Number field sieve

Loïc Grenié

January 16th 2013

1 The number field sieve

1.1 Introduction

We are trying to compute the class group of a number $K$ field using Buchmann algorithm. Let $n$ be the dimension of $K$, $\mathcal{O}_K$ its ring of integers. Basically one computes a limit $T$ such that

$$\mathcal{B}_T = \{ \mathfrak{p} : \mathfrak{p} \text{ is a prime ideal and } N\mathfrak{p} \leq T \}$$

contains a set of generators of the class group of $K$ and then a set $E_T$ of elements of $K$ divisible only by the primes in $\mathcal{B}_T$ and such that the ideal class group of $K$ is isomorphic to

$$\langle \mathcal{B}_T \rangle / \langle (x) : x \in E_T \rangle .$$

The elements of $\mathcal{B}_T$ are called generators and the elements of $E_T$ relations. If $\#E_T > \#\mathcal{B}_T$ products of elements of $E_T$ yield units. Using analytic class number formula one can test whether the class group and the unit group of $K$ have been computed.

The two main problems of the algorithm are finding relations (elements of $E_T$) and a linear algebra HNF problem yielding the class group and units through their logarithmic embeddings.

You can find more information in last year’s bnfinit() seminar in Atelier PARI/GP 2012.

The number field sieve aim is to efficiently find elements of $E_T$.

1.2 Biasse and Fieker’s sieve

Building on the quadratic sieve, Biasse and Fieker have suggested a way to find relations. Their basic algorithm, called line sieving is the following.

Init

1. Choose two positive integers $I$ and $J$. 
2. Choose two elements $\alpha$ and $\beta$ of $\mathcal{O}_K$ (precisely the two first elements of an LLL basis of $\mathcal{O}_K$ for a random norm). The idea is to look for elements of $E_T$ of the form $x\alpha + y\beta$ with $(x, y) \in [-I, I] \times [0, J]$.

3. Initialize a table $L$ of reals with an approximation of $\log |N_{K/Q}(x\alpha + y\beta)|$, for $(x, y) \in [-I, I] \times [0, J]$.

4. Let $y_0 \in [1, J]$. Then $P(x) = N(x\alpha + y_0\beta)$ is a polynomial of degree (at most) $n$.

Sieve

5. Fix a prime $p$ below one of the prime ideals of $B_T$.

6. Let $x_0$ be one of the roots of $P \mod p$. Then for all $k \in \mathbb{Z}$, $N((x_0 + kp)\alpha + y_0\beta)$ is congruent to $0 \mod p$. Remove $\log p$ from $L[x_0 + kp, y_0]$ for all suitable $k$.

7. Iterate over $x_0$, $p$ and $y_0$.

Relation recollection

8. Try to factor $x\alpha + y\beta$ for all $(x, y)$ such that $L[x, y] \leq l_0$ for some limit $l_0$.

They also mention

- **Lattice sieving** in which the iteration over $k$ is combined with iteration over $y_0$ through the use of a reduced basis of the lattice spanned by $(x_0, 1)$ and $(p, 0)$ in $[-I, I] \times [0, J]$.

- **Special-$q$ strategy**, where the main loop is over the “large” primes (above $I$).

We will have a false negative if $x\alpha + y\beta \in E_T$ but $L[x, y] > l_0$ after the loops and a false positive if $x\alpha + y\beta \notin E_T$ but $L[x, y] \leq l_0$. Note that both can happen if the initialization of $L[x, y]$ is too far from $\log N_{K/Q}(x\alpha + y\beta)$.

## 2 Our implementation

### 2.1 Some remarks

The algorithm is based on the fact that, for $t \in K$, $|N_{K/Q}(t)| = \prod_p Np^{v_p(t)}$. In Step 6 we thus remove $\log p$ from the log of the norm of $x\alpha + y\beta$ when we have verified that $p$ divides such norm. The limit $l_0$ accounts for the facts that we do not really factor $x\alpha + y\beta$ but find some of its prime factors in $B_T$ and that we have an approximation of the logarithm of its absolute norm.

