Refined Analysis of the Asymptotic Complexity of the Number Field Sieve

Aude Le Gluher, Pierre-Jean Spaenlehauer and Emmanuel Thomé
Université de Lorraine / INRIA Nancy – Grand Est / CNRS
CARAMBA team

March 2020

Motivations

The Number Field Sieve (NFS) is the most efficient method to factor integers or solve discrete logarithm problems.

Question

Given some computational power C, what should be the key sizes that ensure the cost of NFS will exceed C ?

Rely on the asymptotic complexity of NFS ?

NFS heuristical asymptotic complexity

Under various assumptions, the complexity of NFS to factor an integer N is

$$\exp\left(\sqrt[3]{\frac{64}{9}}(\log N)^{1/3}(\log\log N)^{2/3}(1+\xi(N))\right)$$

where $\xi(N) \in o(1)$ as N grows.

Rely on the asymptotic complexity of NFS ?

NFS heuristical asymptotic complexity

Under various assumptions, the complexity of NFS to factor an integer N is

$$\exp\left(\sqrt[3]{\frac{64}{9}}(\log N)^{1/3}(\log\log N)^{2/3}(1+\xi(N))\right)$$

where $\xi(N) \in o(1)$ as N grows.

Classical use of this formula : assume $\xi(N) = 0$. Is it reasonable ?

Rely on the asymptotic complexity of NFS ?

NFS heuristical asymptotic complexity

Under various assumptions, the complexity of NFS to factor an integer N is

$$\exp\left(\sqrt[3]{\frac{64}{9}}(\log N)^{1/3}(\log\log N)^{2/3}(1+\xi(N))\right)$$

where $\xi(N) \in o(1)$ as N grows.

Classical use of this formula : assume $\xi(N) = 0$. Is it reasonable ?

- Give insights on what $\xi(N)$ hides.
- Assess the relevance of the classical simplification $\xi(N) = 0$.

Main results

- Method to compute an asymptotic expansion of ξ which is a bivariate series S evaluated at $(\log\log\log N)/(\log\log N)$ and $1/(\log\log N)$. In particular,

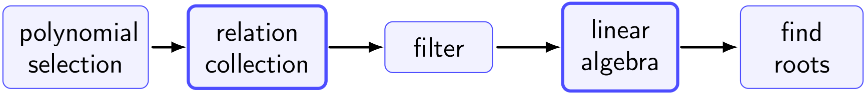
$$\xi(N) \sim \frac{4\log\log\log N}{3\log\log N}$$

- Algorithm that implements this method and computes the coefficients of S.
- Study of the convergence range of S. It is huge (around $e^{e^{25}}$), so using any approximation of ξ for N sizes relevant in cryptography means replacing ξ by the first terms of a divergent series...

Plan

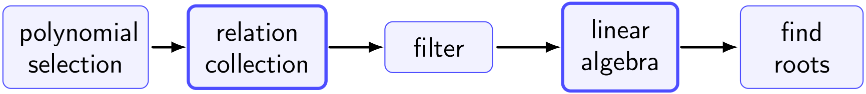
1. NFS complexity is the solution of an optimization problem
2. Smoothness formulas
3. Asymptotic expansion of ξ

NFS, briefly



To factor an integer N :

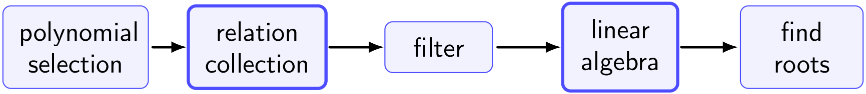
NFS, briefly



To factor an integer N :

- Build two number fields $K_0 = \mathbb{Q}[X]/(f_0)$ and $K_1 = \mathbb{Q}[X]/(f_1)$.

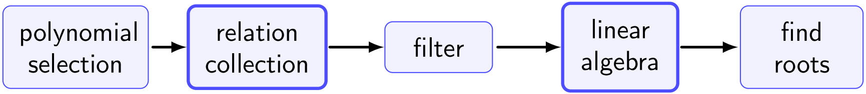
NFS, briefly



To factor an integer N :

- Build two number fields $K_0 = \mathbb{Q}[X]/(f_0)$ and $K_1 = \mathbb{Q}[X]/(f_1)$.
- Given integers (u, v) in a search space, check if the norm of $u - vX$ is smooth in K_0 and in K_1.

