out their role. One could allow the smooth elements to have one factor above the limits B_1 and B_2, but below some other limits B_3 and B_4. These relations would be called partial relations and would increase the amount of relations greatly, but also increase

the size of the matrix in the next step of linear algebra. One partial relation would also require at least one more partial relation to be of any use.

One factor base

Instead of two factor bases, there is a practical way to let B' and B'' be equal(remember to check that the p's have corresponding first degree prime ideals). In this way it is enough to check if $|(a+bm)\mathbf{N}(a+b\alpha)|$ is factored by the factor base. This does not change the algorithm, as one can freely choose the primes to put in the factor base.

Obstruction

The technique described above using the norm of $a+b\alpha$ to check for smoothness is not bulletproof. The problem is that for each p there can be several corresponding c's. Still there is one first degree prime ideal for each pair $(p, c \pmod p)$ but the norm does not distinguish between the different ideals as they have the same norm p. As an example of this consider the polynomial $f = x^2 + 1 (\alpha = i)$, now the prime $p = 13$ has two corresponding c's, that is the pairs $(13, 5 \pmod{13})$, $(13, 8 \pmod{13})$.

Consider the relations (a,b): (2,3), (3,2)

$$\mathbf{N}(2+3i) = 13$$

$$\mathbf{N}(3+2i) = 13$$

Multiplying the norm of these together gives a square $13 \cdot 13 = 169$ but the element $(2+3i)(3+2i)$ we get by multiplying the generators for the first degree prime ideals is not a square in $\mathbb{Z}[\alpha]$.

The reason for this is of course that the relation $(2, 3)$ with the factor 13 correspond to the pair $(13, 8 \pmod{13})$, but the relation $(3, 2)$ with the factor 13 correspond to the pair $(13, 5 \pmod{13})$(Here both elements are generators for two different first degree prime ideals of norm 13). This means no square of an element when they are multiplied together, and we can't use these relations alone together.

The easiest way to get around this is of course to not use the primes that have several corresponding c's in the factor base.

Example

In the example all but the prime 2 has two different integers c such that $f(c) \equiv 0 \pmod p$. We then chose the smallest c value for each of them and

tested each smooth integer if it corresponded to the c we wanted. This would be very time consuming for a larger n, but was satisfying for the purpose of our example and made sure all the relations that contained the same factor corresponded to the same first degree prime ideal.

Sieving Technique

Here we describe a technique to perform the sieving in practice. This technique uses two separate sieves, and combines their results to use the relations which both sieves find it likely to be smooth.

First decide what range of a's to search from, $[a_{min}, a_{max}]$ and a start value for b, in practice there is no real need for a maximum b value as one can just continue until the desired number of relations is achieved.

Start the first sieve by fixing the value of b, and search for a values that are $-bm \pmod{p}$, for all the p's in you factor base B'. This gives you a's that have a reasonable chance of being B_1-smooth.

In practice one will have an array with $a_{max} - a_{min} + 1$ columns for each b, and every time a number a is $-bm \pmod{p}$ one adds $\log p$ to the a's place in the array. Then after sieving through all p's the a's that have a value on its place in the array that are close to $\log(a + bm)$ will be the most likely candidates to be B_1-smooth.

In the second sieve fix the value for b again and start looking for a values that are $-bc \pmod{p}$, this to find the pairs (a, b) in which first degree prime ideals occur in the ideal generated by $a + b\alpha$. And $a + b\alpha$ is contained in an ideal if and only if $a \equiv -bc \pmod{p}$. In the same way as the first sieve one would arrange an array and add $\log p$ to a's location every time it is $-bc \pmod{p}$, and when you are finished the locations that have a value close to $\log \mathbf{N}(a + b\alpha)$ are the most likely candidates to be B_2-smooth

After the two sieving parts are done combine the candidates that both sieves find it likely to be smooth, check them for gcd and do trial division to find the integers that are completely factored by the factor bases. If after the first sieve, the number of relations that most likely are B_1-smooth aren't that high it can be preferable to just use trial division to check if they also are B_2-smooth, but in a realistic scenario that number will be considerable so it will be more efficient to apply the second sieve right away. The two sieves can then be executed simultaneously.

