
Applications of the Class Field Theory of Global Fields

Claus Fieker *

School of Mathematics and Statistics, University of Sydney, NSW 2006, Australia
`claus@maths.usyd.edu.au`

Summary. Class field theory of global fields provides a description of finite abelian extensions of number fields and of function fields of transcendence degree 1 over finite fields. After a brief review of the handling of both function and number fields in **Magma**, we give a introduction to computational class field theory focusing on applications: We show how to construct tables of small degree extensions and how to utilize the class field theory to find curves with many rational points.

1 Introduction

Class field theory is concerned with abelian extensions of (number) fields, that is finite Galois extensions K of $\mathbb{Q}$ such that its group of automorphisms is abelian. After establishing the theory for number fields, it was realized that almost all of it could be extended to certain infinite extensions of finite fields. Thus the concept of global fields was born as the class of fields for which “the theory works”. In this chapter we will develop the class field theory for both function fields and number fields in **Magma** [BCP97], [CB03, Chapters 50, 53, 54, 56, 57] and give some applications to number theory and coding theory.

We begin by briefly describing **Magma**’s interface to number fields starting with their representation. Next we explain how class groups are represented and computed with a particular focus on the practical aspects of the computation. This is followed by a discussion of ray class groups which parametrize class fields. The fundamental statements of class field theory that are needed for the applications presented here are recalled, again, with a strong emphasis on **Magma**’s view of the material.

The program is then repeated for global function fields: we recall the definitions and show how they are implemented and used in **Magma**. The presentation for function fields focuses mainly on the differences between number fields

*This research was supported in part by a grant from the Australian Research Council (A00104694)

and function fields as they essentially behave in the same way. We introduce divisors, divisor class groups and divisor ray class groups.

As applications of the class field theory developed, we show how to compile certain tables of global fields indexed by degree, discriminant and Galois group. For function fields, we show how the theory may be used to find fields with “many rational places” which are of interest in certain applications such as in the construction of “good codes”.

2 Number Fields

We briefly recall the definition of number fields and summarize their properties with a strong emphasis on **Magma**’s view of them.

2.1 Basic Properties

A number field K is a finite extension of $\mathbb{Q}$. By the primitive element theorem, there is some $\alpha \in K$ such that $K = \mathbb{Q}(\alpha)$ holds. Since K is a finite extension, α is algebraic over $\mathbb{Q}$, so α is a root of an irreducible polynomial $f(t) \in \mathbb{Q}[t]$ and as a field, we have an isomorphism

$$K \rightarrow \mathbb{Q}[t]/\langle f \rangle : \alpha \mapsto t.$$

To create a number field in **Magma**, we reverse this process: we first define an irreducible polynomial $f(t)$ and then use it to create K and α :

```
> Q := Rationals();
> Qt<t> := PolynomialRing(Q);
> f := t^3-25;
> K<a> := NumberField(f);
> a^3;
25
```

Now we have a number field at our disposal. Note that K is *not* considered to be a subfield of the field of complex numbers $\mathbb{C}$! This is due to the fact that a number field K of degree $n := [K : \mathbb{Q}]$ has n different embeddings into $\mathbb{C}$ and there is no canonical embedding.

The most important subring of a number field is its ring of integers $\mathbb{Z}_K$, ie. the elements $\beta \in K$ that are roots of monic polynomials $g \in \mathbb{Z}[t]$. It is well known that $\mathbb{Z}_K$ is a free $\mathbb{Z}$ -module of rank $n := [K : \mathbb{Q}]$ so we can represent elements with respect to a fixed basis of $\mathbb{Z}_K$. A so-called *integral basis* can be computed, for example, using either the Round2 or the Round4 method [Coh93, PZ89, FL94, Pau01].

By allowing an integral denominator, we extend the basis representation of $\mathbb{Z}_K$ to all elements of K :

```

> Z_K := RingOfIntegers(K);
> Basis(Z_K, K);
[
    1,
    a,
    1/5*a^2
]

```

The ring $\mathbb{Z}_K$ is an invariant of the field so, in particular, it does not depend on the choice of the representation of the field or the primitive element. However, for computational purposes the representation is very important. A “bad” choice of primitive element or defining polynomial can potentially slow down even simple operations, such as multiplication of two field elements, to the point where computation becomes impractical.

Elements of $\mathbb{Z}_K$ are printed as sequences of coefficients; elements of the field of fractions of $\mathbb{Z}_K$ are displayed as linear combination of the (named) basis elements; elements of K are printed as polynomials in a :

```

> Z_K![1,2,3];
[1, 2, 3]
> $1/1;
Z_K.1 + 2/1*Z_K.2 + 3/1*Z_K.3
> K!$1;
1/5*(3*a^2 + 10*a + 5)
> Z_K!$1;
[1, 2, 3]

```

Note that “\$1” denotes the last printed result and “!” coerces some object of one structure into another. It is used here to change the representation which depends on the parent object. Finally, “Z_K.1, Z_K.2, Z_K.3” denote the first, second and third basis elements respectively in the fixed basis of $\mathbb{Z}_K$.

It is possible to build finite extensions of K and to consider *relative extensions*. As an example, we will construct the field $E := K[\zeta_3]$, where ζ_3 is a third root of unity:

```

> E<z> := NumberField(Polynomial(K, CyclotomicPolynomial(3)));
> E;
Number Field with defining polynomial $.1^2 + $.1 + 1 over K
> E:Maximal;
E<z>
|
|
K
|
|
Q

E : $.1^2 + $.1 + 1

```

```

K : t^3 - 25
> z^3;
1

```

The `$.1` is used to indicate the transcendental generator of the polynomial ring over K . As E is an extension of K , all invariants of E are automatically relative to K , so, for example, the norm of $z + a$ will be in E . In order to get the absolute norm over $\mathbb{Q}$ we have to indicate the ‘target’ ring as a second argument or use `AbsoluteNorm`:

```

> Norm(z+a);
a^2 - a + 1
> Norm(z+a, Q);
676
> Norm(z+a, E);
z + a

```

A major difference between absolute and relative extensions becomes visible when we look at the ring of integers: $\mathbb{Z}_E$ is a $\mathbb{Z}_K$ -module, but since $\mathbb{Z}_K$, in general, is not a PID, it is no longer a free module. Instead, since $\mathbb{Z}_K$ is a Dedekind ring, the best we can achieve is an *almost-free* representation. There are fractional ideals $\mathfrak{a}_i$ of $\mathbb{Z}_K$ and elements $\omega_i \in E$ such that

$$\mathbb{Z}_E = \sum_{i=1}^m \mathfrak{a}_i \omega_i.$$

Recall that a fractional ideal of $\mathbb{Z}_K$ is a $\mathbb{Z}_K$ -module $\mathfrak{a}$ for which there is some $d \in K$ such that $d\mathfrak{a}$ is an ideal in $\mathbb{Z}_K$. In particular, fractional ideals are, in general, neither ideals nor rings.

By replacing $\mathfrak{a}_i \omega_i$ by $(1/\lambda_i \mathfrak{a}_i)(\lambda_i \omega_i)$ for $0 \neq \lambda_i \in K$ (which does not change the sum), we see that in general

1. $\omega_i \notin \mathbb{Z}_E$
2. $\mathfrak{a}_i \not\subset \mathbb{Z}_K$.

Therefore, in general, integral elements will appear to have denominators (their coefficients are elements of some fractional ideal) and elements that appear to be integral may not be in $\mathbb{Z}_E$, if their integral coefficients are not contained in the proper coefficient ideals. The system $(\mathfrak{a}_i, \omega_i)_i$ ($1 \leq i \leq n$) is called a *pseudo-basis* of $\mathbb{Z}_E$ [Coh00, Fri00]. Strictly, an almost-free representation of the module should have $\mathfrak{a}_1 = \dots = \mathfrak{a}_{n-1} = \mathbb{Z}_K$, but for our purposes the weaker version is sufficient.

It is possible to convert E and $\mathbb{Z}_E$ to absolute extensions. The resulting absolute field allows coercion between the different representations. To illustrate this we create an extension of $\mathbb{Q}$ isomorphic to E :

```

> E_Q := AbsoluteField(E);
> E_Q:Maximal;

```

$$\begin{array}{c} E_Q \\ | \\ | \\ Q \end{array}$$

$$E_Q : t^6 + 3t^5 + 6t^4 - 43t^3 - 69t^2 + 78t + 676$$

```
> Z_E_Q := MaximalOrder(E_Q);
> Z_E ! Z_E_Q.3;
[[0, 0, 1], [0, 0, 0]]
> E_Q ! $1;
1/4680*(-25*E_Q.1^5 - 62*E_Q.1^4 - 124*E_Q.1^3 + 1439*E_Q.1^2
+ 1426*E_Q.1 - 1612)
```

Ideals and fractional ideals of orders are modules over the same coefficient ring as the order. As with the representation of the maximal order, ideals in absolute extensions are free $\mathbb{Z}$ -modules whereas ideals in relative extensions are, in general, not free. Ideals can be represented in three different ways:

- By means of a (pseudo) basis.
- By means of two generators. The ring of integers of a number field is a Dedekind domain, and a straightforward application of the weak approximation theorem shows that every ideal can be generated using only two elements. Furthermore, since one of the elements can be chosen arbitrarily from the ideal, any ideal can be generated by an integer and some algebraic number.
- By means of a single generator when the ideal is principal.