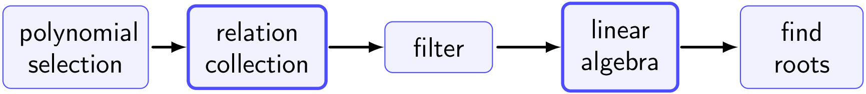
NFS, briefly



To factor an integer N :

- Build two number fields $K_0 = \mathbb{Q}[X]/(f_0)$ and $K_1 = \mathbb{Q}[X]/(f_1)$.
- Given integers (u, v) in a search space, check if the norm of $u - vX$ is smooth in K_0 and in K_1.
- If so, store the factorizations in a matrix. Densify.

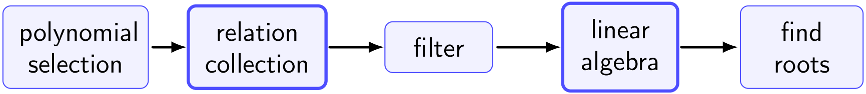
NFS, briefly



To factor an integer N :

- Build two number fields $K_0 = \mathbb{Q}[X]/(f_0)$ and $K_1 = \mathbb{Q}[X]/(f_1)$.
- Given integers (u, v) in a search space, check if the norm of $u - vX$ is smooth in K_0 and in K_1.
- If so, store the factorizations in a matrix. Densify.
- Compute the left kernel of the matrix : it gives $x^2 \equiv y^2 \bmod N$.

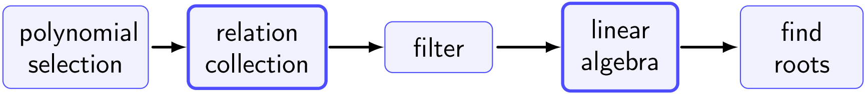
NFS, briefly



To factor an integer N :

- Build two number fields $K_0 = \mathbb{Q}[X]/(f_0)$ and $K_1 = \mathbb{Q}[X]/(f_1)$.
- Given integers (u, v) in a search space, check if the norm of $u - vX$ is smooth in K_0 and in K_1.
- If so, store the factorizations in a matrix. Densify.
- Compute the left kernel of the matrix : it gives $x^2 \equiv y^2 \bmod N$.
- Find x and y. With good probability, $\gcd(N, x - y)$ is a non trivial factor of N.

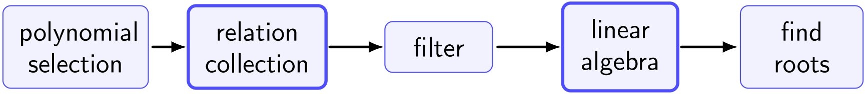
NFS, briefly



To factor an integer N :

- Build two number fields $K_0 = \mathbb{Q}[X]/(\mathbf{f_0})$ and $K_1 = \mathbb{Q}[X]/(\mathbf{f_1})$.
- Given integers (u, v) in a **search space**, check il the norm of $u - vX$ is **smooth** in K_0 and in K_1.
- If so, store the factorizations in a matrix. Densify said matrix.
- Compute the left kernel of the matrix : it gives $x^2 \equiv y^2 \mod N$.
- Find x and y. With good probability, gcd$(N, x + y)$ is a non trivial factor of N.

NFS, briefly



Parameters

- Degree of the polynomial : d.
- Size of the search space : a.
- Size of the smoothness bound : b.

Remark

The more costly steps are relation collection and linear algebra.

Optimization problem

Goal : find a, b, d such that they

- Minimize the cost of (relation collection + linear algebra).
- Satisfy a constraint that ensures that the matrix in the linear algebra step has a non trivial left-kernel *ie* (size of the search space) $\times$ (probability of smoothness in K_0) $\times$ (probability of smoothness in K_1) $\geq$ (number of primes below smoothness bound)

Simplified optimization problem

Find three functions of $\nu = \log N$, a, b, d that minimize $\max(a, b)$ under the constraint :

$$p(a + \nu/d, b) + p(da + \nu/d, b) + 2a - b = 0$$

Plan

Smoothness notations

Definition : smoothness

An integer is y-smooth if all its prime factors are below y.

