

# NFS Obstructions

*Jonas Lindstrøm Jensen (jonas@imf.au.dk)*

*IMF, 2007*

## 1 Introduction

In the presentation of the NFS algorithm in [1], we skipped two important points.

- How do we conclude from the  $S$ -subset of rows in the matrix adding up to zero that  $\beta = \prod_{(a,b) \in S} (a - b\alpha)$  is a square in  $\mathbb{Z}[\alpha]$  and  $b = \prod_{(a,b) \in S} (a - b\alpha)$  is a square in  $\mathbb{Z}$ ?
- How do we find the square root of  $b$  in  $\mathbb{Z}$  and  $f'(\alpha)^2 \prod_{(a,b) \in S} (a - b\alpha)$  in  $\mathbb{Z}[\alpha]$ ?

Both require some abstract algebra, not all of which we will do in detail.

## 2 Is $\beta$ and $b$ squares?

The condition that the sum of the rows from  $S$  is the zero vector gives us the following information

- $b \geq 0$ .
- $|b|$  is a square in  $\mathbb{Z}$ .
- $\sum_{(a,b) \in S} \mathbb{v}(a - b\alpha)$  is even.
- $\prod_{(a,b) \in S} a - bs_j$  is a square modulo  $q_j$  for  $j = 1, 2, \dots, k$ .

And why is that? Every row in the matrix correspond to a pair  $(a, b)$ . The first bit of every row is set if  $G(a, b) < 0$ . We have  $b = \prod_{(a,b) \in S} G(a, b)$ , so if the first bit of the sum is 0, then  $b > 0$ . The following  $\pi(B)$  bits is the exponents of the prime factorization of  $|G(a, b)|$ . Each bit is set if the exponent is odd. So if the sum of these are zero we have that all exponents of  $|b|$  is even. These two conditions give us, that  $b$  is a square in  $\mathbb{Z}$ .

It is clear that if the next  $B'$  bits of the sum all are zero then  $\sum_{(a,b) \in S} \mathbb{v}(a - b\alpha)$  is even. According to Lemma 6.2.1 in [1] this is a necessary condition of  $\beta$  being a square in  $\mathbb{Z}[\alpha]$ . Finally, if the last  $k$  bits of the sum is zero, then the product

$$\prod_{(a,b) \in S} \left( \frac{a - bs_j}{q_j} \right)$$

contains an even number of  $-1$ 's for all  $j$ , and so the product is 1. Since the Lagrange symbol is multiplicative, this implies that  $a - bs_j$  is a square modulo  $q_j$  for  $j = 1, 2, \dots, k$ . The question is now: How do we derive that  $\beta$  is a square in  $\mathbb{Z}[\alpha]$  from this information?

Let us state the following theorem that assures us that it is enough to prove that  $\beta$  is a square in  $I$ , where  $I$  denote the elements in  $\mathbb{Q}[\alpha]$  that is the root of an integer polynomial.

**Lemma 2.1.** *Let  $f(x)$  be a monic irreducible polynomial in  $\mathbb{Z}[x]$ , with root  $\alpha \in \mathbb{C}$ . Let  $I$  be as above and let  $\beta \in I$ . Then  $f'(\alpha)\beta \in \mathbb{Z}[\alpha]$ .*

$I$  is a ring, so since  $\alpha \in I$  (it is root of  $f$ ) and  $\beta$  is expressed by  $\alpha$ , we have  $\beta \in I$ . Let us define the Quadratic Character. The notation is taken from [2].

**Definition 2.2.** Let  $q$  be an odd prime and  $s$  an integer such that  $f(s) \equiv 0 \pmod{q}$  and  $f'(s) \equiv 0 \pmod{q}$ . Let an element of  $\mathbb{Z}[\alpha]$  be written as  $c_0 + c_1\alpha + \dots + c_{d-1}\alpha^{d-1}$ , and assume  $q$  does not divide  $c_0 + c_1m + \dots + c_{d-1}m^{d-1}$ . Then we define  $\chi_{(q,s)} : \mathbb{Z}[\alpha] \rightarrow \{\pm 1\}$  by

$$\chi_{q,s}(c_0 + c_1\alpha + \dots + c_{d-1}\alpha^{d-1}) = \left( \frac{c_0 + c_1m + \dots + c_{d-1}m^{d-1}}{q} \right).$$

With that notation Lemma 6.2.4 becomes

**Lemma 2.3.** *Let  $q$  be an odd prime and  $s$  an integer such that  $f(s) \equiv 0 \pmod{q}$  and  $f'(s) \equiv 0 \pmod{q}$ . Let  $S$  be a set of coprime pairs of integers  $(a, b)$  such that  $q$  does not divide  $a - bs$  for any  $(a, b) \in S$ . Assume  $\beta = \prod_{(a,b) \in S} (a - b\alpha)$  is a square in  $\mathbb{Q}[\alpha]$  then*

$$\chi_{q,s}(\beta) = 1.$$

All the conditions, except that  $\beta$  is a square, is met in our case. But the lemma gives us a *necessary* condition for squareness.