We see here a three possible additional imprecisions: when $p$ divides the norm of $t$ we know that some ideal $p | p$ divides $t$ but

- it is possible that $p \notin B_T$ if $K$ is not Galois over $Q$, case in which we should not remove $\log p$ from $L[x, y]$;

- the norm of $p$ could be greater than $p$ or
• there could be more than one ideal above \( p \) in \( \mathcal{B}_T \) dividing \( t \). In the last two cases we should remove a multiple of \( \log p \) from \( L[x, y] \).

It would be much better to know which elements of \( \mathcal{B}_T \) divide \( x\alpha + y\beta \) and what is the log of their norm.

### 2.2 Norm

The log of the norm of each ideal of \( \mathcal{B}_T \) can easily be computed beforehand, during the computation of \( \mathcal{B}_T \). Indeed for each prime \( p \leq T \), \textsc{pari} computes \( \log p \) and the inertial index of each \( p \mid p \).

### 2.3 Divisibility

It is not very difficult to check whether a given element \( t \in K \) is an divisible by a prime ideal \( p \). We suppose that we have a \( \mathbb{Z} \)-basis \( \mathcal{V} \) of \( \mathcal{O}_K \), that we have a \( \mathbb{Z} \)-basis \( \mathcal{H} \) of \( p \) such that its coordinates in \( \mathcal{V} \) are a matrix \( H \) in HNF form, that \( p \cap \mathbb{Z} = p\mathbb{Z} \) and \( N_p = p^f \).

There is a subset \( f = \{1, \ldots\} \) of \( \{1, \ldots, n\} \) such that \( \#f = f \) and if \( H = (h_{ij}) \) then the submatrix of \( H \) made of the rows and columns with indices in \( f \) is equal to \( pI_f \) and

\[
\forall i, \quad i \not\in f \Rightarrow h_{ii} = 1 .
\]

Then \( \mathcal{H} \) is a \( \mathbb{Q} \)-basis of \( K \) and \( t \in p \) if and only if its coordinates in this basis are integral. If \( t \) is identified with the column of its coordinates in \( \mathcal{V} \), then its coordinates in \( \mathcal{H} \) are \( H^{-1}t \). Given the form of \( H \) explained above, to test the integrality of \( H^{-1}t \) it is sufficient to check that the \( f \) coordinates with indices in \( f \) are integral.

This in turn can be done the following way. Let \( C = C_p \) be the submatrix of \( pH^{-1} \) made of the rows with indices in \( f \). Then the coordinates of \( H^{-1}t \) are integral if and only if the coefficients of \( Ct \) are multiple of \( p \) i.e.

\[
t \in \ker C \mod p .
\]

While we compute \( \mathcal{B}_T \) we thus compute for each ideal \( p \) the congruence matrix \( C_p \).  

Turning back to the sieving algorithm, suppose we have chosen \( \alpha \) and \( \beta \), let \( M \) be the \( n \times 2 \) matrix made of the coordinates of \( \alpha \) and \( \beta \) in \( \mathcal{V} \) and denote \( \mathcal{Z} = \mathcal{Z}_\alpha + \mathcal{Z}_\beta \subset \mathcal{O}_K \). We then have \( x\alpha + y\beta \in p \) if and only if (identifying elements of \( K \) with their coordinates in \( \mathcal{V} \))

\[
x\alpha + y\beta \in \ker C_p \mod p
\]

which in turn is equivalent to

\[
\begin{pmatrix} x \\ y \end{pmatrix} \in \ker C_p M \mod p .
\]

At that point we are in a better situation because we can easily compute a \( \mathbb{Z} \)-basis

\[
\begin{pmatrix} a & b \\ 0 & c \end{pmatrix}
\]
in HNF form of the pullback of the lattice \( p \cap \mathbb{Z} \). Moreover \( \{a, c\} \subseteq \{1, p\} \) and \( c = p \Rightarrow b = 0 \) (and if \( a = c = p \) then elements of \( p \) are not interesting for our purpose because they are rational multiples of elements of \( \mathcal{O}_K \)). Informal benchmarks seem to indicate that computing \( \ker C_p M \) for all \( p \mid p \) is marginally faster than computing the roots modulo \( p \) of \( N(x\alpha + \beta) \) but this is not the main point of this observation: the point is that we get the precise ideal(s) that divide given \( x\alpha + y\beta \) and the log of their norm.

