Algebraic number theory with GP

B. Allombert and A. Page

COGENT Summer School

15/06/2022

Documentation

- refcard-nf.pdf: list of functions with a short description.
- users.pdf Section 3.10: introductory paragraph and detailed description of all functions.
- in gp, `?10`: function list.
- in gp, `?functionname`: short description of the function.
- in gp, `??functionname`: long description of the function.
- in gp, `???string`: search string in doc.

To record your commands during the tutorial:

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

Plan

1. Number fields
2. Elements and ideals
3. Class groups and units

Number fields

Irreducibility

In GP, a number field K is described 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)
% = 1
```

GP knows cyclotomic polynomials :

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

Polmod

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

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

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

The roots of g are indeed 30th roots of unity.
We used `lift` for readability.

Compositum

If we want to represent a field such as

$$\mathbb{Q}(\sqrt{2}, \sqrt{3}),$$

we need to construct it as a compositum.

```
? polcompositum(x^2-2,x^2-3)
% = [x^4 - 10*x^2 + 1]
```

The output is a vector of polynomials since in general there might be several compositums.

```
? polcompositum(x^4-2, x^4-2*x^2-1)
% = [x^8 - 4*x^6 - 26*x^4 - 4*x^2 + 1,
    x^8 - 4*x^6 + 22*x^4 - 36*x^2 + 49]
```

Compositum

We may even adjoin several roots of the same polynomial !

```
? L = polcompositum(x^3-2, x^3-2)
% = [x^3 + 2, x^6 + 40*x^3 + 1372]
```

The first polynomial corresponds to the field obtained by adjoining one root, the second one by adjoining two roots.

```
? nfrootsof1(L[2])
% = [6, -1/36*x^3 - 1/18]
```

Consistency check: the second field contains a nontrivial 3rd root of unity!

Isomorphism and inclusion

We can ask whether two number fields are isomorphic:

```
? nfisisom(x^2-5, x^2-2)
% = 0
? nfisisom(x^2-5, x^2-x-1)
% = [-2*x + 1, 2*x - 1]
```

The function returns the set of all isomorphisms.
We can also ask whether a field can be included in another:

```
? nfisincl(x^2-5,polcyclo(5))
% = [-2*x^3 - 2*x^2 - 1, 2*x^3 + 2*x^2 + 1]
? nfisincl(x^2+5,polcyclo(5))
% = 0
```

Subfields

We can compute the subfields of a given field:

```
? nfsubfields(x^6 + 40*x^3 + 1372,,1)
% = [x, x^2 + 120*x + 12348, x^3 - 128, x^3 + 250,
    x^3 - 2, x^6 + 40*x^3 + 1372]
```

The option "`(...,,1)`" suppresses the inclusion data.
We can ask for a specific degree:

```
? nfsubfields(x^8 + 3*x^4 + 9, 2)
% = [[x^2 + 36, 2/3*x^6 + 4*x^2],
     [x^2 - 12, -2/3*x^6],
     [x^2 + 12*x + 144, 4*x^4]]
```

For each subfield, the second polynomial returned represents a root of the quadratic polynomial in the top field.

Subfields

We can ask for the maximal subfields:

```
? nfsubfieldsmax(x^8-4*x^5+7*x^4-x^2+x+1, 1)
% = [x^2 + 197*x - 199, x^4 - 10*x^2 - 37*x + 121]
```

They do not necessarily have all the same degree.

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)
% = 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 require having computed the ring of integers, which is done by the initialisation function `nfinit` (nf = number field).

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

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

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

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

Computed information

```
? K.disc
% = -1975
? K.zk
% = [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

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

Factorisation of polynomials over a number field

We can factor polynomials with coefficients in a number fields. For this, we must be careful that the variable of the polynomial has higher priority than the one used for the number field.

```
? z = varhigher("z");
? nffactor(K, subst(K.pol,x,z))
% =
[           z + Mod(-x, x^4 - 2*x^3 + x^2 - 5) 1]
[        z + Mod(x - 1, x^4 - 2*x^3 + x^2 - 5) 1]
[z^2 - z + Mod(x^2 - x, x^4 - 2*x^3 + x^2 - 5) 1]
```

The result is a two-column matrix; the first contains the irreducible divisors, and the second contains the exponents.

Roots of polynomials over a number field

We can also simply ask for the roots.

```
? lift(nfroots(K, subst(K.pol,x,z)))
% = [-x + 1, x]
```

We see that K has an automorphism given by $\alpha \mapsto 1 - \alpha$.

Elements and ideals

Elements of a number field

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

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

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

Element operations

The operations on elements are the functions `nfeltxxxx`, and they accept both representations as input.

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

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

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

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

Characteristic and minimal polynomials

We compute the characteristic and minimal polynomial in algebraic form.

```
? charpoly(nfbasistoalg(K,[1,2,0,0]~))
% = x^4 - 10*x^2 + 25
? minpoly(nfbasistoalg(K,[1,2,0,0]~))
% = x^2 - 5
```

The minimal and characteristic polynomials will be the same unless the element lies in a proper subfield.

