Elementary topics in Computational algebraic number theory

Karim Belabas

Karim.Belabas@math.u-psud.fr

http://www.math.u-psud.fr/~belabas/

Université Paris-Sud

France

Setup (1/4)

Let F be a number field. There are many interesting things we can compute about F:

- Invariants: maximal order $\mathcal{O}_F$, class group $\mathrm{Cl}(F)$, units $U(F)$, higher algebraic K-groups, Dedekind $\zeta_F\dots$
- Subfields: Galois group, lattice of subfields.
- Extensions: build L/F, e.g given explicitly by primitive elements or implicitly via Kummer or class field theory. Invariants thereof (e.g in class field towers).

Basic operations: elementary operations on elements and ideals of $\mathcal{O}_F$, mostly multiplications (at least in class field theory).

Setup (2/4)

For most of these problems, there exist efficient algorithms, deterministic or randomized, possibly assuming some deep conjecture (GRH, density of friable elements in appropriate sets…), possibly giving a wrong result with small probability in an appropriate model, possibly not an algorithm at all but *usually* giving sensible results…

But there are a number of pitfalls, especially when the degree $n = [F : \mathbb{Q}]$ gets large, introducing spurious bottlenecks in otherwise sensible computations.

Some pathologies:

- Randomization trouble: good expected cost but bad worst-case behaviour, sometimes inherent to a given instance.
- Coefficient explosion in intermediate, and final, results (polynomial number of operations, but operands of exponential size).
- Numerical instability

Setup (3/4)

Silly example: in $\mathbb{Z}/N\mathbb{Z}$ or $(\mathbb{Z}/N\mathbb{Z})^*$, in order to compute

$$x^k \pmod N$$

for some $k \geqslant 2$, it is advisable to use smallest non-negative (or centered) residues in $\mathbb{Z}$, and to *reduce* intermediate results modulo N whenever possible, *not* at the very end. Preconditioning on N also helps: Montgomery multiplication, FFT representation for a suitable approximation of $1/N$ (dyadic or floating point).

(Minor) Pitfall: on the other hand, when computing

$$\sum_i a_i b_i \pmod N,$$

reduce at the very end, *not* after each multiplication!

Setup (4/4)

We shall explain a number of « folklore » techniques generalizing the obvious part of the $\mathbb{Z}/N\mathbb{Z}$ example, especially when $n := [F : \mathbb{Q}]$ is large. The focus is on class field theoretic examples, in particular the computation of class fields, but the methods are widely applicable.

Some precomputations (integral basis for $\mathcal{O}_F$, its LLL-reduction and multiplication table…) are expensive, and certainly not universally desirable. They are skipped or tuned down when tackling « easier » tasks like e.g. factorization over $F[X]$.

Elements in $F = \mathbb{Q}[X]/(T)$ (1/2)

How to represent the elements of F ? It is generally worth it to separate contents / primitive parts and only deal with integral objects. Then we have

- polynomial representation $F = \mathbb{Q}[X]/(T)$, where T is integral and monic.
- basis representation $F \simeq \mathbb{Q}^{[F:\mathbb{Q}]}$. Often, pick a $\mathbb{Z}$-basis for $\mathcal{O}_F$ as a $\mathbb{Q}$-basis for F.
- regular representation $\begin{array}{rcl} F & \longrightarrow & \mathrm{Hom}_{\mathbb{Q}}(F, F) \\ x & \longmapsto & m_x := \text{multiplication by } x \end{array}$
- embeddings: archimedean ($F \otimes \mathbb{R}$) or p-adic ($F \otimes \mathbb{Q}_p$), truncated to some fixed accuracy.
- unevaluated formal product $x = \prod e_i^{n_i} \in \mathbb{Z}[F^*]$ of elements in any of the above forms ($n_i \in \mathbb{Z}$, we actually take $e_i \in \mathcal{O}_F \setminus \{0\}$).