The printed display of an ideal will change depending upon the information known about the ideal.

2.2 Class Groups

The classification of class fields is achieved through the use of generalized class groups which are extensions of the standard class group. The “standard” class group $Cl := Cl_K$ of a number field K is defined as the quotient of the group of all ideals by the subgroup of principal ideals. An important theorem states that the class group is a finite group. Efficient algorithms are known for computing the class group, ie. to compute the structure of the group and for mapping ideals into the abstract abelian group [Coh93, Heß96].

Since the class group algorithm is also used to compute units, S -units, discrete logarithms and to decide whether an ideal is principal, it is important to understand roughly how the method works.

The class group being a finite group, is finitely generated and in fact, it is generated by the set

$$F := \{\mathfrak{p} \mid N(\mathfrak{p}) \leq B_1\}$$

of all prime ideals having norm bounded by some constant B_1 . The set F is called the *factor basis*. Frequently, in practice, since the true bound B_1 is either not known or is too large to be useful, we choose some convenient bound B_1 and work with the corresponding set F . Assuming that the class group is generated by F , its computation is now equivalent to finding a sufficiently large set of relations involving the factor basis elements. By a relation we mean an element $\gamma \in K$ such that

$$\gamma \mathbb{Z}_K = \prod_{\mathfrak{p} \in F} \mathfrak{p}^{v_{\mathfrak{p}}(\gamma)}.$$

The set of such relations, or more precisely the exponent vectors $(v_{\mathfrak{p}}(\gamma))_{\mathfrak{p} \in F}$, form a lattice Λ . Determining the class group is equivalent to being able to compute a basis for Λ . Note that finding a $\mathbb{Z}$ -linear dependency $a = (a_i)_i$ between relations implies the existence of a unit:

$$\sum_i a_i (v_{\mathfrak{p}}(\gamma_i))_{\mathfrak{p} \in F} = 0 \iff \prod_i \gamma_i^{a_i} \in \mathbb{Z}_K^*$$

The algorithm proceeds by constructing “small” elements in $\mathbb{Z}_K$ and testing if they factor over F . This procedure is repeated until sufficient relations are found. As a stopping condition, we compute an approximation to the Euler product

$$E_{B_2} := \prod (1 - \frac{1}{N(\mathfrak{p})}) \prod (1 - \frac{1}{p})$$

where the first product is over all prime ideals dividing prime numbers not exceeding B_2 and the second is over the prime numbers $\leq B_2$. The bound B_2 is chosen independently of B_1 , where usually, $B_2 > B_1$. The analytic class number formula gives a precise relation between the determinant of the lattice Λ (i.e. the class number $h_K := \#\text{Cl}_K$), the regulator R_K of the unit group (generated by dependencies between the relations) and E :

$$E_{\infty} := \lim_{B_2 \rightarrow \infty} E_{B_2} = c_K h_K R_K$$

where c_K is an explicit constant depending on the signature, the discriminant and the number of torsion units in K . The search stops when this equality is approximately satisfied. To be more precise, we stop when

$$\frac{1}{\log 2} E_{B_2} \leq c_K h_K R_K \leq \log 2 E_{B_2}$$

holds. At this point, the class group is known under the assumption that B_1 is sufficiently large and that E_{B_2} is a good approximation to the true Euler product. Therefore, it remains to prove this.

Using methods from the geometry of numbers, it can be shown that every element in Cl can be represented by an ideal $\mathfrak{a}$ of norm $N(\mathfrak{a}) \leq M_K$, where the *Minkowski bound* is defined as

$$M_K := \frac{n!}{n^n} \left(\frac{4}{\pi} \right)^{r_2} \sqrt{|\text{Disc}(K)|}.$$

In particular, this implies that setting $B_1 := M_K$ would guarantee that Cl_K is generated by F . Therefore, to verify the class group, it remains to show that for all prime ideals $\mathfrak{p}$ such that $B_1 \leq N(\mathfrak{p}) \leq M_K$, there is a relation $\gamma_{\mathfrak{p}}$ such that $\mathfrak{p}^{-1}\gamma_{\mathfrak{p}}$ can be factored over F . Since $\text{Disc}(K)$ and therefore M_K grows rapidly with the degree of K , this process can be very slow due to the large number of primes that need to be considered.

On the other hand, assuming the GRH (generalized Riemann hypotheses), Bach [Bac90] derived a much smaller bound B_K ,

$$B_K := 12 \log^2(|\text{Disc}(K)|)$$

which can be taken as the value of B_1 . So again, for every prime $\mathfrak{p}$ between B_1 and B_K we have to find a relation between $\mathfrak{p}$ and F . An additional advantage of using bounds based on the GRH is that we can also bound the error term of the Euler product, that is we get bounds for the error $|E_{\infty} - E_{B_2}|$. Using this, we can compute a precise approximation for E_{B_2} and use this to verify both the class group and the unit group.

In general, the size of B_K makes it impossible to compute class groups that are guaranteed to be correct even assuming GRH. On the other hand, in practical applications where the class group is just an intermediate step, it is frequently possible to independently verify the final result so a provably correct class group is not required. In this case a much smaller bound, $B_1 \leq 1000$, is usually sufficient.

Note, however, by default **Magma** will always use M_K so that class group computations always return guaranteed results at the possible cost of extremely long execution times.

Magma represents class groups as finite abelian groups together with a map between the (abstract) group and the set of ideals of the number field:

```
> K := NumberField(x^2-10);
> Z_K := RingOfIntegers(K);
> Cl, mCl := ClassGroup(Z_K);
> Cl;
Abelian Group isomorphic to Z/2
Defined on 1 generator
Relations:
  2*Cl.1 = 0
> mCl;
Mapping from: GrpAb: Cl to Set of ideals of Z_K
```

We can use this map to create ideals in any class, while `mCl` allows the calculation of inverse images, where `I @@ mCl` represents an arbitrary ideal in the class group:

```
> I := mCl(Cl.1); I;
```

```

Ideal of Z_K
Two element generators:
  [2, 0]
  [2, 1]
> I * K.1;
Ideal of Z_K
Two element generators:
  [0, 2]
  [10, 2]
> I @@ mCl;
Cl.1
> (I^2) @@ mCl;
0
> IsPrincipal(mCl(Cl.1));
false
> IsPrincipal(mCl(Cl.1)^2);
true

```

The unit group, which is computed at the same time as the class group, is also returned as an abstract abelian group together with a map, formally between the group and the order, but actually between the group and the units of $\mathbb{Z}_K$.

2.3 Ray Class Groups

The next step towards class field theory is the introduction of “generalized class groups”, the so-called *ray class groups*. In order to define them, we need to introduce more terminology.

A number field $K = \mathbb{Q}(\alpha)$ of degree n has exactly n different embeddings into the complex numbers, each mapping α to a complex root of its minimal polynomial. An embedding is said to be *real* if the corresponding root is real, in which case the image of K will be a subfield of the real numbers.

Let $\mathfrak{m}_0$ be a fixed integral ideal and let $\mathfrak{m}_\infty$ a subset of the real embeddings of K into $\mathbb{R}$. A (congruence) *modulus* $\mathfrak{m}$ is the pair $(\mathfrak{m}_0, \mathfrak{m}_\infty)$. For $x \in K$ we write

$$x \equiv 1 \pmod{\mathfrak{m}} \iff \begin{cases} v_{\mathfrak{p}}(x-1) \geq v_{\mathfrak{p}}(\mathfrak{m}_0) & \text{for all } \mathfrak{p} \mid \mathfrak{m}_0 \\ s(x) > 0 & \text{for all } s \in \mathfrak{m}_\infty \end{cases}$$

Now let $K_{\mathfrak{m}} := \{x \in K \mid x \equiv 1 \pmod{\mathfrak{m}}\}$ and let $I^{\mathfrak{m}}$ be the set of ideals coprime to $\mathfrak{m}_0$. Then the ray class group modulo $\mathfrak{m}$ is defined as

$$\text{Cl}_{\mathfrak{m}} := I^{\mathfrak{m}}/K_{\mathfrak{m}}.$$

Using standard techniques [Lan94] it is easy to see that $\text{Cl}_{\mathfrak{m}}$ fits into the exact sequence:

$$1 \rightarrow U_{\mathfrak{m}} \rightarrow (\mathbb{Z}_K/\mathfrak{m}_0)^* \times \{\pm 1\}^{\#m_\infty} \rightarrow \text{Cl}_{\mathfrak{m}} \rightarrow \text{Cl}_K \rightarrow 1$$

with $U_{\mathfrak{m}} := \mathbb{Z}_K^* \cap K_{\mathfrak{m}}$. This exact sequence allows us to compute $\text{Cl}_{\mathfrak{m}}$ ([Coh00, HPP01]) and also shows that this is a finite abelian group.

