Problems and approaches in class groups of large degree fields

Claus Fieker

September 30, 2015

A number field

$$K = \mathbb{Q}[t]/f$$

is generated by some (monic), (integral), irreducible polynomial f of degree $\deg f = K : \mathbb{Q} = n$.
In K we have a nice, canonical ring $\mathbb{Z}_K$ the ring of integers.
$\mathbb{Z}_K$ is a Dedekind domain, hence we have a unique factorisation into prime *ideals*.
Ideals $\mathfrak{a}$ and $\mathfrak{b}$ are equivalent iff

$$\mathfrak{a} = \gamma\mathfrak{b}$$

for some $\gamma \in K^*$.
The *class group* Cl_K is the group generated by this equivalence relation.

Facts about the class group

- Cl_K is a *finite abelian* group
- Cl_K is generated by prime ideals bounded by some explicit integer B
- Cl_K is one of the most important fundamental invariants
- Cl_K can be used for crypto
- Cl_K is a traditional challenge

Algorithmically, computation of the class group follows roughly the same lines as the factorisation (NSF) or the disc log:

- find a factor base B of prime ideals
- find elements α that are B-smooth (relations)
- collect the factorisations in a (large) (sparse) *relation matrix* M
- use linear algebra to derive the result

What's the big deal?

- the field is fixed by the user and can be of large degree (> 100)
- the linear algebra is in $\mathbb{Z}$

Fact

No-one is interested in generic, random large degree fields.

$$\Rightarrow$$

Expect large degree fields to behave non-generic and special.

What's the big deal?

- the field is fixed by the user and can be of large degree (> 100)
- the linear algebra is in $\mathbb{Z}$

Fact

No-one is interested in generic, random large degree fields.

$\Rightarrow$

Expect large degree fields to behave non-generic and special.

What's the big deal?

- the field is fixed by the user and can be of large degree (> 100)
- the linear algebra is in $\mathbb{Z}$

Fact

No-one is interested in generic, random large degree fields.

$\Rightarrow$

Expect large degree fields to behave non-generic and special.

What's the big deal?

- the field is fixed by the user and can be of large degree (> 100)
- the linear algebra is in $\mathbb{Z}$

Fact

No-one is interested in generic, random large degree fields.

$\Rightarrow$

Expect large degree fields to behave non-generic and special.

What's the big deal?

- the field is fixed by the user and can be of large degree (> 100)
- the linear algebra is in $\mathbb{Z}$

Fact

No-one is interested in generic, random large degree fields.

$$\Rightarrow$$

Expect large degree fields to behave non-generic and special.

The Field

Large fields of interest to number theory (and current crypto-attacks) are non-generic!
For example, ideal lattices are frequently in $\mathbb{Q}(\zeta_{2^n})$, class field theory uses $K(\zeta_n)$ for an arbitrary field K and some (smallish) n.
Standard trickery reduces (most) problems in $\mathbb{Q}(\zeta_n)$ to the maximal real subfield of half the degree: $K = \mathbb{Q}(\zeta_n + 1/\zeta_n)$.
Lets fix this K for $n = 2^9$, so K is of degree 128.

Relations

As usual, the idea is to find (small) elements, hoping that small elements will have small norm and small norm elements are more likely to be smooth.
Approaches

- enumeration of short elements
- enumeration of short elements in low-dimensional sub-modules
- (sieving)
- LLL-basis of "random" ideals
- linear cominations of LLL-bases

Keep in mind that the (full) lattices are of rank 128. Thus even plain LLL is non-trivial in runtime.

LLL-basis

Start with a LLL-basis for $\mathbb{Z}_K = \mathbb{Z}[\zeta_n + 1/\zeta_n]$.
Experimentally, up to ordering, this basis is

$$b_1 = 1, \quad b_i = \zeta_n^i + 1/\zeta_n^i$$

Thinking in $\mathbb{C}$ we see that $\|b_i\|^2 < 4d = 2^9 = 512$. Forming the product

$$\prod \|b_i\|^2 < (2^9)^{127} \cdot 128 = 2^{1150}$$

The discriminant is 2^{1000}, computing slighly more carefully, we see this is a really good basis, *much* better than LLL in dimension 128 would allow.
So lucky break.

Conventional wisdon says that generically, we will be able to find relations of norm $N \leq 2^{500} = \sqrt{D}$ in more general lattices of this field, but

$$|N(b_i)^2| \leq (\frac{\|b_i\|}{128})^{128} = 4^{128}2^{256}$$

is much smaller!
Similar, taking sums of l basis elements we get a naive norm estimate of $(2l)^{128}$, so we *can* find many really small elements that should be nice and smooth.
This is good?

