Rigorous arithmetic in the Arakelov divisor class group of a number field

Felix Fontein

PIMS / University of Calgary

(joint work with Michael J. Jacobson, Jr.)

What is the Arakelov divisor class group?

The **Arakelov divisor class group** $\mathrm{Pic}^0(K)$ is an analogue of the divisor class group of a function field or projective curve for number fields K. It is an extension of the **ideal class group** $\mathrm{Cl}(\mathcal{O}_K)$ of $\mathcal{O}_K$, the ring of integers of K, with an n-dimensional torus $\mathbb{R}^n/\Lambda$, where Λ is the **unit lattice**:

- n is the **unit rank** of K, i.e. the rank of $\mathcal{O}_K^*$;
- Λ is a homomorphic image of the free part of $\mathcal{O}_K^*$.

In fact, the theory presented here also works for function fields with finite constant field, with the difference that no numerical approximation is needed. The f**-representations** (see the next column) correspond to **reduced divisors** in the function field setting in the sense of F. Heß.

In the number field case, to our knowledge, explicit arithmetic in $\mathrm{Pic}^0(K)$ was so far only available in real quadratic number fields due to work of Hühnlein and Paulus [HP01], and Jacobson, Scheidler and Williams [JSW01]. Our work **generalizes** this to all number fields. Our method is also **rigorous** in the sense that:

- we have explicit bounds on the required precision for the approximations which ensure that the results are correct up to an arbitrarily small approximation error;
- we have explicit bounds on that approximation error.

The advantage over R. Schoof's treatment of $\mathrm{Pic}^0(K)$ in [Sch08] is that we have a unique representation of elements in $\mathrm{Pic}^0(K)$.

f-representations

Let $\sigma_0, \ldots, \sigma_n : K \to \mathbb{C}$ be the embeddings corresponding to the $n+1$ distinct absolute values $|f|_i := |\sigma_i(f)|$. For the sake of simplicity, let us assume that $\sigma_0(K) \subseteq \mathbb{R}$.

An f**-representation** is a pair $(\mathfrak{a}, (f_1, \ldots, f_n))$, where $\mathfrak{a}$ is a non-zero fractional ideal of $\mathcal{O}_K$ and $f_1, \ldots, f_n \in \mathbb{R}$ are positive numbers, such that

$$\{g \in \mathfrak{a} \mid |g|_0 \leq 1,\ \forall i \geq 1 : |g|_i \leq f_i\} = \{-1, 0, 1\}.$$

Let $f\text{-Rep}(K)$ be the set of all f-representations. One obtains [Fon09]:

Theorem. There exists an **effectively computable bijection**

$$\Psi : f\text{-Rep}(K) \longrightarrow \mathrm{Pic}^0(K).$$

If one defines $A + B := \Psi^{-1}(\Psi(A) + \Psi(B))$ for $A, B \in f\text{-Rep}(K)$, the resulting group operation on $f\text{-Rep}(K)$ can be effectively computed without using Ψ.

Obviously, approximations have to be used for the f_i's. We chose a floating point representation which allows to keep track of the approximation error.

An important property of f-representation is that **they are small**:

- the ideal $\mathfrak{a}$ can be stored using $O((\log|\Delta_K|)^3)$ bits;
- and $1 \leq f_i \leq \sqrt{\Delta_K}$, where Δ_K is the discriminant of K.

Numerical approximation

To compute sums of f-representations or inverses, we need to compute short lattice elements in a lattice whose coordinates are algebraic numbers – these lattices are ideal lattices, considered in the $d = [K : \mathbb{Q}]$-dimensional Minkowski space $K \otimes_{\mathbb{Q}} \mathbb{R}$. Hence, we need to **approximate** a lattice basis $v_1, \ldots, v_d \in \mathbb{R}^d$ and work with the approximated integral lattice

$$\mathbb{Z}\,\mathrm{Round}(2^q v_1) + \cdots + \mathbb{Z}\,\mathrm{Round}(2^q v_d) \subseteq \mathbb{Z}^d.$$

Buchmann and Williams gave asymptotic results on the required precision [BW87, Buc87]: assuming that the field degree is fixed, the required precision is in $O(\log|\Delta_K|)$. One can work these out in detail to obtain the following lower bounds for q:

$[K:\mathbb{Q}]$	K totally real or Galois	general case
2	$10.501 + 5.7708\log\lvert\Delta_K\rvert$	$10.501 + 5.7708\log\lvert\Delta_K\rvert$
3	$17.510 + 9.3776\log\lvert\Delta_K\rvert$	$22.010 + 18.034\log\lvert\Delta_K\rvert$
4	$25.000 + 12.985\log\lvert\Delta_K\rvert$	$37.000 + 36.068\log\lvert\Delta_K\rvert$
5	$32.898 + 16.591\log\lvert\Delta_K\rvert$	$55.398 + 59.872\log\lvert\Delta_K\rvert$
6	$41.143 + 20.198\log\lvert\Delta_K\rvert$	$77.143 + 89.448\log\lvert\Delta_K\rvert$
7	$49.689 + 23.805\log\lvert\Delta_K\rvert$	$102.19 + 124.80\log\lvert\Delta_K\rvert$
8	$58.500 + 27.412\log\lvert\Delta_K\rvert$	$130.50 + 165.91\log\lvert\Delta_K\rvert$
10	$76.812 + 34.625\log\lvert\Delta_K\rvert$	$196.82 + 265.46\log\lvert\Delta_K\rvert$
12	$95.907 + 41.839\log\lvert\Delta_K\rvert$	$275.91 + 388.09\log\lvert\Delta_K\rvert$
14	$115.67 + 49.052\log\lvert\Delta_K\rvert$	$367.67 + 533.80\log\lvert\Delta_K\rvert$
16	$136.00 + 56.266\log\lvert\Delta_K\rvert$	$472.00 + 702.60\log\lvert\Delta_K\rvert$
18	$156.85 + 63.479\log\lvert\Delta_K\rvert$	$588.85 + 894.48\log\lvert\Delta_K\rvert$
20	$178.15 + 70.693\log\lvert\Delta_K\rvert$	$718.15 + 1109.5\log\lvert\Delta_K\rvert$

When using these bounds, everything is **rigorous**: if one uses this precision to do arithmetic in $\mathrm{Pic}^0(K)$ using (f,p)-representations, then the results are always correct, up to a small approximation error which can be explicitly bounded.

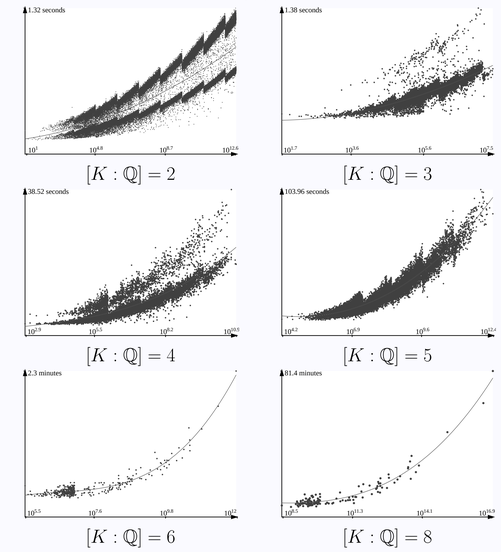
In practice, the required precisions are **huge**. One could work with far less many bits, and often obtain the same results. Our implementation uses the precision given by these bounds to be on the safe side.

Our constants in the bounds presented above are already better than the bounds one obtains when tracing the constants through the analysis of Buchmann and Williams. We hope to be able to make them smaller, giving faster computation and making higher field degrees feasible.

Implementation and timing

We have an implementation of effective arithmetic in $\mathrm{Pic}^0(K)$ using (f,p)-representations and of computation of $\pi(v)$ for most number fields, which uses high precision arithmetic to avoid potential errors due to bad approximation. Unfortunately, this turns out to be slow – which is at least partially the fault of our implementation.

To enumerate short lattice vectors, we use the Schnorr-Euchner enumeration technique. The running time is polynomial in the size of the coefficients and exponential in the dimension, which is the field degree $[K : \mathbb{Q}]$. Our implementation currently uses multi-precision floating point numbers for the Gram-Schmidt coefficients. We tested our implementation with a large amount of totally real number fields up to degree $[K : \mathbb{Q}] = 8$ so far. The following graphs show the time required to compute $\pi(v)$ for 10 random choices of v with coordinates of absolute value up to 2^{64}, including the initializations of the addition chains. This is all done by performing a sequence of additions in $\mathrm{Pic}^0(K)$ using our arithmetic.



The $\mathbf{x}$**-axes** show $\log|\Delta_K|$, while the $\mathbf{y}$**-axes** show the time needed. The curve is a cubic fit function in $\log|\Delta_K|$.

