JOURNAL
de Théorie des Nombres
de BORDEAUX
anciennement Séminaire de Théorie des Nombres de Bordeaux

Loïc GRENIÉ

Fast computation of class fields given their norm group

Tome 20, n° 3 (2008), p. 707-714.

<http://jtnb.cedram.org/item?id=JTNB_2008__20_3_707_0>

Journal de Théorie des Nombres
de Bordeaux **20** (2008), 707-714

Fast computation of class fields given their norm group

par Loïc GRENIÉ

Résumé. Soit K un corps de nombres contenant, pour un nombre premier ℓ, les racines ℓ-ièmes de l'unité. Soit L une extension de Kummer de degré ℓ de K, caractérisée par son module $\mathfrak{m}$ et son groupe de normes. Soit $K_{\mathfrak{m}}$ le compositum des extensions de degré ℓ de K de module divisant $\mathfrak{m}$. En utilisant la structure d'espace vectoriel de $\mathrm{Gal}(K_{\mathfrak{m}}/K)$, nous proposons une amélioration pour la fonction `rnfkummer` de PARI/GP qui permet de ramener la complexité du calcul d'une équation de L sur K d'exponentielle à linéaire.

Abstract. Let K be a number field containing, for some prime ℓ, the ℓ-th roots of unity. Let L be a Kummer extension of degree ℓ of K characterized by its modulus $\mathfrak{m}$ and its norm group. Let $K_{\mathfrak{m}}$ be the compositum of degree ℓ extensions of K of conductor dividing $\mathfrak{m}$. Using the vector-space structure of $\mathrm{Gal}(K_{\mathfrak{m}}/K)$, we suggest a modification of the `rnfkummer` function of PARI/GP which brings the complexity of the computation of an equation of L over K from exponential to linear.

1. Introduction

Let K be a number field and let ℓ be a prime such that K contains the ℓ-th roots of unity. The algorithm [Coh, Algorithm 5.2.14, p239] gives the list of extensions of K with degree ℓ and conductor $\mathfrak{m}$. Let $\mathrm{Cl}_{\mathfrak{m}}(K)$ be the ray class group of conductor $\mathfrak{m}$. Among the extensions with conductor $\mathfrak{m}$, one might be interested in finding the specific one with norm group equal to a specific subgroup N of $\mathrm{Cl}_{\mathfrak{m}}(K)$. Computing this extension can be easily done by testing the norm group of each of the extensions computed by the preceding algorithm, and is indeed implemented that way in PARI/GP. However, it can be faster to use the vector space structure of the quotient $G = \mathrm{Cl}_{\mathfrak{m}}(K)/\ell\,\mathrm{Cl}_{\mathfrak{m}}(K)$: we know, by Galois and Kummer theory, that the set of degree ℓ extensions of K of conductor dividing $\mathfrak{m}$ are in bijection with the set of hyperplanes of G. An hyperplane of G is the kernel of a linear form

Manuscrit reçu le 20 décembre 2006.

over G so that finding L is equivalent to finding a linear form with kernel the hyperplane H_L corresponding to L. Finding a linear form is equivalent to finding its coordinates in the dual of G and while the hyperplanes of G are $\frac{\ell^{\dim G}-1}{\ell-1}$, there are only $\dim G$ vectors in the base of a dual of G. We can thus bring the complexity of the problem from exponential to linear.

We recall that the exponent of a group G is the minimum positive k such that $\forall g \in G$, g^k is the unity of G.