The idea now is to check for how many  $q, s$  we shall test  $\chi_{q,s}$  to be (almost) certain that  $\beta$  is a square. Let us define

$$V = \{z \in \mathbb{Q}[\alpha]^* \mid v_P(zI) \text{ even for all prime } P \subset \mathbb{Z}[\alpha]\},$$

where  $v_P(J)$  denotes the exponent of  $P$  in the prime ideal factorization of the ideal  $J$ . We have  $\beta \in V$ . This comes from the third piece of information we found earlier. In the proof of Lemma 6.2.1 in [1] they use the relation between the  $\underline{v}$ 's and the  $v_P$ 's to say something quite similar.

The problem is now if  $\beta$  also is a square. We therefore study the quotient group

$$V/\mathbb{Q}[\alpha]^{*2}$$

with the natural multiplication. We want to prove that the coset of  $\beta$  in the quotient is zero. That will imply  $\beta \in \mathbb{Q}[\alpha]^{*2}$  (eg.  $\beta$  is a square in  $\mathbb{Q}[\alpha]$ ) and by Lemma 2.1  $f'\alpha^2\beta$  is a square in  $\mathbb{Z}[\alpha]$ . The following lemma is from [2].

**Lemma 2.4.** *Each  $\chi_{q,s}$  with  $q$  odd and  $f'(s) \not\equiv 0 \pmod{q}$  induces a nontrivial group homomorphism from  $V/\mathbb{Q}[\alpha]^{*2}$  to  $\{\pm 1\}$ .*

We will just denote the deduced homomorphism by  $\chi_{q,s}$ . If we could find a set  $T$  of pairs  $q, s$  such that  $\{\chi'_{q,s} \mid (q, s) \in T\}$  spans  $\text{Hom}(V/\mathbb{Q}[\alpha]^{*2}, \{\pm 1\})$  we are done. This is because we then test if  $\chi_{q,s}(\beta) = 1$  for all of the spanning  $\chi_{q,s}$ . If  $\beta$  passes that test  $\phi(\beta) = 1$  for all  $\phi \in \text{Hom}(V/\mathbb{Q}[\alpha]^{*2}, \{\pm 1\})$ . Since there is some nontrivial homomorphism  $\text{Hom}(V/\mathbb{Q}[\alpha]^{*2}, \{\pm 1\})$  this implies that  $\beta = 0$  in the quotient and we are done.

In [3] it is proved that the quotient group can be written as a vector space over  $\mathbb{F}_2$  with dimension  $< \log_2 n$ , so it is actually possible to find a spanning set of  $\chi_{q,s}$ . Every pair  $(q, s)$  induces a prime ideal in  $\mathbb{Q}[\alpha]$ , and we could have stated all the above statements in

that notation (which is actually done in [2]). The Chebotarev density theorem deals with the distribution of these ideals, and gives us that the  $\chi_{q,s}$  are evenly distributed among the nontrivial elements of  $\text{Hom}(V/\mathbb{Q}[\alpha]^{*2}, \{\pm 1\})$  asymptotically. If we assume that also holds in our finite case we can assume that the  $\chi_{q,s}$  we have picked is randomly distributed in  $\text{Hom}(V/\mathbb{Q}[\alpha]^{*2}, \{\pm 1\})$ . The following lemma from [2] tells us, how many  $\chi_{q,s}$  we should pick.

**Lemma 2.5.** *Let  $k, l > 0$  be integers, and let  $W$  be a  $k$ -dimensional vector space over  $\mathbb{F}_2$ . Let  $P$  be the probability that  $l$  randomly picked nontrivial elements of  $W$  span  $W$ . Then*

$$P > 1 - 2^{k-l}.$$

Notice that if we see  $V/\mathbb{Q}[\alpha]^{*2}$  as a vector space over  $\mathbb{F}_2$ , then  $\text{Hom}(V/\mathbb{Q}[\alpha]^{*2}, \{\pm 1\})$  is the dual space, and thus has the same dimension over  $\mathbb{F}_2$ .

So we need to pick a little more than  $\log_2 n$  pairs  $(q, s)$ . In [1] they pick  $k = \lfloor 3 \log_2 n \rfloor$  and in [2]  $k = \lfloor \log_2 n \rfloor + 30$ , but it is not really important for the running time of the algorithm.