Remark

In this sieving technique prime powers can be overlooked, but as they are very rare this is a minor problem and will not be of any great consequence in finding enough smooth relations. The process to find the few relations that have prime powers would be very time consuming and it is actually just a step to speed up the algorithm to overlook prime powers.

Example

We sieved with the values as follows, $-a_{min} = a_{max} = 200$, and $b = 1$ up to $b = 54$, when we had a total of 23 smooth pairs (a, b), which is just enough to be sure a of square as the number of primes in the factor base B' is 12, G has 8 elements and U has 2.

4.5 Linear algebra

Once enough relations have been found it is straight forward to put the exponent vectors into a matrix(modulo 2) and start row reducing to find a zero row. As mentioned in the run time analysis of quadratic sieve there are different algorithms that are more effective than Gaussian elimination, for the current record factorization by the special number field sieve($2^{1039} + 1$), the block Wiedemann algorithm[10] was used to find the right combination of the smooth relations[3]

Example

We used ordinary Gaussian elimination to find a combination of the smooth relations that were a square. The smooth pairs (a, b) we ended up with were:
$S = \{(34, 19), (-70, 1), (-4, 1), (-2, 5), (3, 1), (-5, 7), (3, 2), (-59, 2), (-102, 23)\}$
This gives the two products in $\mathbb{Z}$ and $\mathbb{Z}[\alpha]$ for $(a_i, b_i) \in S$

$$\prod_{(i=1)}^{9} (a_i + b_i m) = 1153263403915814430522226944 00$$

$$\prod_{i=1}^{9} (a_i + b_i \alpha) = (12028960768 - 34623604674i)$$

4.6 Square roots

When all the problems with finding the factor base and, the right combination of relations during the sieving and linear algebra step has been overcome one is left with a set S with the relations that combined are squares both in $\mathbb{Z}$ and $\mathbb{Z}[\alpha]$.

$$x^2 = \prod_{(a,b)\in S} (a_i + b_i m) \in \mathbb{Z}$$

$$y^2 = \prod_{(a,b)\in S} (a_i + b_i \alpha) \in \mathbb{Z}[\alpha]$$

The square root of x is straight forward to compute. We know all exponents of each relation in the product and it is therefore not even necessary to calculate x^2. We just use the prime factorization and compute x straight away.

As for the product of the elements in the number field it is much the same thing. We represent each relation $a + b\alpha \in S$ by it factors in G and U.This leaves us with a factorization where each first degree prime ideal is represented an even number of times with their generator. Each unit will also occur an even number of times, and the square root is found just as the integer square root, from the exponent vectors. The square root can be represented as

$$y = \prod_{u_i \in U} {u_i}^{\frac{1}{2}e(u_i)} \prod_{\pi_p \in G} \pi_p^{\frac{1}{2}e(\pi_p)}.$$

Then applying ϕ gives two integers, that squared are congruent modulo n.

$$\phi(x)^2 \equiv \prod_{(a,b)\in S} \phi(a_i + b_i m) \equiv \prod_{(a,b)\in S} \phi(a_i + b_i \alpha) \equiv \phi(y)^2 \pmod{n}$$

Remark

Here we again see the importance of the extension being an principal ideal domain, being able to find the generators for each ideal makes the process of finding the square roots in $\mathbb{Z}[\alpha]$ a simple task.

Example

As we have now seen it was not necessary to compute the squares as we could just use the already known factorizations. We found the squares of the product of the elements in S to be:

$$\prod_{(i=1)}^{9}(a_i + b_i m) = 1153263403915814430522226944 00 = 339597320942880^2$$

$$\prod_{i=1}^{9}(a_i + b_i i) = (12028960768 - 34623604674i) = (-156017 + 110961i)^2$$

4.7 Example summary

We summarize the factoring of the integer $n = 260101 = 510^2 + 1$ before we present the factors.

