Algebraic number theory with GP

B. Allombert and A. Page

IMB/INRIA/Université de Bordeaux

04/12/2025

Documentation

- ▶ refcard-nf.pdf: list of functions with a short description.
- ▶ users.pdf Section 3.13: introduction and detailed descriptions of the functions.
- ▶ in gp, `?10`: list of functions.
- ▶ in gp, `?functionname`: short description of the function.
- ▶ in gp, `??functionname`: long description of the function.

To record the commands we will type during the tutorial:

```
? \l TAN.log
```

Plan

Number fields

Ideals

Class groups and units

Number Fields

Reminder

A **number field** K is a finite extension of $\mathbb{Q}$.

The **ring of integers** $\mathbb{Z}_K$ of K is the set of elements of K whose monic minimal polynomial has coefficients in $\mathbb{Z}$.
It is free of rank $[K : \mathbb{Q}]$ over $\mathbb{Z}$.

The **discriminant** Δ_K of K is the determinant of the matrix $(\mathrm{Tr}(w_i w_j))_{i,j}$ where $w_1, \ldots, w_n$ is a $\mathbb{Z}$-basis of $\mathbb{Z}_K$.

Irreducibility

In GP, we describe a number field *K* as

$$K = \mathbb{Q}[x]/f(x)$$

where $f \in \mathbb{Z}[x]$ is a monic irreducible polynomial.

```
? f = x^4 - 2*x^3 + x^2 - 5;
? polisirreducible(f)
%2 = 1
```

GP knows cyclotomic polynomials:

```
? g = polcyclo(30)
%3 = x^8 + x^7 - x^5 - x^4 - x^3 + x + 1
```

Algebraic numbers

To perform simple operations in $K = \mathbb{Q}[x]/f(x) = \mathbb{Q}(\alpha)$ where $f(\alpha) = 0$, we can use Mod:

```
? Mod(x,f)^5
%4 = Mod(3*x^3-2*x^2+5*x+10, x^4-2*x^3+x^2-5)
```

Interpretation: $\alpha^5 = 3\alpha^3 - 2\alpha^2 + 5\alpha + 10$.

We check that the roots of *g* are 30th roots of unity:

```
? lift(Mod(x,g)^15)
%5 = -1
```

We used lift to make the output more readable.

polredbest

Sometimes we can find a simpler defining polynomial for the
same number field by using `polredbest`:

```
? {h = x^5 + 7*x^4 + 22550*x^3 - 281686*x^2
  - 85911*x + 3821551};
? polredbest(h)
%7 = x^5 - x^3 - 2*x^2 + 1
```

Interpretation: $\mathbb{Q}[x]/h(x) \cong \mathbb{Q}[x]/(x^5 - x^3 - 2x^2 + 1)$.

nfinit

Most operations on number fields use a structure, which is computed by the initialisation function `nfinit`.

```
? K = nfinit(f);
```

`K` contains the structure for the number field $K = \mathbb{Q}[x]/f(x)$.

```
? K.pol
%9 = x^4 - 2*x^3 + x^2 - 5
? K.sign
%10 = [2, 1]
```

K has signature $(2, 1)$: it has two real embeddings and one pair of conjugate complex embeddings.

Computed information

```
? K.disc
%11 = -1975
? K.zk
%12 = [1,1/2*x^2-1/2*x-1/2,x,1/2*x^3-1/2*x^2-1/2*x]
? w = K.zk[2];
```

K has discriminant -1975, and its ring of integers is

$$\mathbb{Z}_K = \mathbb{Z} + \mathbb{Z}\frac{\alpha^2 - \alpha - 1}{2} + \mathbb{Z}\alpha + \mathbb{Z}\frac{\alpha^3 - \alpha^2 - \alpha}{2} = \mathbb{Z} + \mathbb{Z}w + \mathbb{Z}\alpha + \mathbb{Z}w\alpha.$$

Elements of a number field

We saw that we could represent elements of a number field as polynomials in α. We can also use linear combinations of the integral basis. We can switch between the two representations with nfalgtobasis and nfbasistoalg.

```
? nfalgtobasis(K,x^2)
%14 = [1, 2, 1, 0]~
```

Interpretation: $\alpha^2 = 1 \cdot 1 + 2 \cdot w + 1 \cdot \alpha + 0 \cdot w\alpha = 1 + 2w + \alpha$.

```
? nfbasistoalg(K,[1,1,1,1]~)
%15 = Mod(1/2*x^3 + 1/2, x^4 - 2*x^3 + x^2 - 5)
```

Interpretation: $1 + w + \alpha + w\alpha = \frac{\alpha^3+1}{2}$.

