Théorie algébrique des nombres avancée

A. Page

IMB
Inria / Université de Bordeaux

24/11/2022

polgalois

On peut calculer le groupe de Galois de la clôture galoisienne d'un corps de nombres, comme groupe de permutations. Restreint aux degrés ≤ 7, ou degrés ≤ 11 avec le paquet optionnel `galdata`.

```
P1 = x^4-5;
polgalois(P1)
% = [8, -1, 1, "D(4)"]
```

Interprétation : le groupe de Galois est d'ordre 8, n'est pas contenu dans le groupe alterné (« signature -1 »), et est isomorphe à D_4.

polgalois

```
P2 = x^4-x^3-7*x^2+2*x+9;
polgalois(P2)
% = [12, 1, 1, "A4"]
```

Le groupe de Galois est d'ordre 12, signature 1, et est isomorphe à A_4.

```
P3 = x^4-x^3-3*x^2+x-1;
polgalois(P3)
% = [24, -1, 1, "S4"]
```

Le groupe de Galois est d'ordre 24 et signature -1, et est isomorphe à S_4.

nfsplitting

On peut calculer un polynôme définissant le corps de décomposition d'un polynôme donné, c'est-à-dire le plus petit corps sur lequel le polynôme donné est un produit de facteurs linéaires.

```
Q1 = nfsplitting(P1)
% = x^8 + 70*x^4 + 15625
Q2 = nfsplitting(P2)
% = x^12 - 59*x^10 + 1269*x^8 - 12231*x^6
  + 51997*x^4 - 79707*x^2 + 26569
```

De manière équivalente, c'est un polynôme de définition de la clôture galoisienne du corps de nombre engendré par une racine du polynôme donné.

nfsplitting

Le polynôme calculé par `nfsplitting` peut être gros.

```
Q3 = nfsplitting(P3)
% = x^24+12*x^23-66*x^22-1232*x^21+735*x^20
 +54012*x^19+51764*x^18-1348092*x^17-2201841*x^16
 +21708244*x^15+41344014*x^14-241723272*x^13
 -454688929*x^12+1972336584*x^11+3130578366*x^10
 -12348327032*x^9-13356023346*x^8+59757161004*x^7
 +32173517686*x^6-204540935496*x^5-11176476888*x^4
 +433089193668*x^3-155456858376*x^2-422808875280*x
 +320938557273
```

polredbest

On peut utiliser `polredbest` pour calculer un polynôme plus simple qui définit le même corps de nombres.

```
Q3 = polredbest(Q3)
% = x^24-6*x^23+18*x^22-38*x^21+60*x^20-54*x^19
 -13*x^18+126*x^17-228*x^16+220*x^15+24*x^14
 -396*x^13+521*x^12-216*x^11-48*x^10-32*x^9-66*x^8
 +666*x^7-1013*x^6+348*x^5+510*x^4-654*x^3+234*x^2
 +36*x+9
```

galoisinit

On peut calculer le groupe des automorphismes d'un corps de nombres qui est galoisien sur $\mathbb{Q}$ avec `galoisinit`, sous certaines conditions (groupe « faiblement super-résoluble »).

```
gal = galoisinit(Q3);
```

La composante `gen` est une liste de générateurs du groupe d'automorphisme, exprimés comme permutations des racines.

```
gal.gen
% = [Vecsmall([19,11,17,14,13,12,10,9,8,7,2,6,5,
 4,23,22,3,21,1,24,18,16,15,20]),Vecsmall([14,10,5,
 19,3,24,11,16,22,2,7,20,17,1,21,8,13,23,4,12,15,9,
 18,6]),Vecsmall([5,15,6,13,20,19,23,7,11,18,21,4,
 12,17,16,2,24,22,3,1,9,10,8,14]),Vecsmall([2,1,9,
 10,16,21,14,17,3,4,19,18,22,7,20,5,8,12,11,15,6,
 13,24,23])]
```

galoisinit

La composante `orders` contient les ordres de facteurs de composition du groupe, et leur produit est l'ordre du groupe.

```
ord = gal.orders
% = Vecsmall([2, 2, 3, 2])
prod(i=1,#ord,ord[i])
% = 24
```

On peut obtenir l'identifiant GAP4 du groupe avec `galoisidentify`.

```
galoisidentify(gal)
% = [24, 12]
```

Théorie de Galois effective

`galoissubgroups` calcule la liste de tous les sous-groupes d'un groupe.

```
L = galoissubgroups(gal);
#L
% = 30
```

On peut ensuite calculer les corps fixés par différents sous-groupes du groupe de Galois avec `galoisfixedfield`.

```
R1 = galoisfixedfield(gal,L[25])[1];
polgalois(R1)
% = [24, 1, 1, "S_4(6d) = [2^2]S(3)"]
R2 = galoisfixedfield(gal,L[28])[1];
polgalois(R2)
% = [24, -1, 1, "S_4(6c) = 1/2[2^3]S(3)"]
```

galoissplittinginit

On peut remplacer `nfsplitting` suivi de `galoisinit` par `galoissplittinginit`, qui est plus rapide et n'a pas de restriction sur le groupe.

```
P = x^5+20*x+16;
polgalois(P)
% = [60, 1, 4, "A5"]
```