To compute the units of $\mathbb{Z}_K/\mathfrak{m}_0$ we apply the Chinese remainder theorem:

$$(\mathbb{Z}_K/\mathfrak{m}_0)^* = (\mathbb{Z}_K / \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{m}_0)})^* = \prod_{\mathfrak{p}} (\mathbb{Z}_K/\mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{m}_0)})^*.$$

Using Hasse's 1-units we have

$$(\mathbb{Z}_K/\mathfrak{p}^s)^* = (\mathbb{Z}_K/\mathfrak{p})^* \times (1 + \mathfrak{p})/(1 + \mathfrak{p}^s)$$

We can compute $(1 + \mathfrak{p})/(1 + \mathfrak{p}^s)$ by repeatedly using the exact sequence

$$1 \rightarrow (1 + \mathfrak{p}^l)/(1 + \mathfrak{p}^m) \rightarrow (1 + \mathfrak{p}^k)/(1 + \mathfrak{p}^m) \rightarrow (1 + \mathfrak{p}^k)/(1 + \mathfrak{p}^l) \rightarrow 1$$

(for $l \leq 2k$ and $m \leq 2l$) as well as

$$(1 + \mathfrak{p}^l)/(1 + \mathfrak{p}^m) \equiv (\mathfrak{p}^l/\mathfrak{p}^k)^+.$$

In particular, we obtain

$$\phi(\mathfrak{m}_0) := \#(\mathbb{Z}_K/\mathfrak{m}_0)^* = \prod_{\mathfrak{p}|\mathfrak{m}_0} (N(\mathfrak{p}) - 1)N(\mathfrak{p})^{v_{\mathfrak{p}}(\mathfrak{m}_0)-1}$$

which will be important subsequently.

In **Magma**, all of the above sequences are hidden in two intrinsics: **RayClassGroup** which computes $\text{Cl}_{\mathfrak{m}}$ and **RayResidueRing** which computes $(\mathbb{Z}_K/\mathfrak{m}_0)^* \times \{\pm 1\}^{\#m_\infty}$. As with the class group computation, the result will be an abstract abelian group and a map between this group and the (ideals of the) number field. In order to specify the defining modulus $\mathfrak{m}$ we have to supply an integral ideal $\mathfrak{m}_0$ and a sequence $[s_1, \dots, s_t]$ of the indices giving the real places in $\mathfrak{m}_\infty$:

```
> m_0 := 7*5*9*Z_K;
> m_inf := [2];
> R, mR := RayClassGroup(m_0, m_inf);
> R;
Abelian Group isomorphic to Z/2 + Z/6 + Z/24
Defined on 3 generators
Relations:
  6*R.1 = 0
  24*R.2 = 0
  2*R.3 = 0
> mR;
Mapping from: GrpAb: R to Set of ideals of Z_K
```

For two moduli $\mathfrak{m} = (\mathfrak{m}_0, \mathfrak{m}_\infty)$ and $\mathfrak{n} = (\mathfrak{n}_0, \mathfrak{n}_\infty)$ we write $\mathfrak{m} \leq \mathfrak{n}$ or $\mathfrak{n} \mid \mathfrak{m}$ if $\mathfrak{n}_0 \mid \mathfrak{m}_0$ and $\mathfrak{m}_\infty \supseteq \mathfrak{n}_\infty$. In this situation we have a canonical surjection

$$\phi_{\mathfrak{m}}^{\mathfrak{n}} : \text{Cl}_{\mathfrak{m}} \rightarrow \text{Cl}_{\mathfrak{n}} : xK_{\mathfrak{m}} \mapsto xK_{\mathfrak{n}}.$$

Thus it makes sense to define the *conductor* $\mathfrak{f}$ of a given ray class group $\text{Cl}_{\mathfrak{m}}$ as the smallest $\mathfrak{n}$ such that $\text{Cl}_{\mathfrak{m}} \rightarrow \text{Cl}_{\mathfrak{n}}$ is injective.

For moduli $\mathfrak{m}$ and $\mathfrak{n}$, let U be a subgroup of $\text{Cl}_{\mathfrak{m}}$ and let V be a subgroup of $\text{Cl}_{\mathfrak{n}}$ and let $\mathfrak{mn} = (\mathfrak{m}_0\mathfrak{n}_0, \mathfrak{m}_\infty \cup \mathfrak{n}_\infty)$. Then both projections $\phi_{\mathfrak{mn}}^{\mathfrak{m}} : \text{Cl}_{\mathfrak{mn}} \rightarrow \text{Cl}_{\mathfrak{m}}$ and $\phi_{\mathfrak{mn}}^{\mathfrak{n}} : \text{Cl}_{\mathfrak{mn}} \rightarrow \text{Cl}_{\mathfrak{n}}$ are surjective. We write $(\text{Cl}_{\mathfrak{m}}, U) \equiv (\text{Cl}_{\mathfrak{n}}, V)$ if U and V agree in $\text{Cl}_{\mathfrak{mn}}$, that is if $(\phi_{\mathfrak{mn}}^{\mathfrak{m}})^{-1}(U) = (\phi_{\mathfrak{mn}}^{\mathfrak{n}})^{-1}(V)$. Using this, the conductor of $(\text{Cl}_{\mathfrak{m}}, U)$ is the smallest module $\mathfrak{n}$ for which there is a subgroup $V \leq \text{Cl}_{\mathfrak{n}}$ such that $(\text{Cl}_{\mathfrak{m}}, U) \equiv (\text{Cl}_{\mathfrak{n}}, V)$.

It is important to remember that ray class groups depend upon both the class group and the unit group. If either is missing, it will be computed using **Magma**'s default settings for unconditionally proven results. Thus for large fields it might be necessary to compute both the class group and the unit group conditionally in advance, since **Magma** will use them if they are already known — regardless of whether they have been proven or not.

2.4 Class Field Theory

Now that we have defined (almost) all objects needed to compute class fields, we are ready to start. As mentioned before, class fields will be parametrized by ray class groups and their quotients. Before stating the theorems, we will present an example. Starting with $k := \mathbb{Q}[\sqrt{10}]$ (of class number 2), we will define the full ray class field corresponding to modulus $\mathfrak{m} = (\mathfrak{m}_0, \mathfrak{m}_\infty)$, where $\mathfrak{m}_0 = 5\mathbb{Z}_k$ and $\mathfrak{m}_\infty$ contains both real embeddings:

```
> k := NumberField(x^2-10);
> Z_k := RingOfIntegers(k);
> R, mR := RayClassGroup(5*Z_k, [1,2]);
> A := AbelianExtension(mR);
> A;
FldAb, defined by (<[5, 0]>, [1 2])
of structure: Z/4
```

In order to see what A really looks like, we have to convert it to a number field:

```
> K := NumberField(A);
> K;
Number Field with defining polynomial $.1^4 + (36*k.1 + 280)
*$.1^2 + 4200*k.1 + 13820 over k
> G, _, mG := AutomorphismGroup(A);
> G;
Abelian Group isomorphic to Z/4
```

Defined on 1 generator

Relations:

$$4 * G.1 = 0$$

Note that the group $\text{Aut}(K/k)$ of k -automorphisms of K is isomorphic to Cl_m , thus illustrating the first part of the correspondence between abelian fields and ray class groups. Furthermore, this isomorphism is available explicitly, being the extension of the map that sends unramified prime ideals to their Frobenius automorphism. The Frobenius automorphism $(\mathfrak{p}, K/k) \in \text{Aut}(K/k)$ of a prime ideal $\mathfrak{p}$ is the unique automorphism such that

$$(\mathfrak{p}, K/k)(x) \equiv x^{N(\mathfrak{p})} \pmod{\mathfrak{p}\mathbb{Z}_K}$$

for all $x \in K$. We will demonstrate this for a prime ideal over 13:

```
> lp := Decomposition(Z_k, 13);
> p := lp[1][1];
> Norm(p);
13
```