Elements of a number field: operations

We perform operations on elements with the functions
`nfeltxxxx`, which accept both representations as input.

```
? nfeltmul(K,[1,-1,0,0]~,x^2)
%16 = [-1, 3, 1, -1]~
```

Interpretation: $(1 - w) \cdot \alpha^2 = -1 + 3w + \alpha - w\alpha$.

```
? nfeltnorm(K,x-2)
%17 = -1
? nfelttrace(K,[0,1,2,0]~)
%18 = 2
```

Interpretation: $N_{K/\mathbb{Q}}(\alpha - 2) = -1$, $\mathrm{Tr}_{K/\mathbb{Q}}(w + 2\alpha) = 2$.

Ideals

Reminder

In $\mathbb{Z}_K$, ideals factor uniquely into products of prime ideals:

$$\mathfrak{a} = \prod_i \mathfrak{p}_i^{a_i}.$$

In particular, prime numbers admit a decomposition:

$$p\mathbb{Z}_K = \prod_i \mathfrak{p}_i^{e_i} \text{ with } \mathbb{Z}_K/\mathfrak{p}_i \cong \mathbb{F}_{p^{f_i}}.$$

▶ e_i = ramification index of $\mathfrak{p}_i$.
▶ f_i = residue degree of $\mathfrak{p}_i$.

Decomposition of primes

We can decompose primes with `idealprimedec`:

```
? dec = idealprimedec(K,5);
? #dec
%20 = 2
? [pr1,pr2] = dec;
```

Interpretation: $\mathbb{Z}_K$ has two prime ideals above 5, which we call $\mathfrak{p}_1$ and $\mathfrak{p}_2$.

```
? pr1.f
%22 = 1
? pr1.e
%23 = 2
```

$\mathfrak{p}_1$ has residue degree 1 and ramification index 2.

Decomposition of primes

```
? pr1.gen
%24 = [5, [-1, 0, 1, 0]~]
```

$\mathfrak{p}_1$ is generated by 5 and $-1 + 0 \cdot w + \alpha + 0 \cdot w\alpha$, i.e. we have $\mathfrak{p}_1 = 5\mathbb{Z}_K + (\alpha - 1)\mathbb{Z}_K$.

```
? pr2.f
%25 = 1
? pr2.e
%26 = 2
```

$\mathfrak{p}_2$ also has residue degree 1 and ramification index 2.

Ideals

An arbitrary ideal is represented by its Hermite normal form
(HNF) with respect to the integral basis. We can obtain this
form with `idealhnf`.

```
? idealhnf(K,pr1)
%27 =
[5 3 4 3]
[0 1 0 0]
[0 0 1 0]
[0 0 0 1]
```

Interpretation: $\mathfrak{p}_1$ can be described as

$$\mathfrak{p}_1 = \mathbb{Z} \cdot 5 + \mathbb{Z} \cdot (w + 3) + \mathbb{Z} \cdot (\alpha + 4) + \mathbb{Z} \cdot (w\alpha + 3).$$

Ideals

```
? a = idealhnf(K,[23, 10, -5, 1]~)
%28 =
[260   0 228 123]
[  0 260 123 105]
[  0   0   1   0]
[  0   0   0   1]
```

We obtain the HNF of the ideal $\mathfrak{a} = (23 + 10w - 5\alpha + w\alpha)$.

```
? idealnorm(K,a)
%29 = 67600
```

We have $N(\mathfrak{a}) = 67600$.

Ideals: operations

We perform operations on ideals with the functions
`idealxxxx`, which accept HNF forms, prime ideal structures
(output of `idealprimedec`), and elements (interpreted as
principal ideals).

```
? idealpow(K,pr2,3)
%30 =
[25 15 21 7]
[ 0  5  2 4]
[ 0  0  1 0]
[ 0  0  0 1]
? idealnorm(K,idealadd(K,a,pr2))
%31 = 1
```

We have $\mathfrak{a} + \mathfrak{p}_2 = \mathbb{Z}_K$: the ideals $\mathfrak{a}$ and $\mathfrak{p}_2$ are coprime.