Elements in $F = \mathbb{Q}[X]/(T)$ (2/2)

Let $n := [F : \mathbb{Q}]$. As far as multiplication goes, all representations are useful:

- polynomial yields a $2n^2$ method, and asymptotically better when n or the element's heights increase. But it has denominators even for algebraic integers. Over $\mathcal{O}_F$, denominators are bounded by the exponent of the additive group $\mathcal{O}_F/(\mathbb{Z}[X]/(T))$, which may be large.
- multiplication xy in basis representation first computes regular representation m_x or m_y (n^3 method). Knowing m_x makes multiplication by x an n^2 method. Useful if about $n/2$ multiplications by the *same* x are needed.
- embeddings cancel *intermediate* coefficient explosion, but suffer from stability problems. Requires *final* coefficients of bounded height for unique reconstruction. Archimedean embeddings introduce further rounding problems, but may be used in low accuracy as height estimator.
- formal representation defers actual computations to later stages. Hardly ever evaluated directly in F^*. Rather in $(\mathcal{O}_F/\mathfrak{f})^*$, $F \otimes \mathbb{R}$ (stable), $F \otimes \mathbb{Q}_p$, $F^*/(F^*)^\ell$...

[Multiplication] Ideals (1/2)

A fractional ideal is also best separated into content and primitive part. The latter is integral and can be given as

- a $\mathbb{Z}$-module: n generators.
- an $\mathcal{O}_F$-module: 2 generators. Requires solving an approximation problem.

Assuming one of $\mathfrak{a}$ and $\mathfrak{b}$ is given by two $\mathcal{O}_F$-generators, the multiplication $\mathfrak{ab}$ takes $O(n^3)$ elementary operations modulo $(\mathfrak{a} \cap \mathbb{Z})(\mathfrak{b} \cap \mathbb{Z})$. Otherwise $O(n^4)$.

In fact, thanks to the LLL algorithm, it is relatively easy to extract a large « principal part » from an ideal, rather than simply a content:

Definition: The $T_2 : F \to \mathbb{R}^+$ quadratic form is defined by

$$T_2(x) := \sum_{\sigma : F \to \mathbb{C}} |x|_\sigma^2$$

A $\mathbb{Z}$-submodule Λ of F becomes a lattice when equipped with T_2.

Let $\mathfrak{A}$ a non-zero fractional ideal. The first vector of an LLL-reduced basis for $\mathfrak{A}$ is an $\alpha \in \mathfrak{A}$ of relatively small norm. Rewrite

$$\mathfrak{A} = (\alpha)(\mathfrak{A}/\alpha) = (a)(\alpha)\mathfrak{a},$$

where $\mathfrak{a}$ is integral and primitive, $\alpha \in \mathcal{O}_F$ and $a \in \mathbb{Q}^*$. All three components depend on the specific LLL-reduction variant used, but

Lemma: *$N\mathfrak{a}$ is bounded by a constant depending only on F.*

So, any product of ideals can be represented in the form $(\alpha)\mathfrak{a}$, where α is an accumulated formal product in $\mathbb{Z}[F^*]$, and $\mathfrak{a}$ is a *small* integral ideal.

Example: discrete log in $\mathrm{Cl}(F)$

Input: An ideal I, possibly given as a product of ideals. We are given $\mathrm{Cl}(F) = \bigoplus(\mathbb{Z}/d_i\mathbb{Z})g_i$.

Output: (e_j) and $\tau \in \mathbb{Z}[F^*]$, such that $I = \tau \prod g_j^{e_j}$.

(1) Compute I as $(\alpha)\mathfrak{a}$, $\alpha \in \mathbb{Z}[F^*]$, $\mathfrak{a} \subset \mathcal{O}_F$.

(2) Solve discrete log problem for *small* ideal $\mathfrak{a}$ in $\mathrm{Cl}(F)$ as $\mathfrak{a} = (\tau) \prod g_i^{e_i}$, for some yet unknown principal ideal (τ).

(Multiply $\mathfrak{a}$ by random products of prime ideals in the factor base used to compute the class group, then reduce as in previous slide, until the ideal component of the reduction is smooth.)

(3) Compute $\mathfrak{a} \prod g_i^{-e_i}$, as $(\beta)\mathfrak{b}$, $\beta \in \mathbb{Z}[F^*]$.

(4) Realize *small* principal ideal $\mathfrak{b}$ as (γ), using same method as in Step (2), but this time computing logarithmic distance components. Yields the Archimedean embeddings of γ, from which γ is recovered algebraically.

