Optimising the Quadratic Sieve

Christiana Mavroyiakoumou

Imperial College London

Introduction

The quadratic sieve is an integer factorisation algorithm created in 1981 by Carl Pomerance. It is the second fastest algorithm known to date and can be used to factorise numbers up to around 120 digits. Its runtime is sub-exponential.

Some Useful Definitions

A *quadratic residue* (mod n) is a number a such that

$$x^2 \equiv a \,(\mathrm{mod}\, n)$$

has a solution for $a, n \in \mathbb{N}$ coprime.
If the prime decomposition of n is $p_1{}^{a_1} p_2{}^{a_2} \dots p_k{}^{a_k}$, then the *exponent vector*, v, is the vector $(a_1, a_2, \dots, a_k)$.
For an odd prime p, the *Legendre Symbol* is defined as

$$\left(\frac{n}{p}\right) = \begin{cases} 1 & \text{if } n \text{ is a quadratic residue } (\mathrm{mod}\, p) \\ -1 & \text{if } n \text{ is a non-residue } (\mathrm{mod}\, p) \\ 0 & \text{if } n \equiv 0 \,(\mathrm{mod}\, p) \end{cases}$$

An integer n is *B-smooth* if all its prime factors are less than or equal to B. The quadratic sieve searches for *B-smooth* numbers.

Choosing B

Choosing B implies a tradeoff: a small B facilitates checking if a number is *B-smooth*, but makes it unlikely to find any. A large B increases the chance of finding *B-smooth* numbers but at the same time factorising each number becomes harder. Heuristic analysis showed that the best choice for the smoothness bound B is about $e^{(\frac{1}{2}\sqrt{\ln n \ln \ln n})}$.