Ideals: factorisation

We factor an ideal into a product of prime ideals
with idealfactor. The result is a two-column matrix: the first
column contains the prime ideals, and the second one contains
the exponents.

```
? fa = idealfactor(K,a);
? matsize(fa)
%33 = [3,2]
```

The ideal a is divisible by three prime ideals.

```
? [fa[1,1].p, fa[1,1].f, fa[1,1].e, fa[1,2]]
%34 = [2, 2, 1, 2]
```

The first one is a prime ideal above 2, is unramified with
residue degree 2, and appears with exponent 2.

Ideals: factorisation

```
? [fa[2,1].p, fa[2,1].f, fa[2,1].e, fa[2,2]]
%35 = [5, 1, 2, 2]
? fa[2,1]==pr1
%36 = 1
```

The second one is $\mathfrak{p}_1$, and it appears with exponent 2.

```
? [fa[3,1].p, fa[3,1].f, fa[3,1].e, fa[3,2]]
%37 = [13, 2, 1, 1]
```

The third one is a prime ideal above 13, is unramified with residue degree 2, and appears with exponent 2.

Chinese remainders

We can use the Chinese remainder theorem with
`idealchinese`:

```
? b = idealchinese(K,[pr1,2;pr2,1],[1,-1]);
```

We are looking for an element $b \in \mathbb{Z}_K$ such that $b = 1 \bmod \mathfrak{p}_1^2$
and $b = -1 \bmod \mathfrak{p}_2$.

```
? nfeltval(K,b-1,pr1)
%39 = 2
? nfeltval(K,b+1,pr2)
%40 = 1
```

We check the output by computing valuations: $v_{\mathfrak{p}_1}(b-1) = 2$
and $v_{\mathfrak{p}_2}(b+1) = 1$.

Chinese remainders with signs

We can compute the sign of real embeddings of *b*:

```
? nfeltsign(K,b)
%41 = [-1, 1]
```

We have $\sigma_1(b) < 0$ and $\sigma_2(b) > 0$, where σ_1, σ_2 are the two real embeddings of *K*.

We can ask idealchinese to compute an element that, in addition to the congruences, is totally positive:

```
? c = idealchinese(K,[[pr1,2;pr2,1],[1,1]],[1,-1]);
? nfeltsign(K,c)
%43 = [1, 1]
```

Indeed we have $\sigma_1(c) > 0$ and $\sigma_2(c) > 0$.

Class groups and units

Reminder

The class group

$$\mathrm{Cl}(K) = \frac{(\text{nonzero ideals of } K)}{(\text{principal ideals } \beta \mathbb{Z}_K)}.$$

is a finite abelian group.

The unit group

$$\mathbb{Z}_K^\times \cong \mathbb{Z}/w\mathbb{Z} \times \mathbb{Z}^{r_1+r_2-1}$$

is a lattice under the logarithmic embedding, whose covolume is called the regulator Reg_K.

bnfinit

To obtain the class group and unit group of a number field, we need a more expensive computation than `nfinit`. The relevant information is contained in the structure computed with `bnfinit` (b = Buchmann).

```
? K2 = bnfinit(K);
? K2.nf == K
%50 = 1
? K2.no
%51 = 1
```

K has a trivial class group (no = class number).

```
? K2.reg
%52 = 1.7763300299706546701307646106399605586
```

We obtain an approximation of the regulator of *K*.

bnfcertify

The output of `bnfisprincipal` is a priori only correct under GRH (Generalised Riemann Hypothesis). We can unconditionally certify it with `bnfcertify`.

```
? bnfcertify(K2)
%52 = 1
```

The computation is now certified! If `bnfcertify` outputs 0, it means we have found a counter-example to GRH (or more likely a bug in PARI/GP)!

bnfinit: units

```
? lift(K2.tu)
%54 = [2, -1]
? K2.tu[1]==nfrootsof1(K)[1]
%55 = 1
```

K has two roots of unity (tu = torsion units), ± 1. We can also compute them with `nfrootsof1`.

```
? lift(K2.fu)
%56 = [1/2*x^2-1/2*x-1/2, 1/2*x^3-3/2*x^2+3/2*x-1]
```

