Computing class groups and class fields

A tutorial

B. Allombert

IMB
CNRS/Université de Bordeaux

14/01/2026

Introduction to `bnfinit`

`bnfinit` initializes the class group and the unit group of a number field.

```
? bnf = bnfinit(x^3-11);
? bnf.pol
%21 = x^3-11
? bnf.cyc
%4 = [2]
? bnf.reg
%20 = 5.587206626060907761855413020519975370 1
? bnf.fu
%19 = [Mod(-2*x^2+4*x+1,x^3-11)]
```

Fundamental units

To compute the fundamental units (needed by `bnfunits` and `bnfisprincipal`) use the flag 1.

```
? bnf = bnfinit(x^3+1048583,1);
? sizebyte(bnf.fu)
%5 = 283432
? bnfunits(bnf)
? sizebyte(bnfunits(bnf))
%7 = 8232
? bnfsignunit(bnf)
%8 = Mat(-1)
```

S-units

`bnfunits` can also be used to compute *S*-units:

```
? bnf = bnfinit(x^3-17);
? bnfunits(bnf)
? bnfunits(bnf)[1]
%4 = [[[2,1,0]~,-1;[-2,1,0]~,1;[-1,1,0]~,2;
       [0,1,1]~,-4;[2,-1,1]~,2],Mat([-1,1])]
? id = idealprimedec(bnf,2);
? bnfunits(bnf,id)[1]
%6 = [Mat([[2,1,1]~,1]),Mat([[-2,1,0]~,1]),
      [[2,1,0]~,-1;[-2,1,0]~,1;[-1,1,0]~,2;
       [0,1,1]~,-4;[2,-1,1]~,2],Mat([-1,1])]
```

Finding relations

Let S a finite set of prime ideals that generate the class group, let I an ideal. The basic strategy is to find elements of small T_2 norm in I and try to factor them over S:

$$(\alpha) = \prod \mathfrak{p}_i^{a_i}$$

when this happens, then $\prod \mathfrak{p}_i^{a_i}$ is principal and α is a S-unit.
There are two strategies to chose I.
"small norm relations" : I is a product of at most 2 elements of S.
"random relations": I is a product of at least 3 elements of S with random exponents.
Unfortunately the probability of finding a relation is very hard to estimate.

The Bach constant

The factor basis is chosen as the set of all primes ideal of norm at most $c_1 \log(D)^2$. Increasing c_1 increase the probability of finding relations but increase the number of relations to find and the cost of linear algebra

Imaginary quadratic fields, random rel. (PARI 2.17)

```
? quadclassunit(1-2^127)
*** Bach constant: 1.3340686047266399061
Time factor base: 1
Time subFBquad = Vecsmall([2,11,13,17,23,31,37,41,6
Time powsubFBquad: 1
KC = 642, need 647 relations
Time random rel [#rel/#test = 647/450475]: 7681
#### Tentative class number: 573727530352480587 5
Time be honest: 340
%1 = [573727530352480587 5,[573727530352480587 5],
%    [Qfb(10177,9881,41795515245275924076763118727 5
  ***   last result computed in 10,421 ms.
```

Imaginary quadratic fields, sieving (PARI 2.18)

```
? quadclassunit(1-2^127)
*** Bach constant: 1.3340686047266399061
Time factor base: 2
Time subFBquad = Vecsmall([2,11,13,17,23,31,37,41,6
Time powsubFBquad: 1
KC = 642, need 645 relations
Time MPQS rel [#rel = 645]: 33
Time hnfadd: 1808
#### Tentative class number: 5737275303524805875
Time be honest: 0
%1 = [5737275303524805875,[5737275303524805875],
     [Qfb(10151,8649,41902567101878935999331914027 1
  ***   last result computed in 1,881 ms.
```

The Bach constant

```
? \g1bnf
? bnf = bnfinit(x^3+nextprime(2^72));
PREC = 256
Time nfinit & nfrootsof1: 2
Floating point bnf: R1 = 1, R2 = 1
D = 602120120360326824825289510212630173284571067
LIMC2 = 16275
Time computing Bach constant: 7
Time computing inverse of hR: 0
*** Bach constant: 1.5308321954873111981
```

The tech parameter

`bnfinit` has a an optional third argument: `tech = [c_1, c_2, nrpid, max_fact, idex, usethr]` that allow to tune the algorithm.
`c_1=c_2` is the Bach constant fot the chosen factor basis.

nrpid

`nrpid` can be set to -1 to disable search for small norm relations and only use random relations.

max_fact

`max_fact` is the number of factorisation to try for each ideals. Increasing this value slows down the processing of each ideals but increase the probability of finding relations. The default value is 500. This is the most useful parameter.

idex

When searching for small norm relations we look in ideals of the shape $\mathfrak{p}^e\mathfrak{q}$ with `e=idex`

parallelism

Set `usethr` to 1 to use the parallel implementation. This requires to deactivate early-abort strategies when searching in ideals.