2. The ray class group and the Kummer compositum

2.1. The situation. We fix an algebraic closure $\overline{\mathbf{Q}}$ of $\mathbf{Q}$, a prime number ℓ and a primitive ℓ-th root of unity $\zeta_\ell \in \overline{\mathbf{Q}}$. Let $K \subset \overline{\mathbf{Q}}$ be a number field containing ζ_ℓ. We fix a conductor $\mathfrak{m} = (\mathfrak{m}_0, \mathfrak{m}_\infty)$ with $\mathfrak{m}_0$ an ideal of $\mathcal{O}_K$ and $\mathfrak{m}_\infty$ a subset of the set of real embeddings of K in $\mathbf{C}$. We are interested in the compositum $K_\mathfrak{m}$ of the Kummer extensions of K contained in $\overline{\mathbf{Q}}$ which have degree ℓ and conductor dividing $\mathfrak{m}$. Then $G = \mathrm{Gal}(K_\mathfrak{m}/K)$ is an abelian group of exponent ℓ which means that it is an $\mathbf{F}_\ell$-vector space. By class field theory, we can identify G with $\mathrm{Cl}_\mathfrak{m}(K)/\ell\,\mathrm{Cl}_\mathfrak{m}(K)$.

Any extension $L \subset K_\mathfrak{m}$ of degree ℓ over K is characterized by the subgroup $H_L = \mathrm{Gal}(K_\mathfrak{m}/L)$ which has index ℓ in G. Such a subgroup is an hyperplane of G, when G is seen as a vector space over $\mathbf{F}_\ell$. This means that it is the kernel of a linear form, i.e. that it is the orthogonal of a line of the $\mathbf{F}_\ell$-dual of G.

Let $(K^\times)^\ell = \left\{x^\ell / x \in K^\times\right\}$. Then there is a perfect $\mathbf{F}_\ell$-pairing between G and the subgroup G^* of $K^\times/(K^\times)^\ell$ of classes of elements of

$$\mathcal{O}_{K,\mathfrak{m}} = \left\{x \in K^\times/(x) \subseteq \mathfrak{m}_0 \text{ and } \forall \sigma \notin \mathfrak{m}_\infty, \sigma(x) \notin \mathbf{R}^-\right\}.$$

Such a pairing can easily be made explicit by

$$\begin{array}{ccc} G \times G^* & \longrightarrow & \mu_\ell \\ (\sigma\ ,\ \overline{x}) & \longmapsto & \frac{\sigma(\sqrt[\ell]{x})}{\sqrt[\ell]{x}} \end{array}$$

with $\mu_\ell = \{\zeta_\ell^i, i \in \mathbf{Z}\}$ the group of ℓ-th root of unity in K and for any choice of an ℓ-th root of x in $K_\mathfrak{m}$.

A degree ℓ extension L of K inside $K_\mathfrak{m}$ is $K(\sqrt[\ell]{x})$ for some $\overline{x} \in G^*$ and $\overline{x}$ is dual to H_L. Obviously all the powers $\overline{x^i}$ for $1 \leqslant i < \ell$ lead to the same extension, i.e. to the same subgroup H_L; from the point of view of $\mathbf{F}_\ell$-vector spaces, they are linear forms over G which are proportional and thus have the same kernel.

2.2. Identifying vectors in the dual. We can compute the norm group N_L for the extension L given by an equation of the form $X^\ell - x$. Such norm group is in $\mathrm{Cl}_\mathfrak{m}(K)$, but its image H_L in G also has index ℓ. The perfect pairing above provides an isomorphism between G and G^*. Suppose that

we know how to compute the equation of the norm group in a basis of G. If we do not explicit the pairing above, computing the norm group allows us to identify the line $[x]$ directed by the image $f(x)$ of x in a vector space $V = \mathbf{F}_\ell^n$ isomorphic to G^* but the equation of the norm group alone cannot identify $f(x)$ itself (except if $\ell = 2$). If x and y are elements of $\mathcal{O}_{K,\mathfrak{m}}$ (such that $\forall 0 \leqslant i, j < \ell$, $\overline{x^i} \neq \overline{y^j}$), the three elements $\sqrt[\ell]{x}$, $\sqrt[\ell]{y}$ and $\sqrt[\ell]{xy}$ generate three extensions of K of degree ℓ. The norm groups of those extensions allows us to identify the three lines $[x]$, $[y]$ and $[xy]$ directed by $f(x)$, $f(y)$ and $f(xy)$; now if we know the value x_0 of $f(x)$ inside $[x]$, we can identify $y_0 = f(y) \in [y]$ because all the lines $[x_0 t]$, $t \in [y]$ are distinct and exactly one of them is $[xy]$. Let $\lambda \in {\mathbf{F}_\ell}^\times$. If we take λx_0 instead of x_0, then we will get λy_0 instead of y_0. It follows that the choice of x_0 is irrelevant, up to multiplication by a scalar in V.