We compare each automorphism with $x \mapsto x^{13}$ modulo $\mathfrak{p}\mathbb{Z}_K$, by looking for $\sigma \in \text{Aut}(K/k)$ such that for $x := K.1$, (the primitive element of K) we have $\sigma(x) - x^{13} \in \mathfrak{p}\mathbb{Z}_K$:

```
> Z_K := MaximalOrder(A);
> P := Z_K !! p; // to define p Z_K
> [ <s, (s @ mG)(K.1) - K.1^13 in P> : s in G];
[
  <0, false>,
  <G.1, false>,
  <2*G.1, false>,
  <3*G.1, true>
]
> s1 := FrobeniusAutomorphism(A, p);
> s2 := (3*G.1) @ mG;
> s1 eq s2;
true;
> p @@ mR;
3*R.1
```

So indeed, if we map $R.1$ to $G.1$ we get an isomorphism of finite groups. The image of a prime ideal in R gives (via this map) the corresponding Frobenius automorphism. The multiplicative extension of the Frobenius map is called the *Artin map* and is denoted by $(\mathfrak{a}, K/k)$. Thus, for $\mathfrak{a} = \prod \mathfrak{p}_i^{l_i}$ where all the primes occurring in the factorization are unramified, we have

$$(\mathfrak{a}, K/k) := \prod (\mathfrak{p}_i, K/k)^{l_i}.$$

In general we need to work with those subfields of the full ray class field which correspond to quotients of the ray class groups. This is necessary since the groups in question (and therefore the fields) become very large:

```

> R, mR := RayClassGroup(8*3*5*7*Z_k);
> #R;
1536
> A := AbelianExtension(mR);
> A;
FldAb, defined by (<[840, 0]>, [])
of structure: Z/2 + Z/2 + Z/2 + Z/8 + Z/24

> K := NumberField(A);
> K;
Number Field with defining polynomial [ $.1^2 - 3*k.1 - 10,
$.1^2 - 3*k.1 - 24, $.1^2 - k.1 - 8, $.1^8 - 420*$.1^6
+ 57860*$.1^4 - 2788320*$.1^2 + 21715280, $.1^8 -
41160*$.1^6 + (-14776440*k.1 + 408615900)*$.1^4 +
(60243779120*k.1 - 1063468399280)*$.1^2 -
28477255077040*k.1 + 246599356889480, $.1^3 - 147*$.1
- 637] over k

```

Although it is possible to create the field K , its large degree makes it time consuming to construct and expensive to work with.

We are going to use the existence theorem of class field theory in the following form: There is a bijection between abelian extensions K/k and pairs (Cl_m, U) of subgroups of ray class groups (modulo $\equiv$) mapping unramified prime ideals to their Frobenius automorphism.

Suppose we only wish to work with a subfield L of degree 4. By the above theorem, L corresponds to a subgroup U of R that has a quotient of order 4. The correspondence yields $\text{Aut}(L/k) \cong R/U$. In order to define L in **Magma**, we have to define U and R/U and provide a map $\phi : R/U \rightarrow I$ that behaves identically to the map returned by `RayClassGroup` but with R/U as the group.

```

> U := sub<R | R.1, R.2, R.3, 4*R.4, R.5>;
> q, mq := quo<R|U>;
> q;
Abelian Group isomorphic to Z/4
Defined on 1 generator
Relations:
    4*q.1 = 0
> phi := Inverse(mq)*mR;
> phi;
Mapping from: GrpAb: q to Set of ideals of Z_k
Composition of Mapping from: GrpAb: q to GrpAb: R and
Mapping from: GrpAb: R to Set of ideals of Z_k
> A1 := AbelianExtension(phi);
> A1;
FldAb, defined by (<[840, 0]>, [])

```

```

of structure: Z/4
> NumberField(A1);
Number Field with defining polynomial $.1^4 + (-36*k.1 -
280)*$.1^2 + 840*k.1 + 9020 over k

```

Using this technique we can investigate all cyclic degree 4 subfields of K to find, for example, the subfields of maximal and minimal discriminant:

```

> L := Subgroups(R:Quot := [4]);
> #L;
48
> AL := [ AbelianExtension(Inverse(mq)*mR)
>         where _, mq := quo<R|i'subgroup> : i in L];
> Minimum([AbsoluteDiscriminant(i) : i in AL]);
1024000000 3
> Maximum([AbsoluteDiscriminant(i) : i in AL]);
2498119335936000000 47

```

The function `Subgroups(R: Quot := [4])` computes a list of all subgroups U of R such that the quotient R/U is isomorphic to $\mathbb{Z}/4\mathbb{Z}$.

Suppose now that $G \leq \text{Aut}(k/\mathbb{Q})$ is a subgroup of the group of $\mathbb{Q}$ -automorphisms of k such that the module $\mathfrak{m}$ is invariant under the action of G so that G acts on $\text{Cl}_{\mathfrak{m}}$. Finally, let $k_0 := \text{Fix}(G) := \{x \in k \mid \sigma(x) = x \forall \sigma \in G\}$. Then L/k corresponding to $(\text{Cl}_{\mathfrak{m}}, U)$ is normal over k_0 if and only if G acts on U (and therefore on $\text{Cl}_{\mathfrak{m}}/U$). In this situation the elements of G can be extended to k_0 -automorphisms of L/k_0 . The groups are connected by the exact sequence:

$$1 \rightarrow \text{Cl}_{\mathfrak{m}}/U \rightarrow \text{Aut}(L/k_0) \rightarrow G \rightarrow 1.$$

We will use this result in two different ways. Firstly, it allows us to decide normality of extensions simply by looking at the ideal groups. Secondly, by using results from extension theory of groups, we can determine invariants of the resulting Galois group $\text{Aut}(K/k_0)$ by analyzing the action of G on $\text{Cl}_{\mathfrak{m}}/U$.

We apply this to search for normal (over $\mathbb{Q}$) degree 8 subfields of K occurring in the list `AL` and then further restrict the selection to non-central extensions:

```

> N := [ x : x in AL | IsNormal(x:All)];
> nC := [ x : x in N | not IsCentral(x:All)];
> LG := [ AutomorphismGroup(x:All) : x in nC];
> LP := [ CosetImage(x, sub<x|>) : x in LG];
> [ TransitiveGroupIdentification(x) : x in LP];
[ 4, 4, 4, 4, 5, 5, 5, 5 ]
> TransitiveGroupDescription(8, 4);
D_8(8)=[4]2
> TransitiveGroupDescription(8, 5);
Q_8(8)

```

The parameter `All` indicates that all $\mathbb{Q}$ -automorphisms of the base field k will be considered. Otherwise, `IsNormal` would be trivially true for all class

fields. It is possible to restrict to a subgroup of the full automorphism group in this situation.

For later applications we note some formal properties of the class field K associated to the pair (Cl_m, U) :

1. The primes of k that are ramified in K/k are exactly those dividing the conductor $\mathfrak{f}_{K/k}$ of (Cl_m, U) .
2. A prime $\mathfrak{p}$ is wildly ramified if and only if $\mathfrak{p}^2$ divides $\mathfrak{f}_{K/k}$.
3. For cyclic extensions of prime degree l we have the following formula for the relative discriminant:

$$\mathfrak{d}_{K/k} = \mathfrak{f}_{K/k}^{l-1}$$

In general, the discriminant can be computed from (Cl_m, U) but the formula is more involved [Coh00].

4. An unramified prime $\mathfrak{p}$ has degree $f = f(K/k) = \text{ord}_{\text{Cl}_m/U}(\mathfrak{p}U)$. In particular $\mathfrak{p}$ has degree 1, and is therefore totally split, if and only if $\mathfrak{p} \in U$.

3 Global Function Fields

Global function fields are similar to number fields in a very precise way. In practical terms this means that class field theory as outlined above applies in the same way for function fields as it does for number fields. For **Magma** this implies that the user interface and the presentation of fields is closely related to those used in the case of number fields. Internally, a great deal of the code is shared between the two modules. In addition to the number theoretic view, global function fields also admit a geometric point of view as plane curves. Although it is possible in **Magma** to take this viewpoint, our presentation emphasizes the number theoretic view. However, the geometric point is sufficiently strong to enforce some difference in the language.

3.1 Basic Properties

Global function fields are finite separable extensions of $\mathbb{F}_q(x)$, where $\mathbb{F}_q$ is a finite field with $q = l^r$ elements, l a prime. A good way of viewing global function fields is to think of $\mathbb{F}_q(x)$ as the equivalent of $\mathbb{Q}$ and of $\mathbb{F}_q[x]$ as the equivalent of $\mathbb{Z}$. Setting up a function field is more complicated than setting up a number field, but once created, it behaves very much like a number field. We will define an extension of $k := \mathbb{F}_4$. The multiplicative group of k is cyclic of order 3, and we take $\omega := k.1$ as a generator for it. The function field will be $\mathbb{F}_q(x)[y]$ where $y^4 - (\omega x^2)y^2 + y - (\omega^3 x^3) = 0$:

```
> k := GF(4);
> kx<x> := FunctionField(k);
> kxy<y> := PolynomialRing(kx);
> K := FunctionField(y^4-k.1*x^2*y^2 + y - x^3*k.1^3);
```

As in the number field case, elements in K are printed as polynomials in the primitive element. Since we did not assign a name, the primitive element will be printed as $K.1$.