Le polynôme a pour groupe de Galois A_5.

```
G = galoissplittinginit(P);
G.pol == nfsplitting(P)
% = 1
```

Le corps de décomposition est correct.

galoissplittinginit

On vérifie que le groupe de Galois est bien A_5.

```
galoisidentify(G)
% = [60, 5]
```

On peut calculer des sous-corps fixés comme précédemment.

```
galoisfixedfield(G,[G.group[2],G.group[6]],1)
% = x^6 - 1600*x^4 + 1536000*x^2 + 32768000*x
    + 163840000
```

Groupes de ramification

On peut calculer des groupes de ramification. Calculons d'abord les premiers ramifiés.

```
nf = nfinit(Q3);
factor(nf.disc)
% =
[ 3 28]
[11 16]
```

Les premiers ramifiés sont 3 et 11.

```
dec3 = idealprimedec(nf,3);
pr3 = dec3[1];
[#dec3, pr3.f, pr3.e]
% = [4, 1, 6]
```

Il y a 4 idéaux premiers au-dessus de 3. Ils sont de degré résiduel 1 et d'indice de ramification 6.

Groupes de ramification

On calcule la suite des groupes de ramification avec `idealramgroups`.

```
ram3 = idealramgroups(nf,gal,pr3);
#ram3
% = 3
```

Il y a trois groupes de ramification non-triviaux à considérer.

```
galoisidentify(ram3[1])
% = [6, 1]
galoisisabelian(ram3[1])
% = 0
```

Le groupe de décomposition est d'ordre 6, isomorphe à S_3.

Groupes de ramification

```
galoisidentify(ram3[2])
% = [6, 1]
```

Le groupe d'inertie est égal au groupe de décomposition (ce qu'on savait déjà car le degré résiduel est 1).

```
galoisidentify(ram3[3])
% = [3, 1]
```

Le groupe d'inertie sauvage est le groupe cyclique C_3, et tous les groupes de ramification supérieurs sont triviaux.

Groupes de ramification

```
dec11 = idealprimedec(nf,11);
pr11 = dec11[1];
[#dec11, pr11.f, pr11.e]
% = [4, 2, 3]
```

Il y a 4 idéaux premiers au-dessus de 11. Ils sont de degré résiduel 2 et d'indice de ramification 3.

```
ram11 = idealramgroups(nf,gal,pr11);
#ram11
% = 2
```

Le groupe d'inertie sauvage est trivial (ce qu'on savait déjà puisque 11 est premier à l'ordre du groupe).

Groupes de ramification

```
galoisidentify(ram11[1])
% = [6, 1]
galoisidentify(ram11[2])
% = [3, 1]
```

Le groupe de décomposition est isomorphe à S_3 (on savait déjà qu'il était d'indice 4 dans le groupe de Galois), et le groupe d'inertie est C_3 (on savait déjà qu'il était d'indice 2 dans le groupe de décomposition).

Élements de Frobenius

En un premier non ramifié, on peut calculer l'élément de Frobénius avec `idealfrobenius`.

```
dec2 = idealprimedec(nf,2);
pr2 = dec2[1];
[#dec2, pr2.f, pr2.e]
% = [6, 4, 1]
frob2 = idealfrobenius(nf,gal,pr2);
permorder(frob2)
% = 4
```

On vérifie que l'élément de Frobénius est d'ordre égal au degré résiduel.

nflist

Pour certains groupes de Galois (couvrant la plupart des cas en petit degré), on peut énumérer tous les corps de nombres ayant ce groupe de Galois et dont la valeur absolue du discriminant est entre certaines bornes avec `nflist`.

```
v1 = nflist("S3", [10^5,10^6]);
#v1
% = 215064
```

Il y a 215064 corps de nombres de degré 3, groupe de Galois S_3 et dont la valeur absolue du discriminant est entre 10^5 et 10^6, dans le désordre. Voici le dernier de la liste :

```
v1[#v1]
% = x^3 + 42*x^2 + 1118*x + 11492
nfdisc(v1[#v1])
% = -951368
```

nflist

```
v2 = nflist("A4", [1,10^4])
% = [x^4-2*x^3+2*x^2+2, x^4-x^3+5*x^2-4*x+3,
    x^4-2*x^3+6*x^2-4*x+2, x^4-x^3-3*x+4]
apply(nfdisc,v2)
% = [3136, 8281, 5184, 4225]
```

On trouve les 4 corps de degré 4 et groupe de Galois A_4 de discriminant au plus 10^4.

```
v3 = nflist("F5", [1,10^5])
% = [x^5-2, x^5+5*x^3+5*x-1,
    x^5-x^4+2*x^3-4*x^2+x-1, x^5-x^4+x^2+3*x+1]
```

On trouve les 4 corps de degré 5 et groupe de Galois $F_5 = C_5 \rtimes C_4$ de discriminant au plus 10^5.