It follows that, once we fix a value $f(x)$ for an $x \in \mathcal{O}_{K,\mathfrak{m}} \backslash (K^\times)^\ell$ in V, by computing two norm groups, we can compute the value of $f(y)$ for any $y \in \mathcal{O}_{K,\mathfrak{m}}$ which is not a power of x times an ℓ-th power.

3. Algorithm

We provide the following modification of [Coh, Algorithm 5.2.14, p239]. The nine first steps are unchanged, so we do not rewrite them. Instead of looking for all cyclic extensions of given conductor, we look for a specific one with given relative norm group N. All matrices and vectors in the following algorithms are taken in $\mathbf{F}_\ell$. If M is a matrix, M_i will denote its i-th column.

Algorithm 3.1. *Compute the extension of K of degree ℓ with given norm group N given in SNF form.*

Recall that $c = d_{\mathcal{K}}$ is the dimension of the kernel $\mathcal{K}$ of the matrix $\overline{M}$.

10. [Equation of norm group] Set E_N to be the column vector obtained from N by applying only step 2 of subalgorithm 3.2.
11. [Initialize backtracking] (In what follows, $c \leq 1$ and y will be a row vector with $c-1$ components, and $Y(y)$ will be a column vector with r_w components, the $r_w - c$ first ones equal to $(0, ..., 0)^t$, then a single 1 and the components of y^t.) Set $y \leftarrow (0, ..., 0)$ (vector with $c-1$ components). Set D and G to be matrices with 0 columns and r_w rows.
12. [Check linear independence] If $Y(y)$ is in the image of D, then go to step 19
13. [Compute trial vector] Let $X \leftarrow K_c + \sum_{1 \leq j < c} y_j K_j$. Apply subalgorithm [Coh, 5.2.15, p240] to see if X corresponds to a suitable Abelian extension. If it does, set $\alpha = \prod_{1 \leq j \leq r_w} v_j^{x_j}$ (where $X = (x_1, ..., x_{r_w})^t$). Set $T \leftarrow X^\ell - \alpha$. Let $L = K[\sqrt[\ell]{\alpha}]$. Compute the matrix M of the

norm group for the extension L/K using [Coh, Algorithm 4.4.3, p215]. If $M = N$ output the polynomial T and terminate the algorithm.

14. [Update dependency matrix] Set $D \leftarrow (D|Y(y))$, concatenation of $Y(y)$ at the right of D.
15. [Norm group equation] If either G is empty or $\ell = 2$, compute according to subalgorithm 3.2 an equation E for the norm group of extension L/K and set $\alpha_1 \leftarrow \alpha$ and $E_1 \leftarrow E$, otherwise compute the equation E according to subalgorithm 3.3. Set $G \leftarrow (G|E)$.
16. [Test computability] If E_N is in the image of G, let (x_i) be the vector of coordinates: $E_N = \sum_i x_i G_i$. Set $Y_0 \leftarrow \sum_i x_i L_i$. Set y to be the last $c-1$ elements of Y_0 and go to step 13.
17. [Backtracking I] Set $i \leftarrow c$.
18. [Backtracking II] Set $i \leftarrow i-1$. If $i = 0$ set $c \leftarrow c-1$ and go to step 11.
19. [Backtracking III] Set $y_i \leftarrow y_i + 1$, and if $i < c-1$, set $y_{i+1} \leftarrow 0$. If $y_i \geq \ell$, go to step 18; otherwise go to step 13.