Notations

- We let $\Psi(x, y) = \text{Card}(\{\text{integers in } [1, x] \text{ that are } y\text{-smooth}\})$. The probability for a random integer in $[1, x]$ to be y-smooth is $\Psi(x, y)/x$.
- We note $p(u, v) = \log(\Psi(e^u, e^v)/e^u)$.

A suitable formula for smoothness probabilities

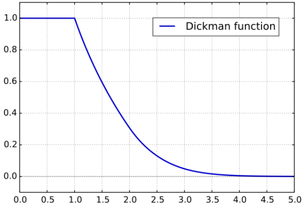
Classical analysis of the asymptotic complexity of NFS relies on a first order estimation of smoothness probabilities by Canfield Erdős and Pomerance (1983).

Hildebrand (1986) formula for smoothness probabilities

For x, y under circumstances satisfied in the NFS context, we have :

$$\frac{\Psi(x,y)}{x} = \rho(u)\left(1 + O\left(\frac{\log(u+1)}{\log y}\right)\right)$$

where ρ is the Dickman function and $u = \log x / \log y$.



Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

We have : $\rho(u) = \frac{e^{\gamma}}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.

Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

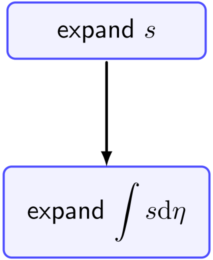
We have : $\rho(u) = \frac{e^\gamma}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.

expand s

Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

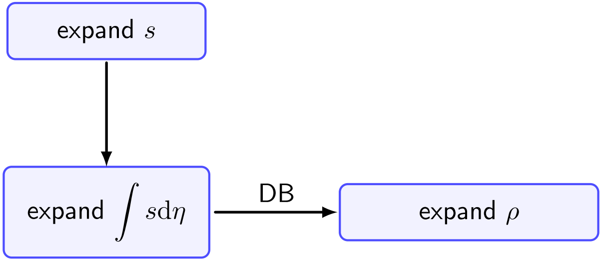
We have : $\rho(u) = \frac{e^{\gamma}}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.



Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

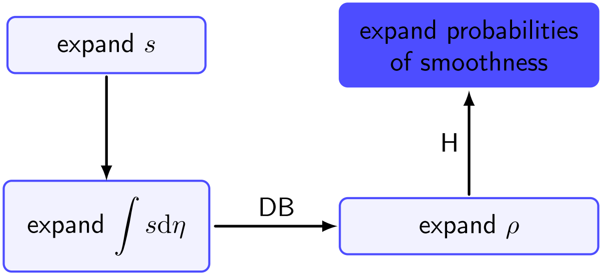
We have : $\rho(u) = \frac{e^\gamma}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.



Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

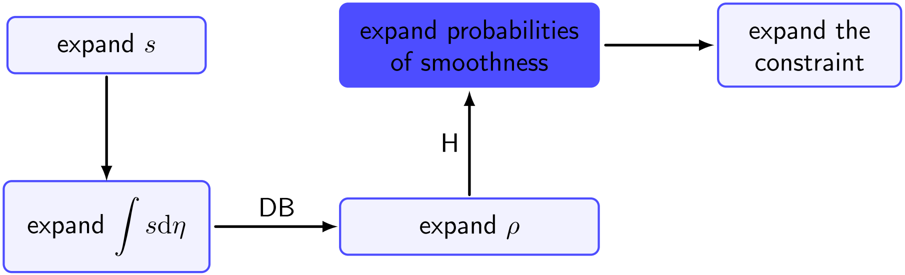
We have : $\rho(u) = \frac{e^{\gamma}}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.



Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

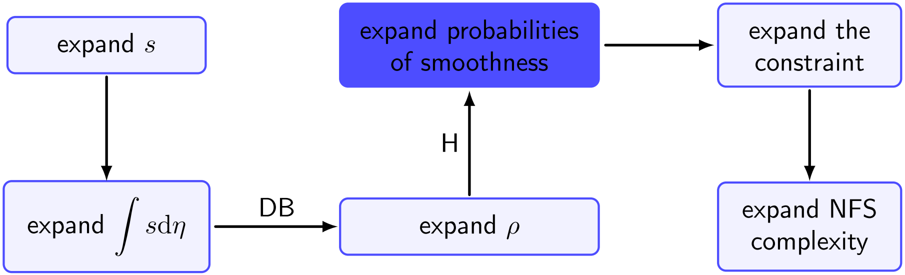
We have : $\rho(u) = \frac{e^\gamma}{\sqrt{2\pi u}} \exp\left(-\int_1^u s\mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.



Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

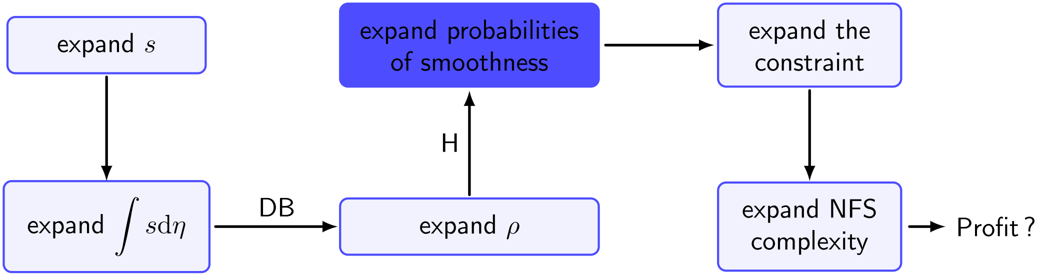
We have : $\rho(u) = \frac{e^{\gamma}}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.



Main steps to expand smoothness probabilities

De Bruijn (1951) formula for ρ

We have : $\rho(u) = \frac{e^{\gamma}}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.



Asymptotic expansion of ρ

De Bruijn (1951) formula for ρ

We have : $\rho(u) = \frac{e^\gamma}{\sqrt{2\pi u}} \exp\left(-\int_1^u s \mathrm{d}\eta\right)$ when $u \to +\infty$ and where $s = \log(1 + s\eta)$.

Method to obtain an asymptotic development of ρ

- Recursively expand s using the expansion of $x \mapsto \log(1+x)$ around 0.
  Proves that $\eta \mapsto s(\eta)/(\log \eta)$ can be expanded as a bivariate series evaluated in $(\log\log\eta)/(\log\eta)$ and $1/\log\eta$.
- Replace s by any of its expansions in the integral. Repeatedly integrate by parts.

Asymptotic expansion for smoothness probabilities

Shape of the asymptotic expansion of ρ

For all $n \in \mathbb{Z}_{\geq 0}$, as $N \to +\infty$ and in the optimal parameter range of NFS, the smoothness probabilities involved in the constraint are :

$$\frac{\Psi(x,y)}{x} = \exp\left(-u \log u \left(Q^{(n)}\left(\frac{\log \log u}{\log u}, \frac{1}{\log u}\right) + o\left(\frac{1}{(\log u)^n}\right)\right)\right)$$

where $Q^{(n)}$ is the truncation up to total degree n of the bivariate series Q and $u = \log x / \log y$.

- The bivariate series Q already appears in the development of ρ.
- The bivariate series Q can be explicitly computed and has coefficients in $\mathbb{Q}$.
- Residual factors in the formulas of Hildebrand and De Bruijn are swallowed in the $o(1/(\log u)^n)$.

Plan

Goal

We already know from the classical analysis of NFS complexity that the functions of $\nu = \log N$, a, b and d satisfy :

$$\begin{cases} a(\nu) & = (8/9)^{1/3}\nu^{1/3}(\log \nu)^{2/3}(1 + o(1)) \\ b(\nu) & = (8/9)^{1/3}\nu^{1/3}(\log \nu)^{2/3}(1 + o(1)) \\ d(\nu) & = (3\nu/\log \nu)^{1/3}(1 + o(1)) \end{cases}$$

Reminder

New terms in the expansions of a, b, d immediately yield new terms in the expansion of NFS complexity.

Step 1 : Find candidate expansions

Main ideas

- Assume more precision : the $o(1)$ are $O(\log\log\nu/\log\nu)$.
- Replace a, b, d by their values in the equation of the constraint.
- Solve the linear / quadratic constraint on the constants associated to the big O's.

This yields candidates to the optimization problem, denoted a_0, b_0, d_0.

Requirements

- Expansion of smoothness probabilities.
- Taylor series expansions of usual functions at infinity.
- Bivariate series computations at finite precision.

Step 2 : existence proof

Main idea

Prove the existence of functions satisfying the constraint and having the same development as a_0, b_0, d_0.

This yields a baseline result : any solution of the optimization problem must be smaller than a_0, b_0, d_0.