The clear **bottleneck** of our method is the enumeration of short lattice vectors, in combination with the huge integral lattice this is applied to.

Motivating applications

The main tool are (f,p)-representations, which are f-representations with floating point approximations of the involved real numbers. For details on f-representations, see the next column.

(1) (f,p)-representations allow to do **effective approximate computations** in the Arakelov divisor class group $\mathrm{Pic}^0(K)$:
- when keeping track of floating point approximation errors, one always knows how far the current approximated element is at most off the element itself;
- moreover, the errors are easy to control, i.e. if one wants the result with a certain precision, one can quickly compute the precision one has to start with.

(2) One can use f-representations to efficiently **test whether** $\mathbf{v} \in \mathbf{\Lambda}$ for a vector $v \in \mathbb{R}^n$, i.e. if v is the logarithm vector of a unit: this is the case if, and only if, $\pi(v) = (\mathcal{O}_K, (0, \ldots, 0))$. (For the definition of π, see the box "*f-representations and the torus $\mathbb{R}^n/\Lambda$*".)

(3) In Buchmann's baby-step giant-step algorithm for computing fundamental units of number fields [Buc87], one needs to quickly **find reduced ideals near to given positions**, which can be computed by taking the ideal part of $\pi(v)$. These are the "giant steps".

f-representations and the torus $\mathbb{R}^n/\Lambda$

The connected component of the neutral element of $\mathrm{Pic}^0(K)$ is isomorphic to $\mathbb{R}^n/\Lambda$. Under the bijection Ψ^{-1} from above, it maps to $f\text{-Rep}(\mathcal{O}_K)$, defined as

$$\left\{(\mathfrak{a}, (f_i)_i) \in f\text{-Rep}(K) \;\middle|\; \mathfrak{a} = \tfrac{1}{\mu}\mathcal{O}_K \text{ for some } \mu \in K^*\right\},$$

and the isomorphism from $f\text{-Rep}(\mathcal{O}_K)$ to $\mathbb{R}^n/\Lambda$ is given by

$$d : \left(\tfrac{1}{\mu}\mathcal{O}_K, (f_i)_i\right) \longmapsto \left(\log f_i - \log|\mu|_i\right)_i + \Lambda.$$

Computing this map is essentially solving the **discrete logarithm problem** in $\mathrm{Pic}^0(K)$.

On the other hand, the map $\pi : \mathbb{R}^n \to f\text{-Rep}(\mathcal{O}_K)$ given by the projection $\mathbb{R}^n \to \mathbb{R}^n/\Lambda$ followed by the inverse of the above isomorphism is **effectively computable**:

- One can quickly compute f-representations representing $\pi(v)$ for $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$ with small $|v_i|$.
- Then, one can use an add-and-double technique to rapidly compute $\pi(v)$ in $O(\sum_{i=1}^n \log|v_i|)$ additions of f-representations.

References

[Buc87] J. Buchmann. Zur Komplexität der Berechnung von Einheiten und Klassenzahl algebraischer Zahlkörper. Habilitationsschrift, October 1987.

[BW87] J. Buchmann and H. C. Williams. On principal ideal testing in algebraic number fields. *J. Symbolic Comput.*, 4(1):11–19, 1987.

[Fon09] F. Fontein. The infrastructure of a global field of arbitrary unit rank, 2009. Preprint available at `http://arxiv.org/abs/0809.1685`. Accepted pending revisions at Math. Comp.

[HP01] D. Hühnlein and S. Paulus. On the implementation of cryptosystems based on real quadratic number fields. In *Selected areas in cryptography (Waterloo, ON, 2000)*, pages 288–302. Springer, Berlin, 2001.

[JSW01] M. J. Jacobson, Jr., R. Scheidler, and H. C. Williams. The efficiency and security of a real quadratic field based key exchange protocol. In *Public-key cryptography and computational number theory (Warsaw, 2000)*, pages 89–112. de Gruyter, Berlin, 2001.

[Sch08] R. Schoof. *Computing Arakelov class groups*, volume 44 of *MSRI Publications*, pages 447–495. Cambridge University Press, Cambridge, 2008.

Future (planned) improvements

(1) Use the rigorous floating point Schnorr-Euchner enumeration method by Pujol and Stehlé.

(2) Find smaller bounds on the minimal precision.

(3) Find better starting values for reductions, to minimize the number of short lattice vector enumerations as well as the time needed for them.

Ninth Algorithmic Number Theory Symposium ANTS-VIII, July 19–23, 2010, INRIA, Nancy (France)