Subalgorithm 3.2. *Compute an equation for the norm group of a Kummer extension L/K.*

1. Compute SNF matrix M of the norm group of the extension L/K according to [Coh, Algorithm 4.4.3, p215].
2. Let i be the index such that $M_{ii} \neq 1$. Set E to be the transpose of the i-th row of M. Change the i-th element of E in -1.
3. Output E and terminate the subalgorithm.

Subalgorithm 3.3. *Compute an equation for the norm group of a Kummer extension $K(\sqrt[\ell]{\alpha})/K$, coherent with a given equation E_1 for a given extension $K(\sqrt[\ell]{\alpha_1})/K$. We suppose the images of α and α_1 generate two distinct cyclic groups of order ℓ in $K^\times/(K^\times)^\ell$.*

1. Compute equation E_2 of extension $K(\sqrt[\ell]{\alpha})/K$ according to subalgorithm 3.2.
2. Compute equation E_3 of extension $K(\sqrt[\ell]{\alpha\alpha_1})/K$ according to subalgorithm 3.2.
3. Find non-zero coefficients λ_1, λ_2 and λ_3 such that $\lambda_1 E_1 + \lambda_2 E_2 + \lambda_3 E_3 = 0$.
4. Output $-\frac{\lambda_2}{\lambda_1} E_2$ and terminate the subalgorithm.

Remark 1. This algorithm is very similar to [Coh, Algorithm 5.2.14, p239], and is very easy to implement in a single function. The relevant differences are: the algorithm stops as soon as the correct equation is found and we do not test for equations which are linearly dependant in G^*; using linear algebra, we deduce an equation for the searched field using equations for a basis of G^*.

Remark 2. In step 4 of subalgorithm 3.3, all three coefficients λ_i are non-zero because the elements α and α_1 generate different cyclic groups in $F^\times/(F^\times)^\ell$. Otherwise, one would have to compute k such that α and α_1^k have same class in that group and output kE_1.

Remark 3. If y is computed at step 4, one can omit the check of norm group group at step 13 of algorithm 3.1 : it has to be N.

Theorem 4. *Algorithm 3.1 computes the extension with at most* $2d_{\mathcal{K}}$ *(resp.* $d_{\mathcal{K}}+1$ *if* $\ell=2$*) computations of norm groups.*

Proof: The proof is now obvious. We compute the norm group once for each α at step 13. Thanks to D we take care to study only linearly independent vectors in G^*. This makes $d_{\mathcal{K}}$ computations of norm groups. If $\ell \neq 2$, for each α starting at the second we need an additional computation of the norm group for $x^\ell - \alpha\alpha_1$ in subalgorithm 3.3, which brings us to $2d_{\mathcal{K}}-1$ norm groups. For the final element we compute the norm group in step 13 (but as observed in remark 3, this computation is not necessary). □

4. Further improving efficiency

As written above, the algorithm enumerates the non-zero lines of $\mathcal{K}$, checking at each step whether they correspond to a suitable abelian extension. The non-zero elements are the elements of a family of subspaces of G^*. As observed in [Coh, top of p241], this is not linear algebra.

However we can try to improve the situation by computing as good a basis of $\mathcal{K}$ as possible. Let B_i be the family of subspaces of $\mathcal{K}$ which provides unsuitable extensions, ordered by increasing codimension. Let E_i be an equation of B_i, i.e. a morphism from $\mathcal{K}$ to $\mathbf{F}_\ell^{\operatorname{codim} B_i}$ such that $B_i = \ker E_i$. Let n be the maximum i such that $\operatorname{codim} B_i = 1$. Let $(v_j)_{1\leqslant i\leqslant d_{\mathcal{K}}}$ be a basis of $\mathcal{K}$. Using Gauss elimination technique, we can try to find a basis $(w_j)_{1\leqslant j\leqslant d_{\mathcal{K}}}$ such that $E_i(w_j)=\delta_{ij}$ for $1\leqslant i\leqslant n$. This is indeed possible if all E_i are linearly independent. Assume such a basis (w_j) has been found. Create a basis (K_i) for $\mathcal{K}$ using the following algorithm,

1. Set $i \leftarrow 1$, $d \leftarrow \min(n, d_{\mathcal{K}})$, and $K_m \leftarrow w_m$ (for all $d < m \leq d_{\mathcal{K}}$).
2. Set $K_d \leftarrow \sum_{1\leq j\leq d} w_j$.
3. Set $K_i \leftarrow K_d + w_i$
4. Set $i \leftarrow i+1$. If $i < d$ go to step 3 otherwise stop the algorithm.