The polynomial is $f(x) = x^2 + 1$, the root $\alpha = i$, $m = 510$, the number field is $K = \mathbb{Q}[i]$, and we have used the subring $\mathbb{Z}[i]$ of K. The norm of an element in $\mathbb{Z}[i]$ is $\mathbf{N}(a+bi) = a^2 + b^2$ The factor bases was created using the limits $B_1 = 40$ and $B_2 = 55$. The homomorphism $\phi : \mathbb{Z}[i] \to \mathbb{Z}/n\mathbb{Z}$ sends i to 510

The sieving was performed for $-a_{min} = a_{max} = 200$, for $b = 1$ up to $b = 54$ when we reached a total of 23 smooth pairs (a,b). That is just enough to be guaranteed a linear dependent set as there are 22 elements in the factor base. After doing Gaussian elimination on the exponent vectors, the following set S of pairs where found to be a square when multiplied together

(a,b)	a+bm	factors	$\mathbf{N}(a+bi)$	factors	ideal factorization
(34,19)	9724	$2^2 \cdot 11 \cdot 13 \cdot 17$	1517	$37 \cdot 41$	$(-1)(i)(1+6i)(5+4i)$
(-70,1)	440	$2^3 \cdot 5 \cdot 11$	4901	$13^2 \cdot 29$	$(i)(3+2i)^2(5+2i)$
(-4,1)	506	$2 \cdot 11 \cdot 23$	17	17	$(i)(1+4i)$
(-2,5)	2548	$2^2 \cdot 7^2 \cdot 13$	29	29	$(i)(5+2i)$
(3,1)	513	$3^3 \cdot 19$	10	$2 \cdot 5$	$(-1)(i)(1+i)(1+2i)$
(-5,7)	3565	$5 \cdot 23 \cdot 31$	74	$2 \cdot 37$	$(1+i)(1+6i)$
(3,2)	1023	$3 \cdot 11 \cdot 31$	13	13	$(3+2i)$
(-59,2)	961	31^2	3485	$5 \cdot 17 \cdot 41$	$(1+2i)(1+4i)(5+4i)$
(-102,23)	11628	$2^2 \cdot 3^2 \cdot 17 \cdot 19$	10933	$13 \cdot 29^2$	$(i)(3+2i)(5+2i)^2$

This gives us the following squares in $\mathbb{Z}$ and $\mathbb{Z}[i]$

$$\prod_{(a,b)\in S}(a_i + b_i m) = (2^5 \cdot 3^3 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 31^2)^2 = 339597320942880^2$$

$$\prod_{(a,b)\in S}(a_i + b_i \alpha) = (i^3(-1)(1+i)(1+2i)(3+2i)^2(1+4i)(5+2i)^2(1+6i)(5+4i))^2$$

$$= (-156017 + 110961i)^2$$

Applying ϕ upon both squares will give us a relation like (1) in $\mathbb{Z}/n\mathbb{Z}$

$$\phi(339597320942880)^2 = (339597320942880 \pmod{260101})^2 = 151328^2$$

$$\phi(156017-110961i)^2 = ((-156017+110961\cdot 510) \pmod{260101})^2 = (-7824)^2$$

$$151328^2 \equiv (-7824)^2 \bmod 260101$$

This gives us the results we wanted and by using the Euclidean algorithm to find the greatest common divisor we achieve two factors of 260101

$$\gcd(260101, 151328 - 7824) = 8969$$
$$\gcd(260101, 151328 + 7824) = 29$$

$$8969 \cdot 29 = 260101$$

4.8 Running time

In the same way as for the quadratic sieve there has not yet come a rigorous proof of the actual running time of the special number field sieve. All estimates are based on careful heuristic analysis with plausible assumptions. The conjectured running time of the number field sieve to factor $n = r^e - s$ is usually denoted in the notation $L_n[\nu, \lambda]$ where:

$$L_n[\nu, \lambda] = e^{(\lambda(\log n)^\nu (\log\log n)^{1-\nu})}$$

The expression $L_n[\nu, \lambda + O(1)]$ is also abbreviated to $L_n[\nu, \lambda]$, here O(1) is for $n \to \infty$