Embeddings

We can compute the real and complex embeddings of an element with `nfeltembed`.

```
? nfeltembed(K,x^3+x)
% = [-2.3250207137883080622303986499385818825,
    11.033224646287677151457919656132410589,
    -2.3541019662496845446137605030969143532
    - 0.33268570002014959478470322160341519810*I]
```

Decomposition of primes

We decompose a prime number with `idealprimedec`:

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

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

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

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

Decomposition of primes

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

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

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

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

Reducing modulo a prime

We reduce elements modulo prime ideals with `nfmodpr`.

```
? p11 = idealprimedec(K,11)[1]; p11.f
% = 2
? modpr = nfmodprinit(K,p11,v);
? a = nfmodpr(K,x^2+x+1,modpr)
% = 2*v + 5
? a^11
% = 9*v + 7
? a^121
% = 2*v + 5
```

Conversely we can compute lifts with `nfmodprlift`.

```
? nfmodprlift(K,a,modpr)
% = 2*x + 5
```

Ideals

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

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

Interpretation : $\mathfrak{p}_1$ equals

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

Ideals

```
? a = idealhnf(K,[23, 10, -5, 1]~)
% =
[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)
% = 67600
```

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

Ideals: operations

Operations on ideals are the functions `idealxxxx` and accept HNF forms, prime ideal structures (output of `idealprimedec`), and elements (representing principal ideals).

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

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

Two generators representation

Every ideal can be generated by two elements. We compute this representation with `idealtwoelt`.

```
? [n,b] = idealtwoelt(K,a)
% = [260, [-32, 123, 1, 0]~]
? idealadd(K,n,b) == a
```

We check that we do get a correct representation.

Operations related to prime ideals

We compute the valuation of an ideal at a prime with `idealval`.

```
? idealval(K,a,pr2)
% = 0
```

The ideal $\mathfrak{a}$ is not divisible by $\mathfrak{p}_2$.
We can test whether an ideal is prime with `idealismaximal`.

```
? idealismaximal(K,a)
% = 0
? idealismaximal(K,idealhnf(K,pr1)) != 0
% = 1
```

If the ideal is prime, the function returns a prime ideal structure, like `idealprimedec`.

Ideals: factorisation

We factor an ideal into primes with `idealfactor`. The result is a two-column matrix, the first containing the prime ideals and the second containing the exponents.

```
? fa = idealfactor(K,a);
? #fa[,1]
% = 3
```

The ideal $\mathfrak{a}$ is divisible by three prime ideals.

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

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

Ideals: factorisation

```
? [fa[2,1].p, fa[2,1].f, fa[2,1].e, fa[2,2]]
% = [5, 1, 2, 2]
? fa[2,1]==pr1
% = 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]]
% = [13, 2, 1, 1]
```

The third one is a prime ideal above 13, of residue degree 2 and unramified, and appears with exponent 1.

Chinese remainder theorem

We may apply the Chinese remainder theorem with `idealchinese`:

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

We are asking 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)
% = 2
? nfeltval(K,b+1,pr2)
% = 1
```

We check the result 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 the real embeddings of b:

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

We indeed have $\sigma_1(c) > 0$ and $\sigma_2(c) > 0$.

Class groups and units

bnfinit

To compute the class group and units of a number field, we need a more expensive computation than `nfinit`. This computation is done by `bnfinit` (b = Buchmann).

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

K is a PID (no = class number).

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

We obtain an approximation of the regulator of K.

bnfinit : units

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

K has two roots of unity (tu = torsion units), ± 1.

```
? lift(K2.fu)
% = [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).

Class group

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

$\mathcal{C}\ell(L) \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$

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

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

Testing whether an ideal is principal

We can test whether an ideal is principal with `bnfisprincipal`:

```
? pr = idealprimedec(L,13)[1]
? [dl,g] = bnfisprincipal(L,pr);
? dl
% = [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, it is not principal, but its square is.

Testing whether an ideal is principal

```
? g
% = [-2/5, 1/5, 0]~
? {idealhnf(L,pr) == idealmul(L,g,
    idealfactorback(L,L.gen,dl))}
% = 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 = $\prod_i \texttt{L.gen[i]}^{\texttt{dl[i]}}$)

Computing a generator of a principal ideal

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

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

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

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

`g2` is a generator of $\mathtt{pr}^2$.

Expressing a unit in terms of the generators

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

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

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

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

Large fundamental units

By default, `bnfinit` only computes fundamental units if they are small.

```
? M = bnfinit(x^2-3019);
? M.fu
% = 0 \\sentinel value: not computed
```

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

```
? M = bnfinit(x^2-3019,1);
? lift(M.fu)
% = [2138951880537520985460710555927255657066690
  871236169789*x - 117525625416599410184425264152
  37539460392094825860314330]
```

Very large fundamental units

Sometimes, the fundamental units are so large that it's not a good idea to write them in terms of the basis. Instead we should keep them as a product of small elements.

```
? D = 10^9 + 1273;
? N = bnfinit(x^2-D,1);
? bnfunits(N)
% = [[[37, -723420; 43, 1884873; 53, -7850; ...
? N.fu
% = ... \\very large
```

Very large ideal generators

Similarly, the generators of principal ideals can be very large.
We can also ask for a compact representation.

```
? bnfisprincipal(N,P)
  *** bnfisprincipal: Warning: precision too low
  for generators, not given.
% = [[]~, []~]
? bnfisprincipal(N,P,4)
% = [[]~, [37, 249358; 43, -581068; ... ]]
? bnfisprincipal(N,P,3)
% = [[]~, [... very large generator ...]~]
```

Multiplicative functions also accept elements in factored form.

S-units

We can add an argument to `bnfunits` to compute *S*-units instead.

```
? P = idealprimedec(N,2)[1];
? S = idealprimedec(N,2);
? U = bnfunits(N,S);
? #U[1]
% = 4
```

Like for the units, we can write an arbitrary *S*-unit in terms of the generators with `bnfisunit`.

```
? bnfisunit(N,2)
% = []~
? bnfisunit(N,2,U)
% = [1, 1, 0, 1]~
```

Questions ?

Have fun with GP !