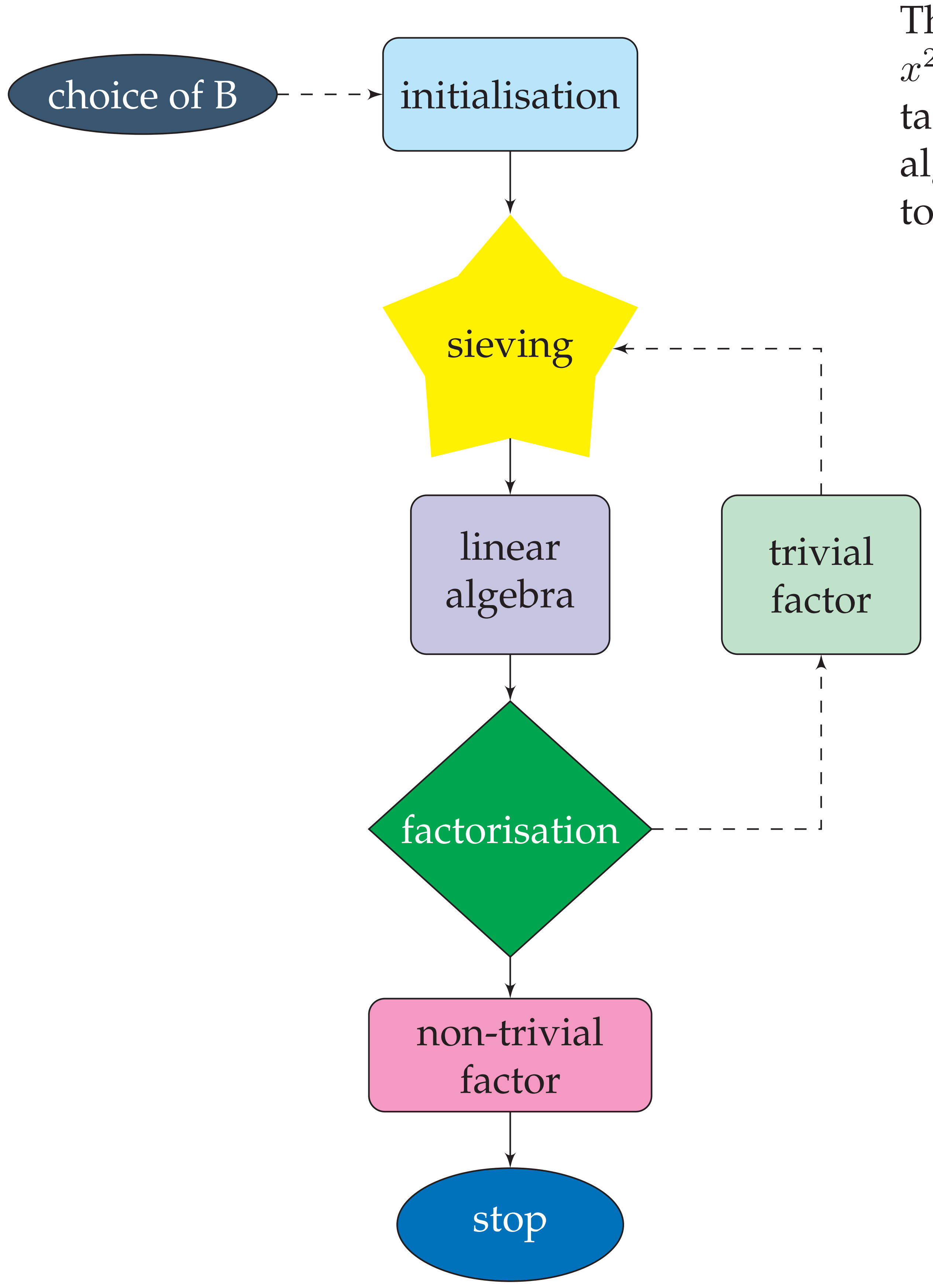
Legendre Symbol

The *Legendre Symbol* is computed using Euler's criterion:

$$\left(\frac{n}{p}\right) \equiv n^{(p-1)/2} \,(\mathrm{mod}\, p)$$

If the Legendre Symbol for a prime is 1 then include this prime in the factor base. Since primes giving -1 do not appear in the factorisations, discard them. About 50% of the primes satisfy the condition: checking for less prime factors makes the sieving step faster.