Analysis and arguments for the running time results presented below can be found in[1, 6] and [6, 10-11]

The probability for the smoothness of integers is described by the following result: Let $C \subset \mathbb{R}^4$ be a compact set such as that for all $(\lambda, \mu, \omega, \nu) \in C$ one has $\lambda > 0,\ \mu > 0,\ 0 < \omega < \nu < 1$. Then the probability that a random positive integer $\leq L_n[\nu, \lambda]$ is $L_n[\omega, \mu]$-smooth equals $L_n[\nu-\omega, \frac{-\lambda(\nu-\omega)}{\mu}+O(1)]$ for $n \to \infty$, uniformly for $(\lambda, \mu, \omega, \nu)$ in C

One can further derive from the above argument optimal choices for the parameters(we assume $a_{max} = -a_{min}$), $a_{max}, b_{max}, B_1, B_2$ and d as functions of n. The optimal choice for a_{max}, b_{max}, B_1 and B_2 is obtained if they are all taken to be equal to

$$e^{\left((\frac{1}{2}+O(1))\left(d\cdot\log d+\sqrt{(d\log d)^2+2\log(n^{\frac{1}{d}})\log\log(n^{\frac{1}{d}})}\right)\right)} \tag{11}$$

here the O(1) is for $e \to \infty$, the r and s are bounded by an upper limit, and $1 < d^{2d^2} < n$. The size of the factor base and the number of relations one expects to find is given by the same argument.

This then gives an estimate of the size of the numbers $|(a+bm)\mathbf{N}(a+b\alpha)|$ which one wants to be smooth of

$$e^{\left((\frac{1}{2}+O(1))\left(d^2\log d+2\log(n^{\frac{1}{d}})+d\sqrt{(d\log d)^2+2\log(n^{\frac{1}{d}})\log\log(n^{\frac{1}{d}})}\right)\right)}$$

The expected running time for the sieving and the linear algebra part is then

$$e^{\left((1+O(1))\left(d\log d+\sqrt{(d\log d)^2+2\log(n^{\frac{1}{d}})\log\log(n^{\frac{1}{d}})}\right)\right)}$$

The rest of the algorithm takes less time and is therefore not considered. To find the optimal choice of d we want to minimize the above expression. This we can see will be when the two factors $(d\log d)^2$ and $\log(n^{\frac{1}{d}})\log\log(n^{\frac{1}{d}})$ are of the same order of magnitude, the optimal choice for d will then be approximately.

$$d = \left(\frac{(3+O(1))\log n}{2\log\log n}\right)^{\frac{1}{3}}, for\ e \to \infty$$

Using this value for d in the expressions for a_{max}, b_{max}, B_1 and B_1 as suggested in (11) will give a value of $L_n[\frac{1}{3}, (\frac{2}{3})^{\frac{2}{3}}]$ for the sieving bounds. If we assume the typical size of $(a+bm)$ and $|\mathbf{N}(a+b\alpha)|$ to be $L_n[\frac{2}{3}, (\frac{2}{3})^{\frac{1}{3}}]$, then the size of the numbers $|(a+bm)\mathbf{N}(a+b\alpha)|$ that we want to be smooth will be $L_n[\frac{2}{3}, (\frac{16}{3})^{\frac{1}{3}}]$