Although it is possible to formulate class field theory in the same way as for number fields, namely using ideals and embeddings, it is more convenient to use the geometric language of places and divisors. A *place* of a function field K is an equivalence class of valuations defining a common topology. It is possible to identify places with prime ideals of suitable subrings of K . Almost all of the places can be represented in terms of prime ideals of the integral closure of $k[x]$ in K , the *finite maximal order*. Exceptions are the places corresponding to the real and complex embeddings in the number field case: they have to be represented using prime ideals of the integral closure of $k[1/x]$ (the valuation ring of the degree valuation). The primes of the finite maximal order are called *finite places*, while the non-finite places are called *infinite places*.

From a theoretical point of view the distinction is artificial, as there is no difference between the finite and the infinite places since, by making the change of variable, $x \mapsto 1/x$, all the infinite places become finite. Thus the classification of a place as infinite or finite depends upon the representation rather than being an invariant of the field. It is important to remember that once K is created, the classification of finite and infinite places is fixed.

In **Magma** this also leads to the fact that there are two different rings of integers attached to each function field. Thus, “**MaximalOrderFinite**” represents the integral closure of $\mathbb{F}_q[x]$ in K and “**MaximalOrderInfinite**” represents the integral closure of the valuation ring of the degree valuation. The latter having only a finite number of prime ideals, is a semi-local Dedekind ring and therefore a principal ideal domain. The PID property of the “infinite maximal order” is not used in **Magma**. Since orders in function fields are subrings of their respective maximal orders, we also distinguish between finite and infinite orders.

Elements of orders are represented with respect to a fixed basis. Since $\mathbb{F}_q[x]$ considered as a polynomial ring over a field is Euclidean, finite orders always have a free basis over $\mathbb{F}_q[x]$. Similarly, since valuation rings are Euclidean, the infinite orders also have a free representation.

```
> MF := MaximalOrderFinite(K);
> MF;
Maximal Equation Order of K over Univariate Polynomial
  Ring in x over GF(2^2)
> MI := MaximalOrderInfinite(K);
> MI;
Maximal Equation Order of K over Valuation ring of
  Univariate rational function field over GF(2^2)
  with generator 1/x
> Basis(MF);
[
```

```

1,
K.1,
K.1^2,
K.1^3
]
> Basis(MI);
[
1,
1/x*K.1,
1/x^2*K.1^2,
1/x^3*K.1^3
]

```

As in the case of number fields, a function field can be repeatedly extended:

```

> Kt<t> := PolynomialRing(K);
> E := FunctionField(t^2+t+K.1);
> E:Maximal;
E
|
| t^2 + t + K.1
|
K
|
| y^4 + k.1*x^2*y^2 + y + x^3
|
Univariate rational function field over GF(2^2)
Variables: x

```

Also, as for number fields, invariants of the field and the orders depend upon the representation, so that, for example, discriminants are ideals while norms and traces are relative to the coefficient field.

3.2 Places and Divisors

As mentioned above, one of the important differences between number and function fields, is that since all places give rise to discrete valuations and are, from a geometric point of view, of equal importance, there are no truly distinguished infinite places. Rather than using prime ideals with the additional problem of the correct choice of the ring, we instead use places. The same idea is extended to ideals: since ideals are (in the number field case) uniquely representable as a product of prime ideals, we will replace them by divisors and use additive notation instead of multiplicative: A divisor D is a formal linear combination of places P_i

$$D = \sum n_i P_i$$

where $n_i \in \mathbb{Z}$ and all but finitely many n_i are 0.

One way of viewing divisors is as factored ideals. In fact, all divisors can be represented as a pair of ideals, one from the finite part (an ideal in the finite maximal order) and the other combining the places and prime ideals of the infinite maximal order. Formally, the correspondence is obtained by replacing each place by its prime ideal and the linear combination by a power product:

$$\sum n_i P_i \leftrightarrow (\prod \mathfrak{p}_i^{n_i}, \prod (\mathfrak{p}_j^\infty)^{n_j})$$

where the $\mathfrak{p}_i$ are the finite prime ideals, and $\mathfrak{p}_j^\infty$ denotes an infinite prime.

To create divisors we combine places of low degree or decompose functions (elements of the function field):

```
> lp1 := Places(K, 1);
> lp1;
[ (1/x, 1/x*K.1 + 1/x), (1/x, 1/x*K.1 + (k.1^2*x + 1)/x),
  (x, K.1), (x, K.1 + 1), (x, K.1 + k.1), (x, K.1 + k.1^2),
  (x + k.1, K.1 + x + k.1^2), (x + k.1^2, K.1 + x),
  (x + 1, K.1 + x + k.1^2) ]
> lp2 := Places(K, 2);
> D := 10 * lp2[2] - lp1[1];
10*(x^2 + x + k.1^2, K.1 + x^2 + k.1*x + 1) - (1/x, 1/x*K.1 +
  1/x)
> Ideal(lp1[3]);
Ideal of MF
Generators:
x
K.1
> Ideal(lp1[1]);
Ideal of MI
Generators:
1/x
1/x*K.1 + 1/x
> Ideals(D);
Ideal of MF
Basis:
[x^20 + x^18 + k.1*x^16 + x^12 + x^10 + k.1*x^8 + k.1*x^4 +
  k.1*x^2 + k.1^2 0 0 0]
[k.1*x^18 + k.1*x^16 + x^14 + x^12 + x^8 + k.1*x^6 + x^3 + 1
  1 0 0]
[x^18 + x^16 + k.1^2*x^12 + x^10 + x^6 + x^4 + k.1*x^2 0
  1 0]
[k.1*x^18 + k.1*x^17 + k.1*x^16 + k.1^2*x^15 + k.1^2*x^14 +
  x^11 + x^10 + k.1^2*x^9 + x^7 + k.1*x^6 + x^4 + k.1*x^3 +
  k.1*x^2 + k.1^2*x + 1 0 0 1]
Fractional ideal of MI
Basis:
```

```

[1/x  0  0  0]
[ 0 1/x  0  0]
[ 0  0 1/x  0]
[ 0 k.1  0  1]

```

Denominator: 1/x

In addition to invariants arising from the number theoretic point of view, function fields also admit geometric invariants such as the genus and invariants based on the Riemann-Roch theorem. Although geometric invariants are representation independent, the time taken to compute them depends heavily on the representation chosen. In general, the representation best suited for invariant computation and general arithmetical operations is the representation as a simple extension of a rational function field. In **Magma**, this representation can be computed using **RationalExtensionRepresentation**. As in the case of **AbsoluteField** on the number field side, the newly created field allows coercion to and from the original field:

```

> Ea := RationalExtensionRepresentation(E);
> Ea:Maximal;
Ea
|
| y^8 + (k.1*x^2 + 1)*y^4 + (k.1*x^2 + 1)*y^2 + y + x^3
|
Univariate rational function field over GF(2^2)
Variables: x
> E!Ea.1;
E.1
> Ea!(E.1+K.1);
Ea.1^2

```

As with the number fields, the creation of the rational extension representation and the conversion between it and the relative extension can be very time consuming. In addition, the representation as a tower of fields is often more compact.

3.3 Ray Divisor Class Groups

The map

$$K^* \rightarrow D : x \mapsto (x) := \sum v_P(x)P$$

allows us to map functions into divisors to get the subgroup of *principal divisors* of the full divisor group. The quotient group of divisors modulo principal divisors is defined as the divisor class group of K , and it may be regarded as the analogue of the ideal class group of a number field. Their computation follows the same method as outlined for the number fields [Heß99]. A notable difference is that the GRH for function fields is known to be true, so we can use the improved bounds that it gives. In general, divisor class groups and ray

divisor class groups of function fields behave similarly to their counterparts in number fields. However, there is one important difference:

```
> C, mC := ClassGroup(K);
> C;
Abelian Group isomorphic to Z/185 + Z
Defined on 2 generators
Relations:
185*$ .1 = 0
```

The class group has a free part and is thus an infinite group. This reflects the fact that function fields have divisors of arbitrarily large degree and that principal divisors are always of degree 0. The *degree* of a divisor $D = \sum v_P(D)P$ is defined as $\deg(D) = \sum v_P(D) \deg(P)$, where the degree of a place is the degree of its residue class field O_P/P over the constant field k .