(5) Output (e_i) and $\tau := \alpha\beta\gamma \in \mathbb{Z}[F^*]$.

Uniformizers (1/4)

Let $\mathfrak{f}$ a non-zero integral ideal, and $\wp/p$ a maximal ideal.

- An integer $\pi \in \mathcal{O}_F$ is an $\mathfrak{f}$-uniformizer for $\wp$, if $v_{\wp}(\pi) = 1$ and $v_{\mathfrak{q}}(\pi) = 0$, for all $\mathfrak{q} \mid \mathfrak{f}$, $\mathfrak{q} \neq \wp$. [In particular $\wp = p\mathcal{O}_F + \pi\mathcal{O}_F$.]
- A $\wp$-integer $\tau \in \mathcal{O}_{F,\wp}$ is an anti-uniformizer for $\wp$, if $v_{\wp}(\tau) = -1$.
- For a given anti-uniformizer τ, define the $\wp$-coprime part as

$$cp_{\wp}(x) := x\tau^{v_{\wp}(x)}$$

(*evaluated*, maps $\mathcal{O}_F \setminus \{0\}$ to $\mathcal{O}_F \setminus \wp$)

Lemma: *Let $\wp/p$ a prime ideal, π a (p)-uniformizer for $\wp$, and $\tau_0 \in \mathcal{O}_F$ such that $\pi\tau_0 \equiv 0 \pmod{p}$, and $p \nmid \tau_0$. Then $\tau = \tau_0/p$ is an anti-uniformizer.*

In other words, any non trivial τ_0 in $\mathrm{Ker}(m_\pi \otimes \mathbb{F}_p)$ will do. Any anti-uniformizer yields an obvious algorithm to compute $v_{\wp}(x)$ for $x \in \mathcal{O}_F \setminus \{0\}$: multiply x by τ while result is integral.In fact, we obtain $cp_{\wp}(x)$ as a byproduct.

This method is quite efficient if the valuation is small, which is guaranteed if we prevent coefficient explosion.

Uniformizers (2/4)

How to find a (p)-uniformizer π: the definition implies $\wp = p\mathcal{O}_F + \pi\mathcal{O}_F$. Recall that F is defined over $\mathbb{Q}$ by a monic integral $T(X)$. If p does not divide the index $[\mathcal{O}_F : \mathbb{Z}[X]/(T)]$, Kummer criterion applies and the answer is trivial.

If not, the Buchman-Lenstra variant of Berlekamp's algorithm splits the étale algebra $\mathcal{O}_F/I_p$, where I_p is the p-radical of $\mathcal{O}_F$. The ideal I_p is the lift to $\mathcal{O}_F$ of the radical of $\mathcal{O}_F/(p)$:

$$I_p = \bigcap_{\mathfrak{q}|p} \mathfrak{q} = \prod_{\mathfrak{q}|p} \mathfrak{q} = \{x \in \mathcal{O}_F, x \text{ nilpotent in } \mathcal{O}_F/(p)\}\,.$$

Let $x \to \overline{x}$ denote the projection from $\mathcal{O}_F$ to $\mathcal{O}_F/I_p$. The splitting yields all the $\overline{\mathfrak{q}} \subset \mathcal{O}_F/I_p$ as $\mathbb{F}_p$-vector spaces.

Uniformizers (3/4)

Lemma: *An $x \in \wp$ chosen uniformly at random is a p-uniformizer with probability $\prod_{\mathfrak{q}|p}(1 - 1/N\mathfrak{q})$.* [more rigorously, take $x \in \wp \mod p^2$]

What if p is small and has many prime divisors? Worst case: $p = 2$ totally split. Then expected running time is 2^n trials, with $n = [F : \mathbb{Q}]$. Each trial consists in the computation of Nx and the check « $pN\wp \mid Nx$? ».

Theorem: *There exists a deterministic algorithm producing a (p)-uniformizer for $\wp$ in at most n trials, and exactly 1 if p is known to be unramified.*

Let $g := \#\{\mathfrak{q} : \mathfrak{q} \mid p\}$. The extra cost is dominated by the computation of $\overline{I_p/\wp}$, using $g - 1$ intersections of $\mathbb{F}_p$-vector spaces in dimension at most n, that is $O(n^4)$ operations in $\mathbb{F}_p$. The total cost for *all* (p)-uniformizers is only a constant factor worse due to amortization: $3g - 4$ intersections in all.