Corps de classe de Hilbert

Pour calculer un corps de classe de Hilbert, on doit d'abord calculer le groupe des classes.

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

Le groupe des classes est isomorphe à $\mathbb{Z}/9\mathbb{Z}$.
On calcule un polynôme de définition relatif pour le corps de classes de Hilbert avec la fonction `bnrclassfield`.

```
R = bnrclassfield(bnf)[1]
% = x^9 + 105*x^7 + (94*a - 47)*x^6 + 1692*x^5
    + (1866*a - 933)*x^4 + 1157*x^3
    + (12348*a - 6174)*x^2 - 143031*x
    + (-11662*a + 5831)
```

Corps de classe de Hilbert

Inversement, à partir d'une extension abélienne, on peut retrouver le groupe de classes correspondant avec `rnfconductor`.

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

Ici le conducteur est trivial, et son groupe des normes est trivial dans le groupe des classes.

Corps de classe de Hilbert

On peut aussi calculer un polynôme de définition absolu pour le corps de classes de Hilbert avec l'option `flag=2`.

```
R2 = bnrclassfield(bnf,,2)
% = x^18 + 210*x^16 + 14409*x^14 + 797225*x^12
    + 20272470*x^10 + 262596933*x^8
    + 1700863175*x^6 + 5089314636*x^4
    + 6129630549*x^2 + 6766111639
```

Corps de classe de rayon

On peut également considérer des corps de classe avec un conducteur non-trivial.

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

On peut calculer à l'avance le degré absolu, la signature et le discriminant du corps de classe correspondant avec `bnrdisc`.

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

Ce corps est énorme !

Corps de classe de rayon

Pour des raisons d'efficacité, on calcule le corps de classe comme compositum de plusieurs corps plus petits.

```
bnrclassfield(bnr)
% = [x^2-3, x^8+(6*a-54)*x^6+(-165*a+693)*x^4
    +(309*a-996)*x^2-363, x^9+105*x^7+(94*a-47)*x^6
    +1692*x^5+(1866*a-933)*x^4+1157*x^3
    +(12348*a-6174)*x^2-143031*x+(-11662*a+5831)]
```

On peut forcer le calcul d'un unique polynôme avec `flag=1`.

```
bnrclassfield(bnr,,1)
% = [... énorme polynôme ...]
```

Corps de classe de rayon

On peut aussi calculer un sous-corps du corps de classe de rayon en spécifiant un sous-groupe du groupe de classes.

```
bnr = bnrinit(bnf,7)
bnr.cyc
% = [54,3]
bnrclassfield(bnr,3) \\sous-extension 3-élémentaire
% = [x^3 + (-1344*a + 2121)*x + (24031*a - 288820),
     x^3 + (-129*a + 726)*x + (-575*a + 6506)]
```

Sans le corps explicite

Le calcul d'un polynôme de définition avec `bnrclassfield` peut être très coûteux. Il est donc préférable de calculer les informations pertinentes sans construire le corps, lorsque c'est possible.

On a déjà vu l'utilisation de `bnrdisc`; on peut aussi calculer la décomposition des idéaux premiers sans le corps explicite.

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

Le Frobenius en $\mathfrak{p}_{41}$ est trivial : cet idéal premier est totalement décomposé dans l'extension de degré 162 (qu'on n'a pas calculée).

Corps de classe de rayon

Calculons un exemple complet avec un idéal en HNF et un sous-groupe donné par une matrice.

```
bnr = bnrinit(bnf,[102709,43512;0,1]);
bnr.cyc
% = [17010, 27]
bnrclassfield(bnr,[9,3;0,1]) \\indice 9
% = [x^9 + (594*a - 3567)*x^7 + ... ]
```

Module aux places à l'infini

Si le corps de base a des places réelles, on peut spécifier le module à l'infini en donnant un vecteur de 0 et 1 de longueur le nombre de plongements réels.

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

Le nombre de classes restreint du corps $\mathbb{Q}(\sqrt{217})$ est 2.

Action du groupe de Galois sur le groupe des classes

On peut calculer l'action galoisienne sur les groupes de classes de rayons avec `bnrgaloismatrix`, c'est-à-dire l'action sur le groupe de Galois relatif, sans l'extension explicite.

```
bnf = bnfinit(x^2+2*3*5*7*11);
bnf.cyc
% = [4, 2, 2, 2]
bnr = bnrinit(bnf,1,1);
gal = galoisinit(bnf);
m = bnrgaloismatrix(bnr,gal)[1]
% =
[3 0 0 0]
[0 1 0 0]
[0 0 1 0]
[0 0 0 1]
```

Questions ?

À vos claviers !