In order to introduce the ray divisor class groups, let us fix an effective divisor M , that is $M = \sum n_i P_i$ with $n_i \geq 0$ for all i . As for the number fields we define

$$x \equiv 1 \pmod{*} M \iff v_P(x-1) \geq v_P(M)$$

for all $P|M$. The ray class group Cl_M is then defined as the quotient of divisors coprime to M by the principal divisors (x) generated by $x \equiv 1 \pmod{*} M$. It fits into an exact sequence:

$$1 \rightarrow \mathbb{F}_q^* \rightarrow \prod (O_P/P^{v_P(M)})^* \rightarrow \text{Cl}_M \rightarrow \text{Cl} \rightarrow 1$$

The product $\prod (O_P/P^{v_P(M)})^*$ can be evaluated using 1-units. The function field situation is easier than the number field case, as one is able to write down an explicit basis for the local factors. A short analysis of the groups shows that

$$(O_P/P^r)^* \cong (O_P/P)^* \oplus (\mathbb{F}_q)^{n_1} \oplus (\mathbb{F}_{q^2})^{n_2} \oplus \dots$$

where the sum is finite and depends on r only [Aue99].

In **Magma**, ray divisor class groups are represented in the same way as for number fields: as an abelian group and a map between the group and the group of divisors coprime to M :

```
> M := 2*lp1[2]+3*lp2[3];
> R, mR := RayClassGroup(M);
> R;
Abelian Group isomorphic to Z/2 + Z/2 + Z/4 + Z/4 + Z/11100 + Z
Defined on 6 generators
Relations:
2*R.1 = 0
2*R.2 = 0
4*R.3 = 0
4*R.4 = 0
11100*R.5 = 0
```

Again there is a free part inherited from the divisor class group that we have to factor out to get finite quotients.

3.4 Class Field Theory

Class fields are abelian extensions of function fields. In exactly the same way as for number fields, these extensions are parametrized by the finite quotients of ray class groups and the splitting behaviour of the places is governed by their order in the class group. Since the fields are uniquely determined by the class groups, we are able to compute invariants by just analyzing the groups. In particular, we are able to compute the degree, genus, discriminant divisor, the exact constant field, and the number of places of any degree *without* computing defining equations. This makes class field techniques very powerful tools for constructing fields with many rational places. Also, since this only requires computations in the base field, we are able to compute invariants for much larger fields:

```
> U := sub<R|lp1[1] @@ mR >;
> degree := #quo<R|U>;degree;
710400
> Genus(M, U);
4043361
> DiscriminantDivisor(M, U);
1184000*(1/x, 1/x*K.1 + (k.1^2*x + 1)/x) + 2030560*(x^2 +
  k.1^2*x + k.1^2, K.1 + x^2 + x + k.1^2)
> DegreeOfExactConstantField(M, U);
1
> NumberOfPlacesOfDegreeOne(M, U);
710400
```

Even if the field is small enough for direct computation, it is much faster to use the indirect methods:

```
> U := sub<R|U, lp2[4] @@ mR, lp1[4]@@mR, lp2[1]@@mR,
  lp2[2]@@mR>;
> degree := #quo<R|U>;degree;
4
> time Genus(M, U);
17
Time: 0.290
> NumberOfPlacesOfDegreeOne(M, U);
> E := AbelianExtension(M, U);
> time Genus(E);
17
Time: 8980.890
```

4 Applications

The applications of class field theory (CFT) presented here are based on two properties: as CFT can be used to construct large families of non-isomorphic fields it can be used for the compilation of extensive tables of number (or function) fields with a given Galois group or a given degree [FK03, CDyDO00]. Other approaches based on the geometry of numbers have the disadvantage of only finding a large number of candidates for defining polynomials that have to be further processed [FP93, FPDH98, Oli90].

The other important property of CFT is that some invariants that are hard to compute from the polynomials can be easily read off the class group representation. For example, while it is difficult to construct a field with a large number of rational places, it is rather easy to compute the precise number once the parameterizing class group is found.

In what follows, we give two examples for the construction of families of fields and one of a probabilistic approach to good codes.

4.1 C_l -fields

We will demonstrate one approach to solve the following problem:

Given a field k , find ‘all’ cyclic extensions of degree l , where l is a prime.

Formulated in this way, the problem obviously has no finite solution, so we restrict further:

Given a number field k , find all cyclic degree l extensions K of absolute discriminant $|\text{Disc}(K/\mathbb{Q})| \leq B$ for some bound B .

In the function field case this becomes

Given a function field k , find all cyclic degree l extensions K of genus $g \leq B$ for some bound B having the same exact constant field as k .

Both problems are very similar - they differ only in minor details. So let us start with the number field case.

Class field theory gives us the following information about the relative discriminant of K/k :

$$\text{Disc}(K/k) = \mathfrak{f}_{K/k}^{l-1}$$

Basic number theory tells us:

$$|\text{Disc}(K/\mathbb{Q})| = |\text{Disc}(k/\mathbb{Q})|^l N(\text{Disc}(K/k)).$$

Putting this together, we get a bound for the conductor $\mathfrak{f}_{K/k}$:

$$N(\mathfrak{f}_{K/k}) \leq \frac{B}{|\text{Disc}(k/\mathbb{Q})|^l} =: B'$$

so the task is reduced to enumerating all integral ideals of norm $\leq B'$ — which is a finite problem. For each ideal $\mathfrak{f}$ we then have to enumerate all subgroups U of $\text{Cl}_{\mathfrak{f}}$ that give rise to a cyclic quotient of size l having conductor $\mathfrak{f}$ and then compute the fields. To avoid repetition in our list, for each subgroup U we will compute the conductor of $(\text{Cl}_{\mathfrak{f}}, U)$ and only proceed if the conductor is $\mathfrak{f}$.

The method thus outlined will indeed find all fields K/k subject to our conditions only once, so it provides us with an algorithmic solution. However, we can easily make further improvements to the method: We need to find ideals such that the ray class group has a cyclic factor of degree l , so the ray class number has to be divisible by l . Referring back to the exact sequence used to compute $\text{Cl}_{\mathfrak{m}}$, this means that we have only to consider prime ideals $\mathfrak{p}$ such that $N(\mathfrak{p}) - 1$ is divisible by l (which immediately implies $l|p - 1$) or such that $l \in \mathfrak{p}$ — which will greatly reduce the number of useless candidates.

A second optimization concerns the exponents of the prime in $\mathfrak{f}$. We already know that a prime $\mathfrak{p}$ is wildly ramified if and only if $\mathfrak{p}^2|\mathfrak{f}$. In addition it is known that if $\mathfrak{p}$ is wildly ramified then $\min \mathfrak{p} \cap \mathbb{Z}_{>0} \leq l$ since only primes smaller than the degree can be wildly ramified.

So we essentially have to enumerate only *square-free* ideals $\mathfrak{f}$! One can further analyze the contribution of the wildly ramified primes and give tight bounds for the exponents. Let $\mathfrak{p}$ be a prime in k that is wildly ramified in K/k , $\mathfrak{p}\mathbb{Z}_K = \prod \mathfrak{P}_i^{e_i}$. Serre ([Ser95]) asserts that $v_{\mathfrak{p}}(\mathfrak{d}_{K/k}) \leq l - 1 + lv_{\mathfrak{p}}(l)$ so that we get $v_{\mathfrak{p}}(\mathfrak{f}_{K/k}) \leq 1 + \frac{lv_{\mathfrak{p}}(l)}{l-1}$. This implies that the potentially wildly ramified primes are those that divide l only.

To enumerate all (square-free) ideals of norm $\leq B$ we first generate all prime ideals of norm not exceeding B by decomposing all (small) prime numbers.

```
> k := NumberField(x^2-10);
> B := 5*10^7; l := 3;
> Z_k := MaximalOrder(k);
> Bs := Floor(B / AbsoluteDiscriminant(Z_k)^l); Bs;
781
> lp := [ x : x in [2..Bs] | IsPrime(x) and x mod l eq 1 ];
> LP := &cat [ [ x[1] : x in Decomposition(Z_k, y) |
Norm(x[1]) le Bs ] : y in lp];
```

Next we have to enumerate the square-free ideals of norm not exceeding B :

```
> L := [1*Z_k];
> for i in LP do
>   b := Floor(Bs / Norm(i));
>   L cat:= [ x*i : x in L | Norm(x) le b];
> end for;
> #L;
76
```

Finally, we have to supplement those ideals by including the “exceptional” primes $\mathfrak{p}_{3,1}$ and $\mathfrak{p}_{3,2}$ with larger exponents for the wild ramification.

```
> p3_1 := Decomposition(Z_k, 3)[1][1];
> p3_2 := Decomposition(Z_k, 3)[2][1];
> b3 := Floor(1 + 3*Valuation(Z_k!3, p3_1));
> b3;
4
> L3_1 := [p3_1^i : i in [1..4]];
> L3_2 := [p3_2^i : i in [1..4]];
> L cat:= &cat [ [ x*i : x in L | Norm(x) le b]
               where b := Floor(Bs/Norm(i)) : i in L3_1];
> L cat:= &cat [ [ x*i : x in L | Norm(x) le b]
               where b := Floor(Bs/Norm(i)) : i in L3_2];
> #L;
192
```

Now we can set up the class fields:

```
> LA := [];
> for i in L do
>   "Now doing: ", i;
>   R, mR := RayClassGroup(i);
>   l := Subgroups(R: Quot := [3]);
>   "with ", #l, " possible subgroups";
>   for j in l do
>     _, mq := quo<R | j'subgroup>;
>     A := AbelianExtension(Inverse(mq)* mR);
>     if Conductor(A) ne i then continue; end if;
>     Append(~LA, A);
>   end for;
> end for;
Now doing: Principal Ideal of Z_k
Generator:
[1, 0]
with 0 possible subgroups
Now doing: Principal Ideal of Z_k
Generator:
[7, 0]
with 1 possible subgroups
...
Now doing: Ideal of Z_k
Basis:
[ 9 72]
[ 0 81]
with 1 possible subgroups
> #LA;
```

So there are far fewer fields than there are candidate conductors.