Proof. Use an approximation argument and $\mathfrak{a}\mathfrak{b} = \mathfrak{a} \cap \mathfrak{b}$ for two coprime integral ideals $\mathfrak{a}$ and $\mathfrak{b}$. Translates ideal multiplication to intersection of $\mathbb{F}_p$-vector spaces: $\overline{\mathfrak{a}\mathfrak{b}} = \overline{\mathfrak{a}} \cap \overline{\mathfrak{b}}$. □

Uniformizers (4/4)

Input: $\overline{\wp}$ and $\overline{I_p/\wp}$, subspaces of $\mathcal{O}_F/I_p$ given by $\mathbb{F}_p$-bases.

Output: a p-uniformizer for $\wp$.

(1) Compute $(u, v) \in \wp \times I_p/\wp$ such that $u + v = 1 \pmod{p}$. [Simple $\mathbb{F}_p$-linear algebra: $O(n^3)$].

(2) At this point, we have $v_{\mathfrak{q}}(u) = 0$ for all $\mathfrak{q} \mid p$, $\mathfrak{q} \neq \wp$, and $v_\wp(u) \geqslant 1$.

- (a) [succeeds iff $v_\wp(u) = 1$] Let $x := u$. If $pN\wp \nmid Nx$, return x.
- (b) [succeeds iff $e(\wp/p) = 1$] Let $x := u + p$. If $pN\wp \nmid Nx$, return x.

(3) Let $\gamma_1, \dots, \gamma_k \in \mathcal{O}_F$ be lifted $\mathbb{F}_p$-generators of $\overline{\wp}$. For $i = 1, \dots, k-1$, repeat

- (a) [succeeds iff $\gamma_i \notin \wp^2$] Let $x := v\gamma_i + u \pmod{p}$. If $pN\wp \nmid Nx$, return x.

(4) Return $x := v\gamma_k + u \pmod{p}$.

Extended Euclidean Algorithm

Let $\mathfrak{a}$, $\mathfrak{b}$ be two coprime integral ideals. There exists $\alpha \in \mathfrak{a}$, $\beta \in \mathfrak{b}$ such that $\alpha + \beta = 1$. A variant of the modular HNF algorithm modulo $b := \mathfrak{b} \cap \mathbb{Z}$ (resp. $a := \mathfrak{a} \cap \mathbb{Z}$) yields α (resp. β).

More precisely, assume $\mathfrak{a}$, $\mathfrak{b}$ are given by matrices of generators with respect to a fixed basis $(w_1 = 1, \dots, w_n)$ of $\mathcal{O}_F$. Full HNF reduction would compute

$$U = \begin{pmatrix} U_{\mathfrak{a}} \\ U_{\mathfrak{b}} \end{pmatrix} \quad \text{such that} \quad (\mathfrak{a} \mid \mathfrak{b})U = (\mathrm{Id} \mid 0)$$

Let u the first column of $U_{\mathfrak{a}}$, we can set $\alpha := \mathfrak{a}u$. So, one

- only needs to keep track of $U_{\mathfrak{a}}$, not $U_{\mathfrak{b}}$.
- reduces modulo b at will, including the coordinates of $U_{\mathfrak{a}}$.
- can successively pair columns known to have fewest non-zero coefficients (if $\mathfrak{a}$ and $\mathfrak{b}$ happen to be in HNF), and stop as soon as 1 is found.

Frequent special case $(a, b) = 1$: only two columns are considered, in fact only a and b themselves to which ordinary extended Euclidean algorithm is applied.