In terms of this notation, and with the above values the expected run time for the special number field sieve with $r, |s|$ below a fixed upper bound is

$$L_n\left[\frac{1}{3}, c\right]$$

$$c = \left(\frac{32}{9}\right)^{\frac{1}{3}} = 1,5263$$

this is irrespectively of the size of the factors of n

The reason for not having a rigorously proved result for the running time is the same as for the quadratic sieve, no one has managed to prove accurately how fast one will be able to find the relations needed, and a consequence of this is that both algorithms run faster in practice, as the estimates are for worst case $n \to \infty$

Comparing the running times for the special number field sieve and the quadratic sieve we see that the special number field sieve has a significantly lower estimate than the quadratic sieve. The running time for quadratic sieve is

$$e^{C\sqrt{\log n \log\log n}} = L_n[\frac{1}{2}, C]$$

The most significant parameter in the estimates of the form $L_n[\nu, \lambda]$ is ν [1]. The special number field sieve has a much lower value than the quadratic sieve, $\frac{1}{3}$ against $\frac{1}{2}$, this is a cube root against a square root.

5 Discussion

The difference in the various index calculus algorithms for integer factorization is the choice of factor base and the sieving step. The linear algebra is the same for all of them. The difference comes down to how fast one can find the smooth integers, and of course if the smooth integers can be found at all? This is illustrated by the quadratic sieve as it finds enough relations for integers up to about 100 digits, but when the integer gets larger it just takes to much time to find the smooth integers as there are fewer of them.

The interesting thing about these differences is that each algorithm has its own intervals where it is fastest, but since the 'fastest/best' algorithms most often refers to which can factor the largest integers in a reasonable amount of time the special number field sieve is considered the best at the current moment.

The special number field sieve is the asymptotically fastest algorithm when the number n to factor is assumed as $n \to \infty$. But there are many technicalities and aspects of the algorithm that do not make it practical for the factoring of a random integer. The most obvious is of course that the special number field sieve only applies to integers of the form $n = r^e - s$. There has been a generalization, the general number field sieve and it will be commented on below. In this section some of the aspects that makes the special number field sieve not so practical will be addressed..

The special number field sieve cannot take advantage of already known small factors in an integer n. This seems very odd, but is due to the algorithms use of the integers form $n = r^e - s$. Since the polynomial and then the number field is constructed from the integers r,e, and s, it cannot use the quotient, as this integer probably will not be of the special form. And then there would not be such a nice number field to work with, and the algorithm would lose its best advantages.

The size of the factor base and the interval of a's to sieve in has to be set before one starts the algorithm. This makes the job harder as one must mostly rely on experience when deciding the sizes of the factor bases and the interval to sieve over. This also applies to the degree of the polynomial that must be chosen without any real help from the integer to factor. These choices can result in doing a lot more work than is necessary, but the wrong choice can also result in failure for the algorithm. These choices would not be difficult if there were better estimates for the running time.

The special number field does not factor 'small' integers fast. As mentioned the quadratic sieve is preferred for integers of up to about 100 digits. This is due to all the work that has to be done in the sieving and linear algebra step of the special number field sieve. This work still has to be done for small n, it does not decrease that much as n gets smaller. Then again for really small integers neither quadratic sieve nor the special number field sieve are effective as the steps they go through are not really necessary. Even though they are asymptotically fast, it does not make them faster all the time. The special number field sieve is an algorithm only practical for huge integers, and for those huge integers there is no better algorithm.