Hilbert class field

To compute a Hilbert class field, we first need to compute the class group.

```
? bnf = bnfinit(a^2-a+50,1);
? bnf.cyc
%53 = [9]
```

The class group is isomorphic to $\mathbb{Z}/9\mathbb{Z}$. We compute a relative defining polynomial for the Hilbert class field with the function `bnrclassfield`.

```
? R = bnrclassfield(bnf)[1]
%54 = x^9 - 24*x^7 + (2*a - 1)*x^6 + 495*x^5
 + (-12*a + 6)*x^4 - 30*x^3 + (18*a - 9)*x^2
 + 18*x + (-2*a + 1)
```

Hilbert class field

Conversely, from an abelian extension, we can recover its corresponding class group `rnfconductor`.

```
? [cond,bnr,subg] = rnfconductor(bnf,R);
? cond
%56 = [[1, 0; 0, 1], []]
? subg
%57 = [9]
```

Here the conductor is trivial, and its norm group is trivial in the class group.

Hilbert class field

We can also ask for an absolute defining polynomial for the Hilbert class field with the optional `flag=2`.

```
? R2 = bnrclassfield(bnf,,2)
%58 = x^18 - 48*x^16 + 1566*x^14 - 23621*x^12
 + 244113*x^10 - 19818*x^8 - 3170*x^6
 + 17427*x^4 - 3258*x^2 + 199
```

Ray class fields

We can also consider class fields with nontrivial conductor.

```
? bnr = bnrinit(bnf,12); bnr.cyc
%60 = [72,2]
```

We can compute in advance the absolute degree, signature and discriminant of the corresponding class field with `bnrdisc`.

```
? [deg,r1,D] = bnrdisc(bnr);
? [deg,r1]
%62 = [288,0]
? D
%63 = 92477896[...538 digits...]84942237696
```

This field is huge!

Ray class fields

For efficiency, the class field is computed as a compositum of several smaller fields.

```
? bnrclassfield(bnr)
%64 = [x^2 - 3, x^8 + (-27*a+24)*x^6
+ (-294*a-3273)*x^4 + (-3*a-3852)*x^2 - 3,
x^9 - 24*x^7 + (2*a-1)*x^6 + 495*x^5
+ (-12*a+6)*x^4 - 30*x^3 + (18*a-9)*x^2
+ 18*x + (-2*a+1)]
```

We can force the computation of a single polynomial with `flag=1`.

```
? bnrclassfield(bnr,,1)
%65 = [x^144+... huge polynomial ...]
```

Ray class fields

We can also compute a subfield of the ray class field by specifying a subgroup.

```
? bnr = bnrinit(bnf,7)
? bnr.cyc
%67 = [54,3]
? bnrclassfield(bnr,3) \\elementary 3-subextension
%68 = [x^3 + 3*x + (14*a - 7),
x^3 + (-1008*a - 651)*x + (-1103067*a - 8072813)]
```

Without the explicit field

Computing a defining polynomial with `bnrclassfield` can be time-consuming, so it is better to compute the relevant information without constructing the field, if possible.

We already saw the use of `bnrdisc`; we can also compute splitting information without the explicit field.

```
? pr41 = idealprimedec(bnf,41)[1];
? bnrisprincipal(bnr,pr41,0)
%70 = [0,0]~
```

The Frobenius at $\mathfrak{p}_{41}$ is trivial: this prime splits completely in the degree 162 extension (which we did not compute).

Ray class fields

Let's do a full example with an HNF ideal and a subgroup given by a matrix.

```
? bnr = bnrinit(bnf,[102709,43512;0,1]);
? bnr.cyc
%72 = [17010, 27]
? bnrclassfield(bnr,[9,3;0,1]) \\subgroup of index
? %73 = [x^9 + (-297*a - 4470)*x^7 + ... ]
```

Modulus with infinite places

If the base field has real places, we can specify the modulus at infinity by providing a list of 0 or 1 of length the number of real embeddings.

```
? bnf=bnfinit(a^2-217,1);
? bnf.cyc
%75 = []
? bnrinit(bnf,1).cyc
%76 = []
? bnrinit(bnf,[1,[1,1]]).cyc
%77 = [2]
```

The field $\mathbb{Q}(\sqrt{217})$ has narrow class number 2.

p-part of the ray class group

`bnfinit` require computing discrete logarithm, which is slow. Often, one only need the group $\mathcal{C}\ell_f(K)/\mathcal{C}\ell_f(K)^n$ for some small n which is much easier to compute.

```
? bnr = bnrinit(bnf,[102709,43512;0,1],,3);
? bnr.cyc;
%78 = [3,3]
? bnrclassfield(bnr)
%7 = [x^3+(129*a+597)*x+(-575*a-5931),x^3+(2079*a-
```

Multiple subgroups at once

Since `bnrclassfield` can involve costly computations of `bnf`, It is possible to specify several subgroups at once .

```
? S = [s|s<-subgrouplist(bnr,,1)|matdet(s)==3];
? bnrclassfield(bnr,S)
? bnrclassfield(bnr,S,1)
%25 = [x^3+(81*a+2046)*x+(5233*a-2630),
%       x^3+(81*a+2046)*x+(-9607*a-107854),
%       x^3+(2079*a-3957)*x+(-90203*a-509303),
%       x^3+(129*a+597)*x+(-575*a-5931)]
```