Applications

Using the solution of (p)-uniformizer problem, the extended Euclidean algorithm and related ideas, one can efficiently solve [with solutions in $\mathbb{Z}[F^*]$]

- $\mathfrak{f}$-uniformizer problem for prime ideals.
- 2-generators problem for general ideals, i.e. write $\mathfrak{a} = a\mathcal{O}_F + b\mathcal{O}_F$ for any fixed $a \in \mathfrak{a} \setminus \{0\}$, in particular $a\mathbb{Z} = \mathfrak{a} \cap \mathbb{Z}$. Hence efficiently multiply by $\mathfrak{a}$. Picking $b \in \mathfrak{a}/(a)$ at random succeeds with probability

$$\prod (1 - 1/N\wp), \qquad \wp \in \{\wp : v_\wp(a) > v_\wp(\mathfrak{a})\}\,.$$

- Chinese remainder problem and general approximation. In particular, the coprime class problem: given integral non-zero ideals $\mathfrak{a}, \mathfrak{f}$, find $\alpha \in F^*$ such that $(\alpha\mathfrak{a}, \mathfrak{f}) = 1$.

Typical improvements when replacing naïve randomized approximation algorithms by the above for « real life[a] » fields of degree 20 to 30 : various computations ranging from a few hours to a few days done in a few seconds.

[a] submitted as bug reports to the PARI/GP team.

Example (continued): discrete log in $\mathrm{Cl}_{\mathfrak{f}}(F)$

Input: An ideal $(\alpha)\mathfrak{a}$, where $(\alpha\mathfrak{a},\mathfrak{f})=1$. We are given

$$\mathrm{Cl}(F)=\oplus(\mathbb{Z}/d_i\mathbb{Z})g_i, \quad \text{and} \quad \mathrm{Cl}_{\mathfrak{f}}(F)=\oplus(\mathbb{Z}/D_j\mathbb{Z})G_j,$$

as well as elements $\gamma_i\in\mathbb{Z}[F^*]$ such that $(\gamma_i g_i,\mathfrak{f})=1$.

Output: (f_j) and $\beta\in\mathbb{Z}[F^*]$, $\beta=1$ mod$^*\mathfrak{f}$, such that $\alpha\mathfrak{a}=\beta\prod G_j^{f_j}$.

(1) *[Work in* $\mathrm{Cl}(F)$*]*. Write $\mathfrak{a}=\tau\prod g_i^{e_i}=(\tau\prod\gamma_i^{-e_i})\prod(\gamma_i g_i)^{e_i}$, $\quad\tau\in\mathbb{Z}[F^*]$.

(2) *[Work in* $(\mathcal{O}_F/\mathfrak{f})^*$*]*. For each $\wp^{n_\wp}\,||\,\mathfrak{f}$, map $\alpha\tau\prod\gamma_i^{-e_i}$ to $(\mathcal{O}_F/\wp^{n_\wp})^*$, first replacing all individual components by their $\wp$-coprime parts.
This works because the evaluated product is coprime to $\mathfrak{f}$, hence to $\wp$.
(Signatures are also easy to compute.)

(3) Glue above results to get discrete log in $\mathrm{Cl}_{\mathfrak{f}}(F)$ as usual.

Remarks: 1) the $cp_\wp(p)$ and $cp_\wp(\gamma_i)$ play a special role. 2) $\mathfrak{f}$ is liable to change.

One last example

Question: compute the Hilbert class field H_F of $F = \mathbb{Q}(\sqrt{181433})$, a degree 5 extension.

Method: adjoin 5-th roots of unity and use Kummer theory: find $\alpha \in F(\zeta_5)$ such that $\alpha^{1/5}$ generates $H_F(\zeta_5)$ over $F(\zeta_5)$, then use Lagrange resolvents to obtain H_F. Easy?

In practice, $\mathrm{Cl}(F(\zeta_5))$ is isomorphic to $\mathbb{Z}/(3620) \times \mathbb{Z}/(20)$ so we raise elements to huge powers. Initial PARI implantation overflowed physical memory after 2 days of computation, dealing with elements of *logarithmic* height 20000 at 10^5 decimal digits of accuracy.

Answer: $X^5 - X^4 - 77X^3 - 71X^2 + 360X - 169$ found in 2 minutes, mostly spent applying a polynomial reduction algorithm to the initial answer. The bottleneck computations above are done in 15 seconds. They deal with the same elements, in a less wasteful form.

Reference

Topics in computational algebraic number theory

Karim.Belabas@math.u-psud.fr

Preprint available at

- `http://www.math.u-psud.fr/~belabas/pub/#modf`