2.4 Higher power divisibility

The same idea as above can easily be used for a prime-power ideal \( p^r \). For each prime ideal \( p \in \mathcal{B}_T \) compute \( r = \left\lfloor \frac{\log 2I}{\log NP} \right\rfloor \). Let \( H_r \) be the matrix of a basis of \( p^r \) and \( C_{p,r} \) be the submatrix of \( p^r H_r^{-1} \) consisting of the non-zero-mod-\( p^r \) rows. It is not difficult to compute a \( \mathbb{Z} \)-basis

\[
\begin{pmatrix}
a_r & b_r \\
0 & c_r
\end{pmatrix}
\]

in HNF form of \( \ker C_{p,r} M \) i.e. of the pullback of the lattice \( p^r \cap \mathbb{Z} \). If we are lucky enough that either \((a_r, c_r) = (p^{r-1}a, c)\) or \((a_r, c_r) = (a, p^{r-1}c)\), then for \( 1 \leq k \leq r \), we have \( x\alpha + y\beta \in p^k \) if and only if \((x, y)\) lies in the lattice generated by \(\begin{pmatrix} a_r & b_r \\ 0 & c_r \end{pmatrix}\) \( \mod p^k \). If we are in one of these two lucky cases, easy congruences give the elements of all \( p^k \). Otherwise we do not try to be more clever and just be happy with the elements of \( p \).

The computation is fairly efficient. We get nearly exact valuation for all primes in \( \mathcal{B}_T \) of all elements in \([-I, I]\alpha + [1, J]\beta \) in a few milliseconds.

Shortcomings: it is not very efficient for higher degrees \( n \geq 7 \) or too small discriminant.

2.5 General organization

The general algorithm is as Function \texttt{algsieve} shown below. The logarithm of archimedean embeddings is described in the next paragraph.
Input: \( K, T, \mathcal{B}_T, \{C_p\}, \{C_{p,r}\}, S \subset \mathcal{B}_T \)

Output: Some elements of \( K \) factorizable over \( \mathcal{B}_T \)

1. \( R \leftarrow \emptyset; \)
2. \( I \leftarrow \lfloor T \log \log |\Delta_K| \rfloor; \)
3. \( J \leftarrow \lfloor \log |\Delta_K| \rfloor; \)
4. \( l_0 \leftarrow \frac{1}{2} \log T; \)
5. \( N \leftarrow \text{random_norm}; \)
6. for \( I \in \{O_K\} \cup S \) do
   7. \( B \leftarrow \text{LLL}(I, N); \)
   8. \( \alpha \leftarrow B[1]; \)
   9. \( \beta \leftarrow B[2]; \)
   10. if \( \alpha \) is factorizable over \( \mathcal{B}_T \) then
       11. \( R \leftarrow R \cup \{\alpha\}; \)
   12. end
   13. \( M \leftarrow (\alpha|\beta); \)
   14. \( L \leftarrow \logarch(K, \alpha, \beta, I, J); \)
   15. for \( p \in \mathcal{B}_T \) do
       16. \( [a, b, 0, c] \leftarrow \ker C_p M \mod p; \)
       17. \( [a_r, b_r, 0, c_r] \leftarrow \ker C_{p,r} M \mod p^r; \)
       18. for \( j = c \) to \( J \) step \( c \) do
           19. for \( i = -I + ((I + bj) \% a) \) to \( I \) step \( a \) do
               20. if Lucky case and \( i \equiv ja_r \) (mod \( p^2 \)) then
                   21. \( L[i, j] \leftarrow L[i, j] - \min(r, v_p(i - ja_r)) \log N p; \)
               else
                   22. \( L[i, j] \leftarrow L[i, j] - \log N p; \)
               end
           end
       end
   end
26. end
27. for \( -I \leq i \leq I \) do
   28. for \( 1 \leq j \leq J \) do
       29. if \( L[i, j] \leq l_0 \) then
           30. if \( i\alpha + j\beta \) is factorizable over \( \mathcal{B}_T \) then
               31. \( R \leftarrow R \cup \{i\alpha + j\beta\}; \)
           end
       end
   end
34. end
35. end
36. end
37. end
38. return \( R; \)

Function algsieve\((K,T,\mathcal{B}_T,\{C_p\},\{C_{p,r}\},S \subset \mathcal{B}_T)\)
2.6 Archimedean embedding

We used the following method to compute all $L[x, y] \simeq \log |N_{K/Q}(x\alpha + y\beta)|$. First, observe that

$$L[x, y] = n \log y + \log |P(x)|$$

where $P(x) = N_{K/Q}(x\alpha + \beta)$.