If $n \leqslant d_{\mathcal{K}}$ and $\ell > 2$, this algorithm provides a basis of $\mathcal{K}$ which is such that $\forall i, j \leq n$, $B_i(K_j) \neq 0$.

If all B_i are of codimension 1 and all E_i are linearly independent, it is possible to reduce the number of tests at step 13 of algorithm 3.1 to $d_{\mathcal{K}}$.

To do that, insert after 16 a step

16.5. Set $c \leftarrow c - 1$ and go to step 11.

If the above conditions are not met, the improvement is not correct: it can fail to compute a basis of G^*. This improvement can be done anyway, but as a first try, i.e. by repeating twice the step from step 11 to step 19, once with this test enabled and, if needed, once with it disabled.

5. Speed gains

We show on an example of the speed gain that can be obtained from those methods. The author implemented algorithm 3.1 and the technique outlined in section 4. As a base field we took $K = \mathbf{Q}[\sqrt{5}, \zeta_{12}]$, $\ell = 3$ and we selected modulus $\mathfrak{m} = (2^3.3^3.5^2.11^2)$. We then have $G \simeq \mathbf{F}_3^{14}$ and we chose as L the extension whose norm group is the kernel of the projection of G on its ninth factor group. Note that the generators are not canonical but there is a way to choose them unambiguously in `PARI`. The classical algorithm of `PARI` has computed an equation for L in two minutes; the algorithm suggested here has computed it in two seconds. Note that a part of those two seconds is spent by both algorithms on common computations, in particular to compute the modulus of the extension L/K, which is smaller than $\mathfrak{m}$. The commands used to compute this extension are shown in appendix.

6. Other uses of linear independence of fields

This technique can be used to compute extensions that satisfy linear conditions in the subspace G^*.

6.1. Extensions with limited residual extensions. In [Gre], we need to compute a Galois extension L of K ramified over a given set of primes and such that $\mathrm{Gal}(L/K)$ is an ℓ-group of exponent dividing ℓ^k for a given k.

The condition on the exponent is easy to test at the residual level. Indeed let $\mathfrak{P}$ be a prime of L over a prime $\mathfrak{p}$ of K, non ramified in L/K. If the residual field of $L_{\mathfrak{P}}$ has Q elements while the residual field of $K_{\mathfrak{p}}$ has q elements, then $\mathrm{Gal}(L/K)$ will have a cyclic subgroup of order Q/q because extensions of finite fields are cyclic.

Since $\mathrm{Gal}(L/K)$ is an ℓ-group, To compute L it is sufficient to compute a tower of extensions K_i such that K_{i+1}/K_i is a compositum of Kummer extensions. If, for a prime $\mathfrak{p}_i$ of K_i over the prime $\mathfrak{p}$ of K, the residual extension has already degree ℓ^k, we cannot have additional residual extension over $\mathfrak{p}_i$; on the other hand, if the degree of the residual extension is lower that ℓ^k, having residual extension over $\mathfrak{p}_i$ is irrelevant. This construction

can be done with the function `rnfkummer` of PARI/GP, by checking the residual extensions for a predetermined set of primes.