The free part of $\mathbb{Z}_K^\times$ is generated by $\frac{\alpha^2-\alpha-1}{2}$ and $\frac{\alpha^3-3\alpha^2+3\alpha-2}{2}$ (fu = fundamental units).

bnfinit: analytic class number formula

```
? lfun(K,1+x+O(x^2))
%57 = 0.50228472605280111386617636 5679645651*x^-1
  + O(x^0)
? res = polcoeff(lfun(K,1+x+O(x^2)),-1)
%58 = 0.50228472605280111386617636 5567964565169
```

We compute an approximation of the residue of $\zeta_K(s)$ at $s = 1$.

```
? {2^K2.r1*(2*Pi)^K2.r2*K2.no*K2.reg/
    (K2.tu[1]*sqrt(abs(K2.disc))*res)}
%59 = 0.99999999999999999999999999999999999999999999999
```

We numerically check the analytic class number formula.

Class group

```
? L = bnfinit(x^3 - x^2 - 54*x + 169);
? L.cyc
%61 = [2, 2]
```

$Cl(L) \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.

```
? L.gen
%62 = [[5,3,2;0,1,0;0,0,1], [5,4,3;0,1,0;0,0,1]]
```

Generators of the class group, given as ideals in HNF form.

Testing whether an ideal is principal

We test whether an ideal is principal with bnfisprincipal:

```
? pr = idealprimedec(L,13)[1]
? [dl,g] = bnfisprincipal(L,pr);
? dl
%65 = [1, 0]~
```

bnfisprincipal expresses the class of the ideal in terms of
the generators of the class group (discrete logarithm). Here,
the ideal pr is in the same class as the first generator. In
particular, the ideal is not principal, but its square is.

Testing whether an ideal is principal

```
? g
%66 = [0, 1/5, 2/5]~
? {idealhnf(L,pr) == idealmul(L,g,
    idealfactorback(L,L.gen,dl))}
%67 = 1
```

The second component of the output of bnfisprincipal is
an element $g \in L$ that generates the remaining principal ideal.
(idealfactorback = inverse of idealfactor = $\prod_i$ L.gen[i]$^{\text{dl[i]}}$)

Computing a generator of principal ideal

We know that pr is a 2-torsion element; let's compute a generator of its square:

```
? [dl2,g2] = bnfisprincipal(L,idealpow(L,pr,2));
? dl2
%69 = [0, 0]~
```

The ideal is indeed principal (trivial in the class group).

```
? g2
%70 = [1, -1, -1]~
? idealhnf(L,g2) == idealpow(L,pr,2)
%71 = 1
```

$g2$ is a generator of pr^2.

Application: bnfisintnorm

We can use these functionalities to find solutions in $\mathbb{Z}_K$ of norm equations with bnfisintnorm:

```
? bnfisintnorm(L,5)
%72 = []
```

There is no element of norm 5 in $\mathbb{Z}_L$.

```
? bnfisintnorm(L,65)
%73 = [x^2 + 4*x - 36, -x^2 - 3*x + 39, -x + 2]
```

There are three elements of $\mathbb{Z}_L$ of norm 65, up to multiplication by elements of $\mathbb{Z}_L^\times$ with positive norm.

Expressing a unit in terms of the generators

```
? u = [0,2,1]~;
? nfeltnorm(L,u)
%75 = 1
```

We have found a unit $u \in \mathbb{Z}_L^\times$.

```
? bnfisunit(L,u)
%76 = [1, 2, Mod(0, 2)]~
? lift(L.fu)
%77 = [x^2 + 4*x - 34, x - 4]
? lift(L.tu)
%78 = [2, -1]
```

We express it in terms of the generators with bnfisunit:
$u = (\alpha^2 + 4\alpha - 34) \cdot (\alpha - 4)^2 \cdot (-1)^0.$

Large fundamental units

By default, bnfinit only computes fundamental units if they are not too large.

```
? M = bnfinit(x^2-3019);
? M.fu
%80 = 0
```

We can force the computation of fundamental units with bnfinit(,1).

```
? M = bnfinit(x^2-3019,1);
? lift(M.fu)
%82 = [21389518805375209854607105559272556570669
   0871236169789*x - 11752562541659941018442526415
   375394603920948258603143300]
```

Questions ?

Have fun with GP!