Experiment 1

The Bach-bound (GRH bound for generation of the class group) says we should involve prime ideals of norm up to $5.4 \cdot 10^6$. This would be more than 100,000 primes - a lot.
But, we have so many really small elements, lets go for $2 \cdot 10^5$, only 18,000 primes.
Generating relations using random $\{-1, 1\}$ combinations of up to 5 basis elements we find lots of relations easily.
Echelon of the relation matrix M with 30,000 rows and 18,000 columns we get a row rank of 1,000 or so. The yield is very good, 1 in 200, but they are "all" redundant.

(I let it run for about 80,000 relations, the rank went up to 2,000 or so)
What's wrong?
Looking at the echelonised matrix carefully, all prime ideals occur in pairs. All relations involve pairs of prime ideals that are Galois-conjugate.
In other words, all relations come from the index 2 subfield of degree 64.
Next attempt: use Galois to only allow relations *not* in the subfield.
Result: the "same", ie minimally better - the relations are of the form ϵa for ϵ a unit of the large field and a a relation of the small field. So no success.

Experiment 2

Same setup as before, but accept only relations that are progressively larger and larger wrt norm.
Result:

- the yield is down (but still *very* good)
- the rank is growing faster than before
- but still, in the echelon form, all relations are in the subfield.

Experiment 3

Same setup as before, but choose relations from a LLL-basis of one of the totally split primes, thus all relations found this way will have a 1 at that column.
Result: all relations are still also divisible by a conjugate prime

Lets look at this the other way round:

- the largest factor base that we might possibly consider has 10^6 elements, which means a bound of approx. $10^7 = 2^{22}$ which is good.
- using Dickman-ρ to estimate smoothness probability we see that for traditional relations of norm 2^{500} we expect a yield of 1 in $4 \cdot 10^{31}$
- to get a yield of 1 in 200,000 we need to get norms $\leq 2^{150}$

Experiment 4

Using $\{-1, 1\}$ combinations we just check how large the norms are, depending on the number of terms.
Observation:

- the norms grow with the number of terms
- but much slower then expected
- the norms look roughly normally distributed

So taking up to 10 terms with coefficents in $\{\pm 1\}$ we have a 50% chance of finding elements of norm $\leq 2^{150}$
Of course, a fair number are much smaller. From the norm distribution and the Dickman-ρ we estimate the number of relations.
A subsequent test confirms this.

Small norm elements

Steve Donnelly suggested to model small norm elemwents.
Idea: the distribution should follow that of hyperbolic volume elements

$$\mathrm{vol}(\prod x_i \leq b \cap [0,1]^n)$$

So, sampling uniformely randon elements in $[0,1]^n$, we should expect $N(x) = \prod x_i$ according to this volume.
This works quite well and gives a distribution quite close to the one observed. Hence, we can use this to predict yield.

Experiment 5

Setup as above, using 10 terms and $\{\pm 1\}$ coefficients. Checking a few million elements, we find approx. 400 relations.
Next, we throw away relations from the subfield by checking that the part of the norm coming from the unramified degree 1 primes is square-free.
Next, we supplement those relations using the automorphisms.
Result: 12,000 relations and a rank of 4,000. *But* the conjugate prime ideals are separated!
Hence, it is plausible to obtain the full class group in a couple of days.

Relations search

Let $\mathfrak{a}$ be an integral ideal of norm $N(\mathfrak{a})$.
Using LLL, find a basis b_i s.th. $|N(b_i)/N(\mathfrak{a})$ is bounded (by $\sqrt{D}$, roughly)
Now:

- use (random) linear coombinations
- lattice enumeration (in sublattices)
- "sieving" in 1 or 2-dim sublattices
- "sieving" in larger sublattices

Note

Sieving polynomial would have a large degree.

Note

We can change the metric on $\mathfrak{a}$ as well

Relations search

Let $\mathfrak{a}$ be an integral ideal of norm $N(\mathfrak{a})$.
Using LLL, find a basis b_i s.th. $|N(b_i)/N(\mathfrak{a})$ is bounded (by $\sqrt{D}$, roughly)
Now:

- use (random) linear coombinations
- lattice enumeration (in sublattices)
- "sieving" in 1 or 2-dim sublattices
- "sieving" in larger sublattices

Note

Sieving polynomial would have a large degree.

Note

We can change the metric on $\mathfrak{a}$ as well

Relations search

Let $\mathfrak{a}$ be an integral ideal of norm $N(\mathfrak{a})$.
Using LLL, find a basis b_i s.th. $|N(b_i)/N(\mathfrak{a})$ is bounded (by $\sqrt{D}$, roughly)
Now:

- use (random) linear coombinations
- lattice enumeration (in sublattices)
- "sieving" in 1 or 2-dim sublattices
- "sieving" in larger sublattices

Note

Sieving polynomial would have a large degree.

Note

We can change the metric on $\mathfrak{a}$ as well

We implemented possibilites 1 and 2 so far.
As we saw: problem is not finding relations, but finding independent ones.
Observation: if $\mathfrak{a}$ is large enough, or if the metric on $\mathfrak{a}$ is suitably distorted then

- The relations become more "generic", we loose the very small and very smooth ones
- The yield drops sharply due to the increased norms

Special q-descent

So far, we have not tried to implement the q-descent procedure. In the examples are interested in, finding smooth elements is trivial.
This might change.