The basic Quadratic Sieve algorithm



choice of B

initialisation

sieving

linear algebra

trivial factor

factorisation

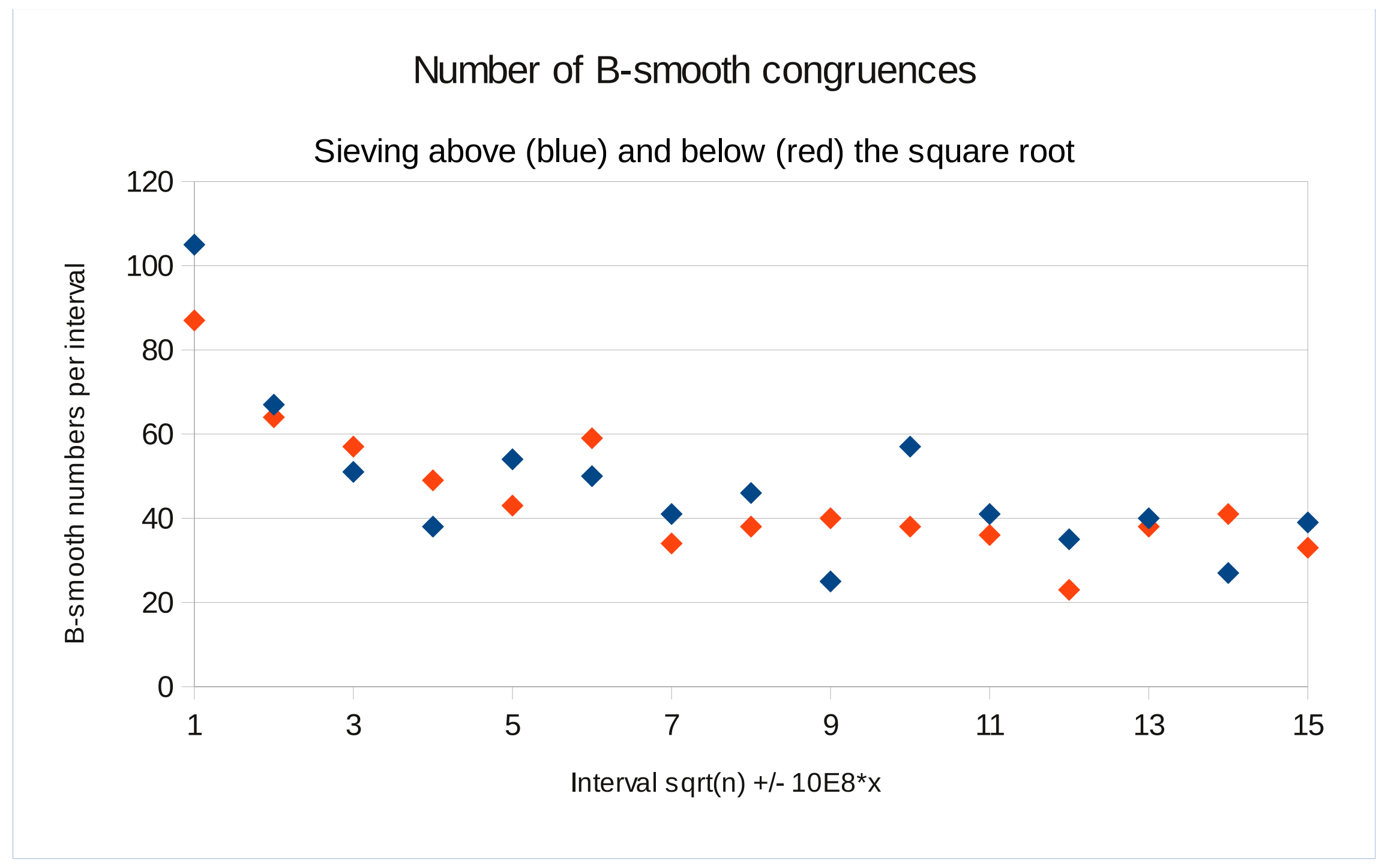
non-trivial factor

stop

The idea is that for x, y with $x \not\equiv \pm y \pmod n$ such that $x^2 \equiv y^2 \pmod n$, a non-trivial factor of n can be obtained via $\gcd(x \pm y, n)$. This gcd is found by Euclid's algorithm and the probability of the corresponding factor being non-trivial is at least $\frac{1}{2}$.

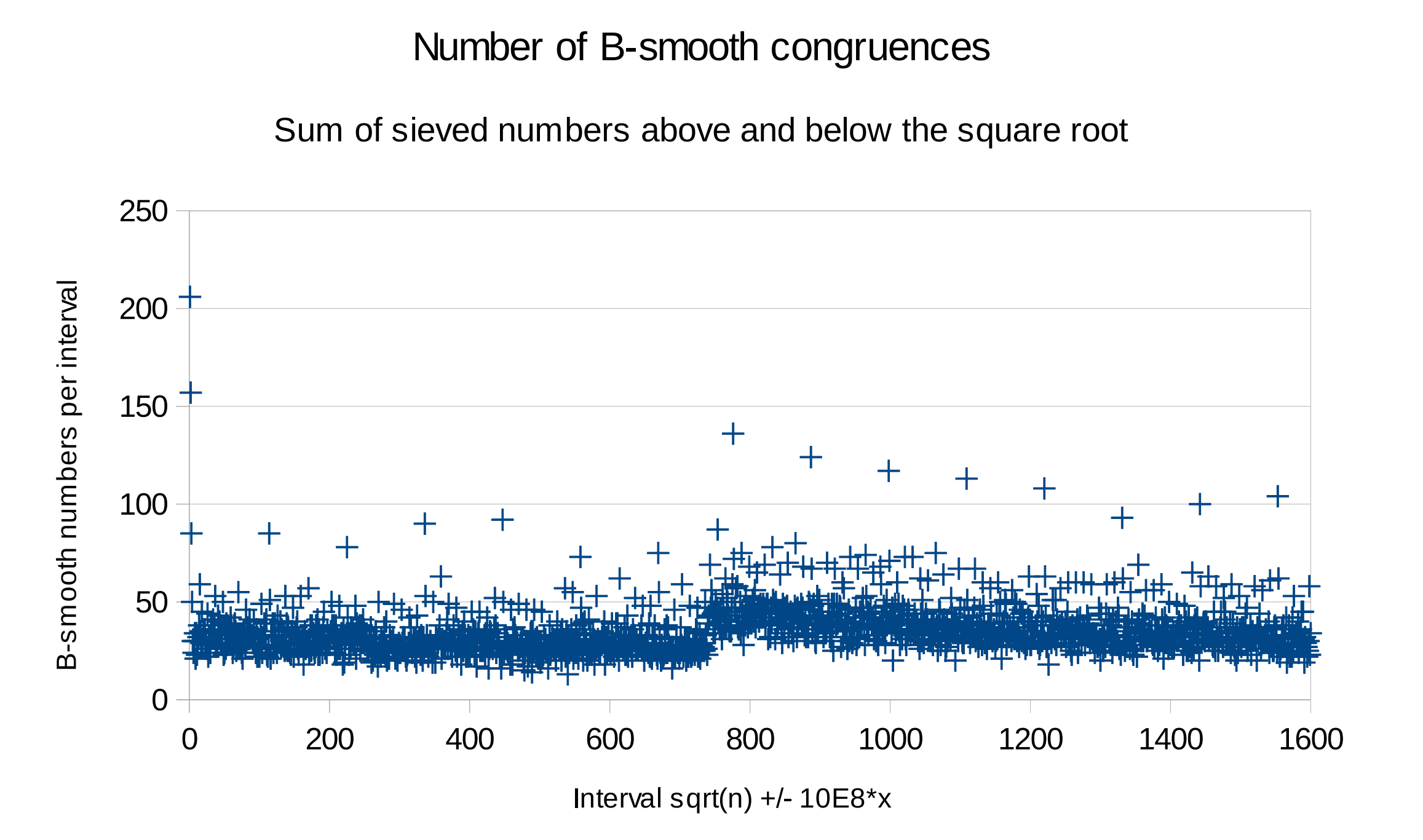
1. Choose B.
2. Form the **factor base** consisting of primes $p \leq B$ for which the *Legendre symbol* equals to 1.
3. Start with $x = \lceil \sqrt{n} \rceil$. Make an array of x^2 mod n, $(x+1)^2$ mod n, $(x+2)^2$ mod n, …and sieve for *B-smooth* numbers. Do this until a subset of at least $d+1$ smooth numbers is formed (to ensure linear dependency), where d is the dimension of the factor base.
4. Form a matrix with its columns being the exponent vectors of each *B-smooth* number mod 2.
5. Compute the kernel mod 2 with Gauss's method. A kernel vector combines the congruences in such a way as to give even exponents, thus a solution of $x^2 \equiv y^2 \pmod n$.
6. Calculate $x = x_1 x_2 \dots x_d$ mod n and $y = \sqrt{(x_1{}^2 - n)(x_2{}^2 - n)\dots(x_d{}^2 - n)}$ mod n.
7. The non-trivial factors of n are $a = \gcd(x \pm y, n)$.

