

# Possible Ways to Extend Zhang's Special Quadratic Sieve

Eric Landquist

June 4, 2003

## 1 Introduction

In this paper we present ideas which could be used to extend Zhang's Special Quadratic Sieve (SQS), an extension of the Quadratic Sieve (QS) designed to factor integers of the form  $n = m^3 + a_2m^2 + a_1m + a_0$ , where  $m \sim n^{1/3}$ ,  $a_i = O(n^\epsilon)$ , for all  $i$ , and  $\epsilon > 0$  is very small. We define the subexponential function  $L_n[\alpha] = \exp(\alpha\sqrt{\log n \log \log n})$ . Recall the asymptotic running time for QS is  $L_n[1+o(1)]$ . By comparison, SQS runs in time  $L_n[(\sqrt{6}/3) + o(1)] \approx L_n[0.8165 + o(1)]$ . It is hoped that by generalizing SQS further that this time will decrease for special integers or be applied to general integers.

## 2 Idea of Factoring

What most modern factoring algorithms do is create a congruence  $X^2 \equiv Y^2 \pmod{n}$  so that  $(X - Y)|n$  and  $(X + Y)|n$ , and that the divisors found are nontrivial. First, though, one needs a polynomial  $Q(x)$ , which is a square  $\pmod{n}$ . We sieve some interval with this polynomial, picking out values,  $Q(x)$  whose prime factors are all small, that is, we pick out smooth values of  $Q(x)$ . From these we can form a product of the  $Q(x)$  which is a square, and thus create a congruence as above. For convenience, we give a brief description below of how QS and SQS do this.

## 2.1 The Quadratic Sieve

Define the function  $Q(x) = (x - \lfloor \sqrt{n} \rfloor)^2 - n$ . Then it is clear that  $Q(x)$  is a square (mod  $n$ ). We define a set of primes  $F = \{p_i | p_i < B, \left(\frac{n}{p}\right) = 1\}$ , bounded by some  $B$ . This is our *factor base*. We pick out via a sieve the values of  $Q(x)$  which are *B-smooth*, that is only divisible by primes in  $F$ . We form a matrix in which the rows are the smooth elements, the columns are the factor base primes, and each entry is the exponent to which that prime occurs (mod 2) in the prime factorization of  $Q(x)$ . When there are more smooth elements than factor base primes, then we find a nontrivial kernel element,  $\vec{k}$ . Notice that the nonzero entries in  $\vec{k}$  correspond to smooth  $Q(x)$  such that the prime factorization of the product of those  $Q(x)$  is 0 (mod 2). In other words, the product of those  $Q(x)$  is a perfect square. Then we can set up the congruence  $X^2 \equiv Y^2 \pmod{n}$ , and perhaps find nontrivial factors of  $n$ .

Much more detail is given in many other sources as well as several speed-ups such as using numbers smooth with the exception of one or two large primes, using multiple polynomials (MPQS), and rapid switching of these polynomials (SIQS). However for the purpose of explaining SQS, I hope this is sufficient.

## 2.2 Zhang's Special Quadratic Sieve

As mentioned previously, we need  $n$  to be of the special form  $n = m^3 + a_2m^2 + a_1m + a_0$ . We can represent any integer  $n$  in this form several different ways, but in general it will be slower than QS unless the  $a_i$  are all very small compared to  $m$ , as we will see below.

Given the form of  $n$  above, the trick is to let

$$x = b_2m^2 + b_1m + b_0, \quad b_i \in \mathbb{Z}.$$

Then

$$x^2 = b_2^2m^4 + 2b_1b_2m^3 + (2b_0b_2 + b_1^2)m^2 + 2b_0b_1m + b_0^2.$$

We will make the right hand side a lot nicer by making a couple substitutions. Notice that

$$m^3 = n - (a_2m^2 + a_1m + a_0) \equiv -(a_2m^2 + a_1m + a_0) \pmod{n},$$

and

$$\begin{aligned}
m^4 &\equiv -(a_2m^3 + a_1m^2 + a_0m) \pmod{n} \\
&\equiv -(a_2(-(a_2m^2 + a_1m + a_0)) + a_1m^2 + a_0m) \pmod{n} \\
&\equiv (a_2^2 - a_1)m^2 + (a_1a_2 - a_0)m + a_0a_2 \pmod{n}
\end{aligned}$$

so that

$$x^2 \equiv c_2m^2 + c_1m + c_0 \pmod{n},$$

where

$$\begin{aligned}
c_2 &= (a_2^2 - a_1)b_2^2 - 2a_2b_1b_2 + b_1^2 + 2b_0b_2 \\
c_1 &= (a_1a_2 - a_0)b_2^2 - 2a_1b_1b_2 + 2b_0b_1 \\
c_0 &= a_0a_2b_2^2 - 2a_0b_1b_2 + b_0^2.
\end{aligned}$$