If B is much larger, the simple approach will not work as the list of ideals will become prohibitively large. Also, using just a simple loop makes it impossible to restart the job or distribute the computations. It is an interesting combinatorial problem to do this efficiently.

The corresponding computation for function fields is not difficult. Here we are interested in the following question:

Given a global function field k , find all cyclic extensions K of degree l such that the genus of K is bounded by B .

In order to get similar bounds for the degree of the possible conductors, we use the Hurwitz-Genus-Formula [Sti93]

$$g_K - 2 = \frac{[K : k]}{[\text{ECF}(K) : \text{ECF}(k)]} (2g_k - 2) + \deg(\text{Diff}(K/k))$$

where $\text{ECF}(K)$ denotes the algebraic closure of $\mathbb{F}_q$ in K , that is the (exact) constant field of K and $\text{Diff}(K/k)$ denotes the different divisor. From the general theory we know that $N(\text{Diff}(K/k)) = \text{Disc}(K/k) = f_{K/k}^{l-1}$ and $\deg(\text{Diff}(K/k)) = \deg N(\text{Diff}(K/k))$. Combining this we get

$$2g_K - 2 = l(2g_k - 2) + (l - 1) \deg f_{K/k}$$

if we require $\text{ECF}(K) = \text{ECF}(k) = \mathbb{F}_q$. This now implies

$$\deg f_{K/k} \leq \frac{2B - 2 - l(2g_k - 2)}{l - 1}$$

The remainder is completely analogous to the number field case. We start by collecting all places of “small” degree of $\mathbb{F}_q(x)$, split them in k to get all “small” places in k , then compile lists of divisors of small degree in k followed by the conductor computation and the definition of the abelian extensions.

4.2 S_3 -fields

Solvable extensions can, in principle, be obtained by computing towers of abelian extensions. Although tempting from a theoretical point of view, this is in general not very efficient as we will see in the sequel. However, for small solvable groups this is feasible.

We will investigate the following problem:

Given a field k find all cubic extensions K/k such that $\text{Gal}(K/k) \cong S_3$, in particular, we want to construct non-normal fields.

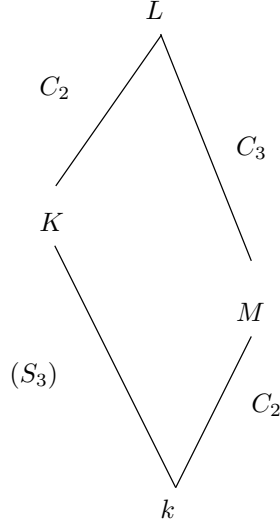


Fig. 1. S_3 structure

We summarize properties of S_3 fields K/k , with the assistance of figure 1. The normal closure of the field K/k will be denoted by L , so L/k is a sextic field with a Galois group isomorphic to S_3 and L/K is a quadratic extension. There is a quadratic field M/k such that L/M is a cyclic cubic field. So one way of constructing K is to find “all” quadratic extensions M of k (which we can do easily using the approach of the previous section), then find all C_3 extensions L of those (again, using the previous section) and finally for all L that are constructed this way and that are normal and non-abelian, find K as a subfield. Although this would work, it is not very efficient. The great majority of sextic fields constructed this way will not be normal.

To get a more efficient version we analyze the various relative discriminants more carefully and thus get restrictions on the possible conductors. The starting point for this situation is the theorem of Klüners ([FK03]):

$$\text{Disc}(K/k) = \text{Disc}(M/k)N_{M/k}(\mathfrak{f}_{L/M}).$$

In fact, one can do even better ([Coh00]) and show that $\mathfrak{f}_{L/M} = \mathfrak{b}\mathbb{Z}_M$ for some ideal $\mathfrak{b}$ of $\mathbb{Z}_k$. Thus we get $\text{Disc}(K/k) = \mathfrak{a}\mathfrak{b}^2$, where $\mathfrak{a}$ is a conductor of a quadratic extension M/k and $\mathfrak{b}$ is a cubic conductor.

Combining the above result with the algorithm from the previous section gives us strong restrictions on the ideals: both have to be square-free (outside 2, 3), their norm is bounded, and, since the ray class group of $\mathfrak{b}$ has to admit a C_3 quotient, only primes $\mathfrak{p}$ with $N_{L/\mathbb{Q}}(\mathfrak{p}\mathbb{Z}_L) \equiv 1 \pmod{3}$ need to be considered.

A second criterion used to reduce the search comes from the Galois group. Suppose $\mathfrak{m}$ is a module in a normal extension L/k , $U < \text{Cl}_{\mathfrak{m}}$ is a subgroup and M/k corresponds to $(\text{Cl}_{\mathfrak{m}}, U)$. Then

- M/k is normal if and only if $\sigma(U) = U$ for all $\sigma \in \text{Aut}(L/k)$.
- M/k is central if and only if $\sigma(\mathfrak{a})U = \mathfrak{a}U$ for all $\sigma \in \text{Aut}(L/k)$.

So an abelian extension of a normal field is normal if the corresponding group is fixed as a subset and central if it is point-wise fixed. Since in the case of extensions M/k of cyclic fields, central extensions and abelian extension coincide, we have a fast check for L/k being normal and non-abelian which implies $\text{Gal}(L/k) = S_3$.

We use a nested loop: an outer loop to find all quadratic extensions M of k followed by an inner loop to find the cyclic cubic extensions L/M such that L/k is normal but non-abelian.

Assuming that the procedure of the last section is implemented as the function `CyclicExtensions` with arguments k , l and B we can summarize the method in the following program:

```

> B := 10^6;
> S3 := [];
> L_C2 := CyclicExtensions(k, 2, B);
> for i in L_C2 do
>   L := NumberField(i);
>   sigma := Automorphisms(L)[2];
>   La := AbsoluteField(L);
>   M_C3 := CyclicExtensions(La, 3, B);
>   for j in M_C3 do
>     if Conductor(j) ne sigma(Conductor(j)) then
>       continue;
>     end if;
>     c := j'DefiningGroup;
>     a := c'm0;
>     U := Kernel(c'GrpMap);
>     S := InducedAutomorphism(c'RcgMap, sigma, Minimum(a));
>     if U ne S(U) then continue; end if;
>     q, mq := quo<Domain(c'RcgMap) | U>;
>     if q.1 ne S(q.1) then continue; end if;
>     Append(~S3, j);
>   end for;
> end for;

```

Again, since the basis of the above procedure is purely class field theoretic, it applies equally to function fields if we use the Hurwitz-Genus-Formula to relate the conductors and discriminants to the genera of the fields in question.

4.3 Codes

A very important application of function fields is the construction of error-correcting codes. Error-correcting codes are used to transmit data over unreliable communication lines. They are used to encode the data in such a way that a certain number of transmission errors can be detected and corrected on the receiving end. Obviously this means that there is some redundancy in the data transmitted. A widely used application for such a code is the storage of information on CDs where in spite of scratches or even holes, the player can correct the data read from the surface.

Mathematically, codes are frequently represented as subspaces of finite dimensional vector spaces over a finite field. The error-correction capabilities depend on “how close” different elements of the subspace can be, so a good code distributes the elements of the subspace (“code words”) so they are maximally separated in the ambient space. The quality of a code is thus a function of the dimension of the ambient space, the minimal distance between code words, and the size of the finite field in question.

One of the most fruitful ways to define codes is through the use of function fields (or equivalently plane curves) over finite fields. Goppa [Gop81] suggested the following idea:

Fix a global function field k , a set S of rational places and a divisor D with support coprime to S . By evaluating the elements of the Riemann-Roch space $\mathcal{L}(D)$ of D at the elements of S , we define a subspace of $\mathbb{F}_q^n$ and thus a linear code.

Goppa was able to relate the parameters of the code to the geometric invariants of the field and the dimension of D . So to find a good code, i.e. a code that can correct many errors in comparison to its length, we have to find a function field

- with many rational places (as the length of the code depends on that)
- of small genus, as the genus links the dimensions of D , the length of the code and the minimal distance.