To compute $f(t) = \log |P(t)|$ we observe that

$$f'(t) = \frac{P'(t)}{P(t)}$$

and thus that $f'$ changes sign where exactly one of $P$ or $P'$ changes sign.

We compute their square-free factorization of $Q_1 = \frac{P}{\gcd(P, P')}$ and $Q_2 = \frac{P'}{\gcd(P, P')}$. We thus have

$$P = \gcd(P, P') \prod_{i=1}^{k} Q_i^{v_i}$$

$$P' = \gcd(P, P') \prod_{i=k+1}^{k+l} Q_i^{v_i}$$

We then compute the real zeros of the $Q_i$’s such that $v_i$ is odd, using Uspensky method. These are the points $\{t_i\}$ where $f'$ changes sign.

As a further optimization, observe that, if $\frac{\beta}{\alpha}$ is of degree $n$, then $P$ changes sign at the real archimedean embeddings of $-\frac{\beta}{\alpha}$ thus we can save the computation of the real roots of $P$. If instead $\frac{\beta}{\alpha}$ is of degree lower than $n$ then $P$ is a power and Uspensky method is way faster.

We substitute the $t_i$ with their best approximations from below and from above with rational numbers of denominators at most $J$ and add $-I$ and $I$ to the list. Then $f(t)$ is monotonous on each $[t_i, t_{i+1}]$. On the segment $[t_i, t_{i+1}]$ we compute $f(t)$ by dichotomy: we compute an approximation of $f(t)$ by linear interpolation if $|f(t_i) - f(t_{i+1})| \leq 2$ and cut the segment in half otherwise. As we can expect, we compute a lot of approximations of $f$ near the roots of $P$.

The result is excellent: the error on the computation of the log of the norm is lower, usually much lower, than 2.

The slowest part is the computation of the real roots of $P'$ and can become a significant part of the whole sieve if $n$ is above 4. To dilute the problem we cannot take $I$ and $J$ too small.

The corresponding algorithm is given as Function $\text{logarch}$ below.
Input: $K, \alpha, \beta, I$ and $J$

Output: A table $L$ such that for $-I \leq x \leq I$ and $1 \leq y \leq J$,

$$L[x, y] \simeq \log |N_{K/Q}(x\alpha + y\beta)|$$

1. $L \leftarrow \text{array}(I, J)$;
2. $P \leftarrow N(x\alpha + \beta)$;
3. $P_1 \leftarrow P'$;
4. $D \leftarrow (P, P_1)$;
5. $P \leftarrow P/D$;
6. $P_1 \leftarrow P_1/D$;
7. $T \leftarrow \text{concat}(\text{realroots}(\text{SQFF of } P), \text{realroots}(\text{SQFF of } P_1))$;
8. $T \leftarrow \text{bestapprs}(T, J)$;
9. $T \leftarrow \text{concat}([-I, T, I])$;
10. $A \leftarrow \text{array}(|T|)$;
11. $A[1] \leftarrow \log |P(T[1])|$;
12. for $2 \leq i \leq |T|$ do
   13.     $A[i] \leftarrow \log |P(T[i])|$;
   15.        Increase the size of $T$ and $A$;
   16.        $T[(i+1)..|T|] = T[i..(|T| - 1)]$;
   17.        $T[i] \leftarrow T[i-1]+T[i]$;
   18.        $A[i] \leftarrow \log |P(T[i])|$;
   19.     end
12. end
21. for $1 \leq j \leq J$ do
22.     $[k, r, dr] \leftarrow [0, 0, 0]$;
23.     for $-I \leq i \leq I$ do
24.         $r \leftarrow r + dr$;
25.         while $k < |T$ and $\frac{i}{j} \geq T[k+1]$ do
26.             $k \leftarrow k + 1$;
27.             $dr \leftarrow \frac{A[k+1]-A[k]}{j(T[k+1]-T[k])}$;
28.             $r \leftarrow n \log j + A[k] + (i - j \cdot A[k])dr$;
29.         end
30.     $L[i, j] \leftarrow r$;
31. end
32. end
33. return $L$

Function $\text{logarch}(K, \alpha, \beta, I, J)$