Recall that since  $m \sim n^{1/3}$ , we want  $c_2 = 0$  so that  $x^2 \pmod{n}$  is not too much larger than  $n^{1/3}$ . The way to do this is to parametrize the  $b_i$ . We use a two-variable parametrization:

$$\begin{aligned}
b_2 &= 2u^2 \\
b_1 &= 2(uv + a_2u^2) \\
b_0 &= a_1u^2 - v^2,
\end{aligned}$$

where  $u, v \in \mathbb{Z}$ , but with the added restrictions that  $(u, v) = 1$  and  $u > 0$ , since otherwise we will produce redundant relations. Our sieving polynomial is a quartic:

$$Q(u, v) = d_0u^4 + d_1u^3v + d_2u^2v^2 + d_3uv^3 + v^4,$$

where

$$\begin{aligned}
d_0 &= -4a_0(m + a_2) + a_1^2 \\
d_1 &= -4(a_1m + 2a_0) \\
d_2 &= -2(2a_2m + a_1) \\
d_3 &= -4m.
\end{aligned}$$

Assuming that  $u$  and  $v$  grow to be much larger than the  $a_i$ , and are on the order of  $n^\epsilon$ ,  $Q(u, v)$  will be dominated by any term, except for  $v^4$ , and will

be on the order of  $n^{1/3+4\epsilon}$ . We endeavor to keep this below  $n^{1/2}$  in order to have a theoretical speed-up over QS, so we want  $\epsilon \leq 1/24$ . This establishes our congruence  $X^2 \equiv Q(u, v) \pmod{n}$ . Since  $Q(u, v)$  will produce numbers smaller than the polynomial used in variants of QS, smooth numbers will be easier to find.

To perform the sieving step, we first need to find roots to the sieving polynomial, so notice that  $Q(u, v) = u^4 f(vu^{-1})$ , where

$$f(t) = d_0 + d_1 t + d_2 t^2 + d_3 t^3 + t^4.$$

Suppose  $p|Q(u, v)$ . It is clear that if  $p|u$ , then  $p|v$ , contradicting our restriction that  $(u, v) = 1$ . Assuming that  $p \nmid u$ , then  $p|f(vu^{-1})$ , and we find solutions to

$$f(t) \equiv 0 \pmod{p}.$$

So we sieve with  $Q(u, v)$ , where  $(u, v) = 1$ . The roots of  $Q(u, v)$  are easy to find, so switching polynomials is very easy. The rest of the details can be found in either of the references above and are similar to how the QS proceeds.

### 3 Extending this Idea

To extend this idea, Zhang tried factoring  $n = m^4 + c$  using a process similar to that above, but with no luck. Another idea is to consider  $n = m^5 + c$ , for  $c \in \mathbb{Z}$ . Let  $x = a_4 m^4 + a_3 m^3 + a_2 m^2 + a_1 m + a_0$ . We want  $x^2 \equiv Q(u_1, u_2, \dots, u_k) \pmod{n}$ , for some parameters  $u_1, u_2, \dots, u_k$ . By the process explained above, this boils down to solving the system:

$$\begin{cases} 2(a_0 a_2 - c a_3 a_4) + a_1^2 = 0 \\ 2(a_1 a_2 + a_0 a_3) - c a_4^2 = 0 \\ 2(a_1 a_3 + a_0 a_4) - a_2^2 = 0 \end{cases}$$

I have only found solutions when  $c = \pm 1, 7, 12, 24, 32, 44, 76, 296$ . In the case  $c = \pm 1$ , this method leads to the fairly trivial factorizations  $m^5 - 1 = (m - 1)(m^4 + m^3 + m^2 + m + 1)$  and  $m^5 + 1 = (m + 1)(m^4 - m^3 + m^2 - m + 1)$ .

For examples other than  $n = m^5 + c$ , one technique would be to use Gröbner Bases, since they can be used to solve systems of equations as above. That is, suppose we have a system of equations  $f_1 = 0, \dots, f_k = 0$ . We define the ideal  $I = \langle f_1, \dots, f_k \rangle \subset \mathbb{R}[\vec{x}]$ . We need to find the variety  $\mathbf{V}(I)$ , so try by

finding a Gröbner Basis,  $\mathbf{G} = \{g_1, \dots, g_l\}$  of  $I$ . It will be easier to solve  $g_i = 0$ , and if so full solutions to the whole system may then be available. Some varieties are parametrizable by rational functions, but we want polynomial parametrizations. Is it possible though to use smooth rational numbers? That is, if  $1/3$  is a factor, combine it with a smooth residue with 3 as a factor.

In general, what is hoped is that this method described, or some other cleverly chosen  $x$  will yield a polynomial  $Q$  with  $x^2 \equiv Q \pmod{n}$  such that the  $Q(\vec{u}) \approx n^{1/d}$  for some  $d > 3$  for some special (or not so special) form of  $n$ . Then the running time will likely be  $L_n[1/2, \sqrt{2d}/d + o(1)]$ , which is still asymptotically slower than NFS, but could be faster for numbers small enough to be able to factor given current limitations.

Another approach is to take a backwards approach. Create one or more sieving polynomials which are small over a long interval (or intervals), and then establish a congruence to factor some integer.

## References

- [1] Crandall, R. and Pomerance, C. *Prime Numbers, A Computational Perspective*, Springer, New York, (2001).
- [2] Zhang, M. *Factorization of the Numbers of the form  $m^3 + c_2m^2 + c_1m + c_0$* , Algorithmic Number Theory (Portland, OR, 1998), 131-136, Lecture Notes in Computer Science, 1423, Springer, Berlin, (1998).