Special q-descent

So far, we have not tried to implement the q-descent procedure. In the examples are interested in, finding smooth elements is trivial. This might change.

Large Primes

We also implemented the large prime variant.

Idea: if $N(\alpha) = p \cdot r$ where r is smooth and p is a prime (in $\mathbb{Z}$) that is too large, then

$$\mathfrak{p} = \langle p, \alpha \rangle$$

is a prime ideal (of degree 1).

If p is generic, ie. coprime to the index, then

$$\gcd_{\mathbb{F}_p}(f, \alpha)$$

is canonical and uniquely identifies the prime ideal $\mathfrak{p}$. Hence we can use this for hashing.

Note

We can also use automorphisms to match partial relations involving the same prime number.

Depending on the parameters this can help.

Linear Algebra

- info about rank(growth)
- info about missing pivonts
- position of non-trivial pivots
- product of elementary divisors
- Smith form
- (some) kernel elements
- link to Smith-form

The first two can be done (heuristically) modulo (small) primes.
So far, all questions are applied to dynamically growing matrices.
The expected product of the elementary divisors, the class number, is small. The bound is not.
The kernel elements yield units. They are used for verification.

Relations processing

- needs norms of a large number of elements (of small bounded norm)
- need smoothness test of large number of integers
- need smoothness test of large number of elements

Norms

Have: elements with a bounded norm and possibly a large known factor (if elements are in a given ideal).
The missing factor is mostly uniformely bounded by $O(\sqrt{D})$
Approaches:

- determinant of matrix: $O(n^3)$ or $O(n^\omega)$
- product of conjugates: $O(n^2) + O(n)$
- resultant with defining polynomial: $O(n \log n)$

Counting operations only. All approaches can utilise the known factor and the bound (precision or CRT number of primes)

Integral Smoothness

Use a pre-computed product tree over all prime numbers involved in the factor base.
Individual tests perform $\gcd$ and division operations.

Note

In general norms cannot be square-free.

Note

For normal fields (e.g. cyclotomic ones), the integral test is sufficient to identify relations. In general this is only neccessary.

Note

The same tree is also used to split the norm over the $\mathbb{Z}$-factor base

Algebraic Smoothness

Given an algebraic nummber α, a prime number p and a list of prime ideals $\mathfrak{p}_i$ containing p.
Want: $\langle \alpha, p^\infty \rangle = \mathfrak{b} \prod \mathfrak{p}_i^{l_i}$ for some implicit, not computed $\mathfrak{b}$ coprime to p.
Classsically: compute $v_{\mathfrak{p}_i}(\alpha)$ for all $\mathfrak{p}_i$ and check result.
Problem: if p is totally split, there are a lot of $\mathfrak{p}_i$ to test.

Algebraic Smoothness

2 step: first, deal with $\langle p, \alpha \rangle$ only. This will identify the $\mathfrak{p}_i$ involved (and mostly settle everything)
2nd: compute the valuations.
Note: generically, $\mathfrak{p}_i = \langle p, g(\beta) \rangle$ for g an irreducible divisor of the defining polynomial modulo p.
Thus use a product tree over the divisors modulo p.

Correctness

If relation matrix has full rank (and unit group has full rank), algebraic saturation will guarantee completness of relation latttice, hence guarantee a sub-group of the class group.
For everything else, the factor base must be large enough according to whatever bound (or additional relations for the gap need to be found)
If the factor base is large enough, then saturation can be replaced by the Euler product (esp. if the units are computed as well)

DiscLog/ PID-test

Tests on factor base elements are "just" projection down into the group structure, straightforward.
For ideals $\mathfrak{a}$ outside, we need an element in $\mathfrak{a}$ sth

$$\mathfrak{a} = \alpha \prod \mathfrak{p}_i^{l_i}$$

where the product runs over the factor base.
If the factor base is too small, we might not find such an element.
If we find one for $\mathfrak{a}^l$ instead, we can use saturation for correct results.

Looking back, what are the challenges?

- the relations are too dependent
- it is difficult to really separate the search phase from the linear algebra phase
- clearly, my understanding of the behaviour of the relations search is wrong - or all the accepted heuristics do *not* hold for large *special* fields.
- we need linear algebra over $\mathbb{Z}$ for enormous sparse matrices
- we need to carefully select relations, the easy ones are too good to be true.
- the huge field degree suddenly shows in unexpected sub-algorihthms

Conclusion

Relation search for class groups is difficult - for the wrong reasons.
We can easily find many small elements that are very smooth.
However, they do not generate the full group.
Using relations of the classically predicted size, the smoothness probability is way too small.
We need larger factor bases than (our) current linear algebra can handle.