Requirements

Same as step 1.

Step 3 : minimality proof

Main ideas

- Prove that the $o(1)$ involved in the expansions of a, b, d known so far can actually be written $(C + o(1))(\log\log\nu/\log\nu)^{\lambda} \times (1/\log\nu)^{\mu}$.
- Prove that the constants C are the same than the ones in the expansions of the candidates a_0, b_0, d_0.

This proves a new term in the expansions of a, b, d.

Requirements

- Step 1 requirements.
- A baseline result (given by step 2).
- Proper patterns in the equations encountered during the proof.

Final shape of NFS complexity

Expansion of the heuristic complexity of NFS $C(N)$

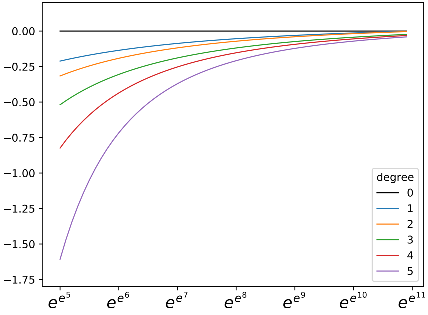
For all $n \in \mathbb{Z}_{\geq 0}$, $C(N)$ is :

$$\exp\left[\sqrt[3]{\frac{64}{9}}(\log N)^{1/3}(\log\log N)^{2/3}\left(1+\underbrace{A^{(n)}\left(\frac{\log\log\log N}{\log\log N},\frac{1}{\log\log N}\right)+o\left(\frac{1}{(\log\log N)^n}\right)}_{=\xi(N)}\right)\right]$$

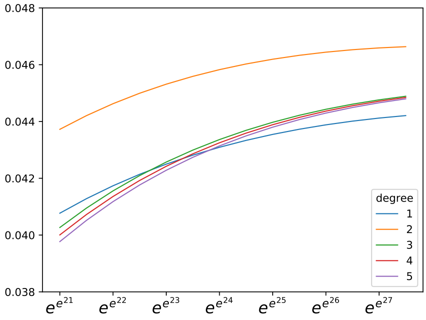
where $A^{(n)}$ is the truncation up to total degree n of the bivariate series A.

The coefficients of A are in $\mathbb{Q}[\log(2),\log(3)]$ and can be algorithmically computed.

A problem of convergence



Truncations of ξ up to total degree 5 for cryptographically relevant values of N.



Converging behaviour of the trucations of ξ.

Take home message

Replacing ξ by any truncated asymptotic expansion = replace a series by its first terms in a range where the series diverges !

The danger of replacing ξ by a truncation

Compare :

Function g	$g(2^{3072})/g(2^{829})$
$g_0 : N \mapsto \exp\left(\sqrt[3]{\frac{64}{9}}(\log N)^{1/3}(\log\log N)^{2/3}\right)$	$\sim 2^{59}$
$g_1 : N \mapsto \exp\left(\sqrt[3]{\frac{64}{9}}\frac{(\log N)^{1/3}(\log\log N)^{2/3}}{1+20/\log\log N}\right)$	$\sim 2^{19}$

Don't do $o(1) = 0$ carelessly...

The function g_1 is in $\exp\left(\sqrt[3]{\frac{64}{9}}(\log N)^{1/3}(\log\log N)^{2/3}(1+o(1))\right)$ but replacing the $o(1)$ by 0 (i.e. g_1 by g_0) for $N \leq e^{e^{20}}$ leads to drastically different results.

Sum up

- Expansion of the function hidden in the $o(1)$ in NFS complexity. See `https://arxiv.org/abs/2007.02730` for more details.
- Algorithm to compute this expansion. Available at : `https://gitlab.inria.fr/NFS_asymptotic_complexity/simulations`
- Be very careful when using truncated versions of the asymptotic complexity.
- Maybe use numerical estimates for ρ or simulations ?

Sum up

- Expansion of the function hidden in the $o(1)$ in NFS complexity. See https://arxiv.org/abs/2007.02730 for more details.
- Algorithm to compute this expansion. Available at : https://gitlab.inria.fr/NFS_asymptotic_complexity/simulations
- Be very careful when using truncated versions of the asymptotic complexity.
- Maybe use numerical estimates for ρ or simulations ?

Thank you for your attention !