### 3 Finding square roots

At the end of the NFS we have that  $\prod_{(a,b) \in S} (a - bm) = v^2$  for  $v \in \mathbb{Z}$  and  $f'(\alpha)^2 \prod_{(a,b) \in S} (a - b\alpha) = \gamma^2$  for  $\gamma \in \mathbb{Z}[\alpha]$ . We now need to find  $v$  and  $\gamma$ . Actually we wan't to find  $u \in \mathbb{Z}$  such that  $u \equiv \phi(\gamma) \pmod{n}$  where  $\phi$  is the homomorphism from  $\mathbb{Z}[\alpha]$  to  $\mathbb{Z}_n$  defined by  $\phi(\alpha) = m$ .

It is rather simple to find  $v$ , since we calculated the prime factorization of  $\prod_{(a,b) \in S} (a - bm)$  when we filled up the rows of the matrix. Then  $v$  is the number with the same primefactorization with half as big exponents.

It's harder to find  $\gamma$ . The method here is mentioned in [1] and [4]. Assume we have written

$$\gamma = a_0 + a_1\alpha + \cdots + a_{d-1}\alpha^{d-1},$$

$u$  will then be defined by

$$u = a_0 + a_1m + ma_{d-1}m^{d-1}.$$

Since  $\gamma^2$  is known, we know  $u^2$ . Since we are only interested in  $v \pmod{n}$  we can calculate  $a_j$  modulo  $n$ . This is good, since the numbers possibly are extremely large.

The idea is to calculate  $u$  modulo a bunch of primes and then use the Chinese Remainder Theorem. We then get a  $y$  such that  $u \equiv y \pmod{P}$ , where  $P$  is the product of the primes we chose.

There is one problem with tis approach. For each prime  $p$  we calculate the square root modulo  $p$ , but there are two choices with opposite sign, and we do not know which one to choose. If  $d$  (the degree of the polynomial  $f$ ) is odd, there is a way around this, since we then have  $N(-1) = -1$ . The norm on  $\mathbb{Q}[\alpha]$  was defined by

$$N(s_0 + s_1\alpha + \cdots + s_{d-1}\alpha^{d-1}) = \prod_{j=1}^d (s_0 + s_1\alpha_j + \cdots + s_{d-1}\alpha_j^{d-1}),$$

where  $\alpha_1, \dots, \alpha_d$  is the roots of  $f$ . When we calculated the rows of the matrix, among other things, we calculated  $\mathfrak{v}(a - b\alpha)$ , which actually gives us the prime factorization of  $N(\beta)$  and hence it is easy to get the prime factorization of  $N(\gamma)$ . For each of the primes  $p$  we chose earlier, we find  $\gamma_p$  such that

$$\gamma_p^2 \equiv \gamma^2 \pmod{p}.$$

We now need to choose  $\gamma_p$  or  $-\gamma_p$ . But in this case that is easy, since we know the  $N(\gamma) \pmod{p}$ . We then choose the one, that has norm congruent to  $N(\gamma)$  modulo  $p$ . This does not generalize to  $d$  even, so we need some other method.

If not done carefully, the numbers will get very large, and as a consequence the computation of the square root will take quite some time to run through. But there are fast methods that seem to work in practice. So there are ways of doing this that does not have any mentionable consequences for the overall running time.

## 4 A newer way of finding square roots

I have studied a rather new implementation of the NFS by Jason Papadopoulos (URL: <http://www.boon.net/~jasonp/qs.html>), his method is more or less brute force. That is quite different from what I have described, since the computational power available today is quite different from what it was when the articles I refer to was written. As an example, he does not do as much work to keep the coefficients from begin large.

He defines a new polynomium  $S(x)$  by

$$S(x) = \prod_{(a,b) \in S} (a - bx) \pmod{f(x)}.$$

He then finds the squareroot  $T(x)$  of  $S(x)$  in  $\mathbb{Z}[x]/(f(x))$  by som kind of Newton iteration. It is enough to calculate modulo  $f(x)$  since we need our result modulo  $n$ . We now have something like  $v = f'(m)T(m) \pmod{n}$ .

This is rather brute force, but it works well on faster computers.

## References

- [1] Richard Crandall & Carl Pomerance (2005), *Prime Numbers, A Computational Perspective*, 2. edition, Springer
- [2] Byrnes, Steven (2005), *The Number Field Sieve*,  
URL: [http://modular.fas.harvard.edu/129-05/final\\_papers/Steve\\_Byrnes.pdf](http://modular.fas.harvard.edu/129-05/final_papers/Steve_Byrnes.pdf)
- [3] J. P. Buhler, H. W. Lenstra Jr. & Carl Pomerance (1993), *Factoring integers with the number field sieve*,  
URL: [https://openaccess.leidenuniv.nl/bitstream/1887/2149/1/346\\_114.pdf](https://openaccess.leidenuniv.nl/bitstream/1887/2149/1/346_114.pdf)
- [4] Briggs, Matthew E. (1998), *An Introduction to the General Number Field Sieve*, Master Thesis from Virginia Polytechnic Institute and State University,  
URL: <http://scholar.lib.vt.edu/theses/available/etd-32298-93111/unrestricted/etd.pdf>