Faster class group computations using norm relations

A. Page
joint work with J.-F. Biasse, C. Fieker and T. Hofmann

LMB, Besançon
Inria / Université de Bordeaux

14/01/2022

Computing class groups

Goal : given a number field K, compute $\mathrm{Cl}(K)$.

Reminder : Buchmann's algorithm.

- Choose S set of primes generating $\mathrm{Cl}(K)$ (GRH).
- Find S-units $R \subset \mathbb{Z}_{K,S}^{\times}$.
- Compute $C = \mathbb{Z}^S / \langle R \rangle$ and $U = \ker(\langle R \rangle \to \mathbb{Z}^S)$.
- Check if $\langle R \rangle = \mathbb{Z}_{K,S}^{\times}$ using class number formula.
- Output C.

Using automorphisms

Question : assume K has a nontrivial group G of automorphisms. Can we use this to compute $\mathrm{Cl}(K)$ faster ?

- Use action of G to get extra relations for free.
- Use structure of module over the group ring for faster linear algebra ?
- By Galois theory, K has many subfields...

Norm relations

For $H \leq G$, define the *norm element*

$$N_H = \sum_{h \in H} h \in \mathbb{Z}[G].$$

Wada, Bauch–Bernstein–de Valence–Lange–van Vredendaal, Biasse–van Vredendaal : $G = C_2 \times C_2 = \langle \sigma, \tau \rangle$.

$$2 = N_{\langle \sigma \rangle} + N_{\langle \sigma \rangle} - \sigma N_{\langle \sigma\tau \rangle}.$$

Parry, Lesavourey–Plantard–Susilo : $G = C_3 \times C_3 = \langle u, v \rangle$.

$$3 = N_{\langle u \rangle} + N_{\langle v \rangle} + N_{\langle uv \rangle} - (u + uv)N_{\langle u^2 v \rangle}.$$

Norm relations

Definition : *norm relation* with *denominator d*

$$d = \sum_i a_i N_{H_i} b_i$$

with $a_i, b_i \in \mathbb{Z}[G]$ and $d \in \mathbb{Z}_{>0}$.

For all $x \in K^\times$, we have

$$x^d = \prod_i \left(N_{K/K^{H_i}}(x^{b_i})\right)^{a_i},$$

so x^d belongs to the subgroup generated by the subfields.

The S-units from the subfields generate a $\mathbb{Z}[G]$-submodule of finite index in the S-units of K.

Existence of norm relations

When do such relations exist ?

Theorem (BFHP, Wolf)

A finite group G admits a norm relation if and only if G contains

- *a non-cyclic subgroup of order pq (p,q, primes not necessarily distinct), or*
- *a subgroup isomorphic to* $\mathrm{SL}_2(\mathbb{F}_p)$ *where* $p = 2^{2^k} + 1$ *is a Fermat prime with* $k > 1$*.*

Also : criterion to test existence with specific subgroups, more precise information in the abelian case.

Saturation

Problem : from $R \subset K^\times$, compute $R' = \{x \in K^\times \text{ s.t. } x^d \in R\}$.

Saturation algorithm (Pohst–Zassenhaus, rediscovered many times) :

- Use reduction modulo primes to detect powers.
- Compute roots.
- Terminate or add more primes.

BFHP : under GRH, polynomial bound on the set of primes required.

Denominators of norm relations

Can we control the denominator d?

Theorem (BFHP)

If G admits a norm relation using certain subgroups, then it also admits one with d dividing $|G|^3$ and using the same subgroups.

Proof sketch : There is a representation-theoretic interpretation of existence of a norm relation. Rewrite it in terms of idempotents, and estimate the denominators of the idempotents.

Reduction to the subfields

Theorem (BFHP)

Assume GRH. Let G admitting a norm relation. The computation of the group of S-units reduces in deterministic polynomial time from any K with an action of G to the corresponding subfields.

Implementations

- Implementation in Julia (Nemo/Hecke) : general case.
- Implementation in gp : requires K to be Galois over $\mathbb{Q}$, only uses relations coming from abelian subgroups, only computes the class group, possible infinite loop, but faster.
- Implementation in libpari : general case, TODO !

Examples : cyclotomic fields

Degree 72, 5 seconds :

```
? abbnf = abelianbnfinit(polcyclo(216));
? getcyc(abbnf)
% = [1714617]
```

Degree 144, 15 seconds :

```
? abbnf = abelianbnfinit(polcyclo(504));
? getcyc(abbnf)
% = [39312, 13104, 252, 252, 252, 126, 2, 2, 2]
```

Degree 288, 3 minutes :

```
? abbnf = abelianbnfinit(polcyclo(1260));
? getcyc(abbnf)
% = [30253421167033428O, 8464152747960, ...]
```

Examples : multiquadratic fields

Degree 16 :

```
? pol = multiquad([-1,2,3,5]);
? abbnf = abelianbnfinit(pol);
cpu time = 1,258 ms.
? getcyc(abbnf)
% = [2]
? bnf = bnfinit(pol);
cpu time = 66 ms.
? bnf.cyc
% = [2]
```

Examples : multiquadratic fields

Degree 32 :

```
? pol = multiquad([-1,2,3,5,7]);
? abbnf = abelianbnfinit(pol);
cpu time = 3,185 ms.
? getcyc(abbnf)
% = [8, 4, 4, 2]
? bnf = bnfinit(pol);
cpu time = 8,271 ms.
? bnf.cyc
% = [8, 4, 4, 2]
```

Examples : multiquadratic fields

Degree 64 :

```
? pol = multiquad([-1,2,3,5,7,11]);
? abbnf = abelianbnfinit(pol);
cpu time = 1min, 4,345 ms.
? getcyc(abbnf)
% = [96, 48, 16, 16, 16, 8, 8, 4, 4, 2, ...]
? \\bnf = bnfinit(pol); \\very long...
```

Examples : Kummer fields

$\mathbb{Q}(\zeta_n, a_1^{1/n}, \ldots, a_k^{1/n})$:

```
? pol = kummer(3,[2,3,5]);
? abbnf = abelianbnfinit(pol);
cpu time = 4,549 ms.
? getcyc(abbnf)
% = [6, 6, 3]
```

Examples : other Galois fields

Degree 81, not abelian :

```
? pol = galoisgetpol(3^4,3)[1];
? abbnf = abelianbnfinit(pol);
cpu time = 17,142 ms.
? getcyc(abbnf)
% = []
? pol = galoisgetpol(3^4,7)[1];
? abbnf = abelianbnfinit(pol);
cpu time = 16,109 ms.
? getcyc(abbnf)
% = []
```

Examples : a larger cyclotomic field

Degree 1728, 4h :

```
? pol = polcyclo(6552);
? abbnf = abelianbnfinit(pol);
```

Questions ?

Merci !