It is possible to improve the speed of the algorithm. We limit ourselves to identifying the maximal compositum of Kummer extensions that has trivial residual extensions over a finite set of primes $\{\mathfrak{p}_1, ..., \mathfrak{p}_n\}$. If L/K is a Kummer extension, we have $L = K(\sqrt[\ell]{\alpha})$ for some $\alpha \in \mathcal{O}_K$. If we reduce modulo $\mathfrak{p}_i$, there will be residual extension if and only if the reduction of α is not an ℓ-th power. Denote $k_{\mathfrak{p}_i}$ the residual field of K modulo $\mathfrak{p}_i$. Consider the quotient $k^{(\ell)}$ of $k_{\mathfrak{p}_i}^{\times}$ by its ℓ-th power. We have that L/K has no residual extension over $\mathfrak{p}_i$ if and only if the image of α in $k^{(\ell)}$ is trivial. But the application f_i from G^* to $k^{(\ell)}$ which associates to the class of α (in G^*) the class of the image of α (in $k^{(\ell)}$) is $\mathbf{F}_\ell$-linear. This means that we are looking for the kernel of the morphism $(f_1, ..., f_n)$. Finding a kernel can be done using a basis of G^* and standard algorithms.

For this reason, we provided `rnfkummer` a mean to output a list of linearly independent (in G^*) equations of fields. This is a minor modification of the algorithm above.

6.2. General cyclic extensions of prime degree. Suppose we want to compute the Galois extension of degree ℓ with given conductor and norm group even when $\zeta_\ell \notin K$. The method is to compute Kummer extensions L_z of the compositum $K_z = K(\zeta_\ell)$ and choose those which are of the form $L(\zeta_\ell)$ for some cyclic extension L of K of degree ℓ. Those extensions are those of the form $L_z = K_z(\sqrt[\ell]{\alpha})$ for α in an eigenspace W_1 of G^* for an endomorphism τ of G^*. To find the extension we look for, we can proceed with the same method, by computing a basis (w_i) of the eigenspace W_1 and computing the extension by using linear algebra and the norm groups of the extensions corresponding to each w_i.

Appendix: timing in PARI

Here are the exact commands used to compute the equation of the field L alluded to in section 5.

```
gp > setrand(1);
gp > P=polredabs(polcompositum(polcyclo(12, 'y), 'y^2-5)[1],
16)
%1 = y^8 - 3*y^6 + 8*y^4 - 3*y^2 + 1
gp > bnf=bnfinit(P);
gp > bnr=bnrinit(bnf, 8*27*25*121, 1);
gp > default(timer, 1);
gp > rnfkummer(bnr, matdiagonal(vector(#bnr.cyc, i,
1+2*(i==9))))
time = 1mn, 58,814 ms.
```

```
%4 = x^3 + Mod(1737/4*y^7 - 450*y^6 - 1062*y^5 + 1008*y^4 +
3213*y^3 - 3177*y^2 + 225/4*y + 423, y^8 - 3*y^6 + 8*y^4 -
3*y^2 + 1)
gp > rnfkummer(bnr, matdiagonal(vector(#bnr.cyc, i,
1+2*(i==9))), -1)
time = 1,952 ms.
%5 = x^3 + Mod(1737/4*y^7 - 450*y^6 - 1062*y^5 + 1008*y^4 +
3213*y^3 - 3177*y^2 + 225/4*y + 423, y^8 - 3*y^6 + 8*y^4 -
3*y^2 + 1)
```

The -1 as last argument of `rnfkummer` chooses algorithm 3.1 instead of the standard one (without argument).

References

[Coh] Henri Cohen, *Advanced Topics in Computational Number Theory*, volume **193** of Graduate Texts in Mathematics. Springer-Verlag, New York, 2000.

[Gre] Loïc Grenié, *Comparison of semi-simplifications of Galois representations*. J. Algebra **316** (2) (2007), 608–618.

[PAR] The PARI Group, Bordeaux. *PARI/GP, version* `2.4.1`, 2006. Available from `http://pari.math.u-bordeaux.fr/`.

Loïc Grenié
Università degli Studi di Bergamo
Facoltà di Ingegneria
viale Marconi 5
24044 Dalmine, ITALY
E-mail: `loic.grenie@gmail.com`