5.1 General number field sieve

The general number field sieve is in practice the same algorithm as the special number field sieve, but when using a general n, that is n of no special form, there are several problems that arise through the steps of the algorithm. It starts already in the first step when the polynomial is to be constructed, this was real easy in the case of $n = r^e - s$, but for a general n it is worse. However the method described in [6, 3] gives a suitable polynomial with a little work. It is based on n and the choice of d. One sets $m = n^{\frac{1}{d}}$, writes the number n in base m, and the coefficients from this number will be the coefficients in the polynomial f. That is

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

$$f(x) = c_d X^d + c_{d-1} X^{d-1} + \cdots + c_1 X + c_0.$$

But this gets further consequences for the number field, which will be bigger, harder to control, and it will be unrealistic to find the generators of the first degree prime ideals. This gives a huge problem later as one is stuck with a square of an element that is the product of many small ones in a number field without an easy option to find the square root. This was a task that at first seemed like a giant bottleneck, to big of task for the algorithm to be of any practical use at all. But due to Couveigne there is a method to find the square roots in a reasonable amount of time. This method is based on

a careful use of the Chinese remainder theorem, but works only when the degree of the number field is odd [4].

These obstructions, especially the problem of finding square roots in a number field causes the general number field sieve to be significantly slower than the special number field sieve. The general number field sieve has according to [1, 1] a worst case running time

$$L_n[\frac{1}{3}, c]$$

$$c = \frac{(92 + 26\sqrt{13})^{1/3}}{3} = 1,9019$$

The difference in the running times of the special and general number field sieve is illustrated in the big gap in the size of the two algorithms record factorizations. The special number field sieve has factored an integer that has a size which is 376 bits more than what the general number field sieve has.

6 Concluding remarks

The special number field sieve(and quadratic sieve) is in the group of index calculus algorithms for integer factorization. The index calculus algorithms follows the same recipe, that is find a suitable factor base, search for smooth integers of some form, do linear algebra to find the right set of the smooth integers, then use Euclid's algorithm to find a factor.

The special number field sieve uses a irreducible polynomial, a number field and an homomorphism on its way to factor an integer n. It searches for smooth integers of the form $a + bm$ and $a + b\alpha$ in $\mathbb{Z}$ and $\mathbb{Z}[\alpha]$, uses linear algebra to find a set of relations that together is a square. The square roots are found from the factor base, and via the homomorphism a relation in $\mathbb{Z}/n\mathbb{Z}$ like (1) is achieved. A factor can then hopefully be found by applying the Euclidean algorithm on n and the difference of the square roots.

The special number field sieve is faster than the quadratic sieve and is the asymptotically fastest algorithm for integer factorization known today. But the special number field sieve works only for integers of the special form $n = r^e - s$. The quadratic sieve works for 'all' integers. The running time estimates however are not deterministic for neither of the algorithms. This is mostly due to the difficulty of computing the probability of finding smooth integers in a given interval accurately. It means the algorithms run faster in practice than what is approximated for the worst case running time. Currently the largest integer factored by the special number field sieve is $2^{1039}-1$, a 1039 bit composite integer, that is 313 digits.

References

[1] M. S. Manasse J. M. Pollard A. K. Lenstra, H.W. Lenstra. The number field sieve. *Lecture notes in mathematic*, (1554):11–42, 1993.

[2] R. Crandall C. Pomerance. *Prime numbers, a computational perspective.* Springer Verlag, 2001.

[3] Scott Contini. Factorization announcements. May 2007. http://www.crypto-world.com/FactorAnnouncements.html.

[4] J. M. Couveigne. Computing a square root for the number field sieve. *Lecture notes in mathematic*, (1554):95–102, 1993.

[5] D.E.Knuth. *The art of computer programming, Semi numerical algorithms*, volume 2. Addison-Wesley, second edition.

[6] C. Pomerance J. P. Buhler, H.W. Lenstra. Factoring integers with the number field sieve. *Lecture notes in mathematic*, (1554):50–94, 1993.

[7] N. Koblitz. A course in number theory and cryptography. *Graduate texts in mathematics*, (114), 1987.

[8] Manasse Pollard Lenstra, Lenstra. The factorization of the ninth fermat number. *Mathematics of Computation*, (61):319–349, 1993.

[9] Victor Shoup. *A computational introduction to Number Theory and Algebra.* Cambridge university press, 2005.

[10] D. H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory,issue 1*, (32):54–62, 1986.