Plots of *B-smooth* numbers



Number of B-smooth congruences

Sieving above (blue) and below (red) the square root



Number of B-smooth congruences

Sum of sieved numbers above and below the square root

Parameters:
n = 41081313706319975070882075012572981246 93

B = 25458.

Used Legendre for factor base but not large prime variation and the sieve method was logs with cutoff 20.

Sieving step

A variation of the **Sieve of Eratosthenes** instead of *trial division* is used to make the sieving step faster. Divide $x^2 \pmod n$ by each of the primes in the factor base and its powers. All the *B-smooth* numbers will be reduced to 1.
And again this can be improved further using logarithms which handle smaller numbers.

Symmetric Sieving

A symmetric sieve stays in a closer distance to $\lceil \sqrt{n} \rceil$ until the matrix is full. A sieve just above or just below needs to go further away from $\lceil \sqrt{n} \rceil$ to find the last *B-smooth* numbers.
This is not good because numbers get harder to factor and *B-smooth* numbers get rare.

Using Logarithms

The most time-consuming step is **sieving**, since it can be required to check a very large set of numbers to see if they are *B-smooth*. Using approximations of the logarithms of each prime being sieved, makes the sieving step more efficient.
The advantage of the method is that we save time by performing a *subtraction* instead of a *division*.

Large Prime Variations

In practice, it often occurs that congruences are almost *B-smooth* except having one larger prime factor, and therefore can't be used in the sieve. These numbers are stored in hope of being matched with another number having the same large prime factor, thus completing the square.
The probability of numbers in the list having the same pair of large primes is analogous to the **birthday paradox**. It says that in a group of 23 people, at least two of them have the same birthday with probability of more than 50%.

References

[1] Richard Crandall and Carl Pomerance A Computational Perspective. Springer, 2005.

[2] Stephani Lee Garrett On the Quadratic Sieve. 2008.

[3] Carl Pomerance Smooth numbers and the quadratic sieve. 2008.

[4] Carl Pomerance A tale of two sieves. 1996.