Class field theory, as outlined above, allows us to construct a function field as an abelian extension of a fixed base field in terms of the base field where we can predict the invariants. Not surprisingly, several of the best known codes are constructed using class field theoretic methods, with the asymptotically best codes being obtained as towers of class fields.

Here we will demonstrate how to get reasonably good “small” codes using a probabilistic method. The idea is straightforward. We choose a small base field k with a reasonable number of rational places. Next we choose some small degree non-rational places to define a ray class group (and thus a large extension) and consider subgroups generated by some (or all) of the rational places thus guaranteeing that those places will split completely in the extension. Then we compute the exact number of rational places and the genus and see if the field is suitable.

```

> F<w> := GF(4);
> Ft<t> := FunctionField(F);
> Fty<y> := PolynomialRing(Ft);
> k := FunctionField(y^2+y+t^3);

```

$k = \mathbb{F}_4(x)[t]/(y^2 + ty + t^3)$ will act as our base field. The divisor D will be constructed randomly out of places of degree 1, 3 and 4.

```

> lp1 := Places(k, 1);
> lp3 := Places(k, 3) cat Places(k, 4);
> opt := 0;
> while opt lt 3 do
>   l1 := SequenceToSet(lp1);
>   D := 0*lp1[1];
>   for j in [1..Random(#lp1 div 2)] do
>     l := Random(l1);
>     D+:=l;
>     Exclude(~l1, l);
>   end for;
>   D += &+ [ Random(lp3) : j in [1..9-#lp1+#l1]];

```

Now that we have a “random” divisor, we need to define the corresponding ray class group. As the group is likely to be too large, we also need to find a “random” subgroup. Here we search for a subgroup U such that the quotient of the ray class group by it has at most 10 elements. If the group is too large, we simply ignore it.

```

> R, mR := RayClassGroup(D);
> U := sub<R | [ x@@mR : x in l1]>;
> q := quo<R|U>;
> if #q eq 1 then continue; end if;
> while #q gt 10 do
>   U := sub<R|U, Random(R)>;
>   q := quo<R|U>;
> end while;
> if #q eq 1 then continue; end if;

```

Next we compute the invariants of the class field corresponding to R/U . To find our target field, we compute the genus g and the number of rational places n . We accept a field if the ratio n/g improves the previously found largest ratio. We terminate the search if the ratio is at least 3. The last number printed in each line is the degree of the extension, that is the size of R/U .

```

> g := Genus(D, U);
> n := NumberOfPlacesOfDegreeOne(D, U);
> g, n, n/g*1.0, #q;
> if n/g gt opt then
>   opt := n/g; D_opt := D; U_opt := U;
> end if;

```

```
> K := AbelianExtension(D_opt, U_opt);
> Kr := RationalExtensionRepresentation(K);
> L := Places(Kr, 1);
> #L;
16
> Genus(Kr);
5
```

```
> L3 := Places(Kr, 2) cat Places(Kr, 3);
> D := &+ L;
> d := [0 : i in [1..#L]];
> for i in [1..100] do
>   G := &+ [Parent(D)|Random(-1,2)*Random(L3) : j in
[1..Random(7)]];
>   dd := Dimension(G) - Dimension(G-D);
>   if dd eq 0 then continue; end if;
>   C := AGCode(L, G);
>   m := MinimumDistance(C);
>   if d[17-dd] lt m then
>     d[17-dd] := m;
>   end if;
> end for;
> d;
[ 16, 0, 11, 9, 0, 8, 6, 5, 5, 0, 4, 3, 0, 2, 2, 2 ]
```

```
> be := [];
> for d in [0..15] do
>   c, f := BestKnownLinearCode(F, 16, d);
```

```

> be[d+1] := MinimumDistance(c);
> end for;
> be; d;
[ 16, 16, 12, 12, 11, 9, 8, 8, 7, 6, 5, 4, 4, 3, 2, 2 ]
[ 16, 0, 11, 9, 0, 8, 6, 5, 5, 0, 4, 3, 0, 2, 2, 2 ]

```

References

- [Aue99] Roland Auer. *Ray class fields of global function fields with many rational places*. PhD thesis, Oldenburg: Carl von Ossietzky Universität Oldenburg, Fachbereich Mathematik, 1999.
- [Bac90] Eric Bach. Explicit bounds for primality testing and related problems. *Math. Comput.*, 55(191):355–380, 1990.
- [BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. *J. of Sym. Comput.*, 24(3):235–265, 1997.
- [CB03] John Cannon and Wieb Bosma, editors. *Handbook of Magma Functions, V2.10*, Sydney, 2003. The Magma Group. URL:<http://magma.maths.usyd.edu.au/>.
- [CDyDO00] Henri Cohen, Francisco Diaz y Diaz, and Michel Olivier. Construction of tables of quartic number fields. In *Bosma, Wieb (ed.), Algorithmic Number Theory. 4th international symposium. ANTS-IV, Leiden, the Netherlands, July 2-7, 2000. Proceedings. Berlin: Springer. Lect. Notes Comput. Sci. 1838, 257-268*. 2000.
- [Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer, Berlin–Heidelberg–New York, 1993.
- [Coh00] Henri Cohen. *Advanced Topics in Computational Number Theory*, volume 193 of *Graduate Texts in Mathematics*. Springer, Berlin–Heidelberg–New York, 2000.
- [FK03] Claus Fieker and Jürgen Klüners. Minimal discriminants for fields with small Frobenius groups as Galois groups. *J. Number Theory*, 99(2):318–337, 2003.
- [FL94] David Ford and Pascal Letard. Implementing the Round Four maximal order algorithm. *J. Theor. Nombres Bordeaux*, 6(1):39–80, 1994.
- [FP93] David Ford and Michael E. Pohst. The totally real A_6 extension of degree 6 with minimum discriminant. *Exp. Math.*, 2(3):231–232, 1993.
- [FPDH98] David Ford, Michael E. Pohst, Mario Daberkow, and Nasser Haddad. The S_5 extensions of degree 6 with minimum discriminant. *Exp. Math.*, 7(2):121–124, 1998.
- [Fri00] Carsten Friedrichs. *Berechnung von Maximalordnungen über Dedekindringen*. Dissertation, Technische Universität Berlin, 2000. URL:http://www.math.tu-berlin.de/~kant/publications/diss/diss_cf.ps.gz.
- [Gop81] V. D. Goppa. Codes on algebraic curves. *Dokl. Akad. Nauk SSSR*, 259(6):1289–1290, 1981.

- [Heß96] Florian Heß. Zur Klassengruppenberechnung in algebraischen Zahlkörpern. Diplomarbeit, Technische Universität Berlin, 1996. URL:<http://www.math.tu-berlin.de/~kant/publications/diplom/hess.ps.gz>.
- [Heß99] Florian Heß. Zur Divisorenklassengruppenberechnung in globalen Funktionenkörpern. Dissertation, Technische Universität Berlin, 1999. URL:http://www.math.tu-berlin.de/~kant/publications/diss/diss_FH.ps.gz.
- [HPP01] Florian Heß, Sebastian Pauli, and Michael E. Pohst. On the computation of the multiplicative group of residue class rings. *To appear in Math. Comp.*, 2001.
- [Lan94] Serge Lang. *Algebraic Number Theory*, volume 110 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 1994.
- [Oli90] Michel Olivier. Corps sextiques primitifs. (Primitive sextic fields). *Ann. Inst. Fourier*, 40(4):757–767, 1990.
- [Pau01] Sebastian Pauli. *Efficient Enumeration of Extensions of Local Fields with Bounded Discriminant*. PhD thesis, Concordia University, 2001. URL:<http://www.math.tu-berlin.de/~pauli/papers/phd.ps.gz>.
- [PZ89] Michael E. Pohst and Hans Zassenhaus. *Algorithmic Algebraic Number Theory*. Encyclopaedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1989.
- [Ser95] J.-P. Serre. *Local Fields*. Springer, New York, 1995.
- [Sti93] Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer-Verlag, Berlin, 1993.

Index

Numbers written in *italic* refer to the page where the corresponding entry is described, the ones underlined to the definition, the rest to the pages where the entry is used.

Artin	map,	11	embedding	Minkowski	bound,	7
			real,	8	modulus,	8
basis			extension			
integral,		2	relative,	3	number	field
pseudo,		4			relative,	3
			factor	basis,	5	
class	group,	5	field		place,	15
ray,		8	cyclotomic,	2	degree,	19
conductor,		10			finite,	15
			GRH,	7	infinite,	15
divisor,		16	map		ray	class
degree,		19	Artin,	11	relation,	group,
effective,		19	maximal	order	Round2,	8
principal,		18	finite,	15	Round4,	6
						2
						2

