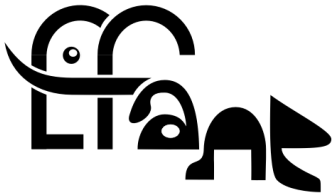
Solving relative norm equations in abelian number fields

Andreas Enge

LFANT project-team
INRIA Bordeaux–Sud-Ouest
andreas.enge@inria.fr
http://www.math.u-bordeaux.fr/~aenge

Finite Geometries, Fifth Irsee Conference, 15 September 2017
(joint work with Bernhard Schmidt, NTU, Singapore)



Solving norm equations

Circulant Hadamard matrices

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & \cdots & h_{n-1} \\ h_{n-1} & h_0 & h_1 & \cdots & h_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1 & h_2 & h_3 i & \cdots & h_0 \end{pmatrix}$$

with $h_i \in \{\pm 1\}$, $\quad H \cdot H^T = n \cdot \mathrm{id}$

Circulant Hadamard matrices

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & \cdots & h_{n-1} \\ h_{n-1} & h_0 & h_1 & \cdots & h_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1 & h_2 & h_3 i & \cdots & h_0 \end{pmatrix}$$

with $h_i \in \{\pm 1\}$, $\quad H \cdot H^T = n \cdot \mathrm{id}$

Let

$$\chi = \sum_{i=0}^{n-1} h_i \zeta_n^i.$$

Then

$$\boxed{\chi\overline{\chi} = n}$$

Abelian difference sets

$$D \subseteq G \quad \leftrightarrow \quad D = \sum_{g \in D} 1 \cdot \langle g \rangle \in \mathbb{Z}[G]$$

Abelian difference sets

$$D \subseteq G \quad \leftrightarrow \quad D = \sum_{g \in D} 1 \cdot \langle g \rangle \in \mathbb{Z}[G]$$

$$\overline{D} = \sum_{g \in D} 1 \cdot \langle g^{-1} \rangle$$

$$D \text{ a } (v, k, \lambda)\text{-difference set} \quad \Leftrightarrow \quad D\overline{D} = \sum_{g \in G \setminus \{1\}} \lambda \cdot \langle g \rangle + k \cdot \langle 1 \rangle$$

$$= (k - \lambda) \cdot \langle 1 \rangle + \lambda \cdot G$$

Abelian difference sets

$$D \subseteq G \quad \leftrightarrow \quad D = \sum_{g \in D} 1 \cdot \langle g \rangle \in \mathbb{Z}[G]$$

$$\overline{D} = \sum_{g \in D} 1 \cdot \langle g^{-1} \rangle$$

$$D \text{ a } (v, k, \lambda)\text{-difference set} \quad \Leftrightarrow \quad D\overline{D} = \sum_{g \in G \setminus \{1\}} \lambda \cdot \langle g \rangle + k \cdot \langle 1 \rangle$$

$$= (k - \lambda) \cdot \langle 1 \rangle + \lambda \cdot G$$

$$\chi : G \to \{\zeta_v^i : i = 0, \ldots, v-1\} \subseteq \mathbb{C}$$

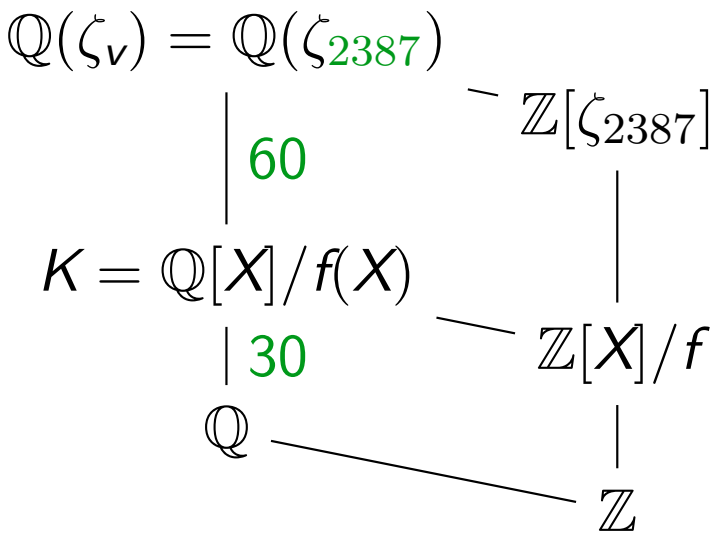
$$\chi(D)\overline{\chi}(D) = k - \lambda = n$$

Cyclic case:

$$D \subseteq \{0, \ldots, v-1\}, \quad \chi(D) = \sum_{i \in D} \zeta_v^i$$

Solving norm equations

(Abelian) number fields

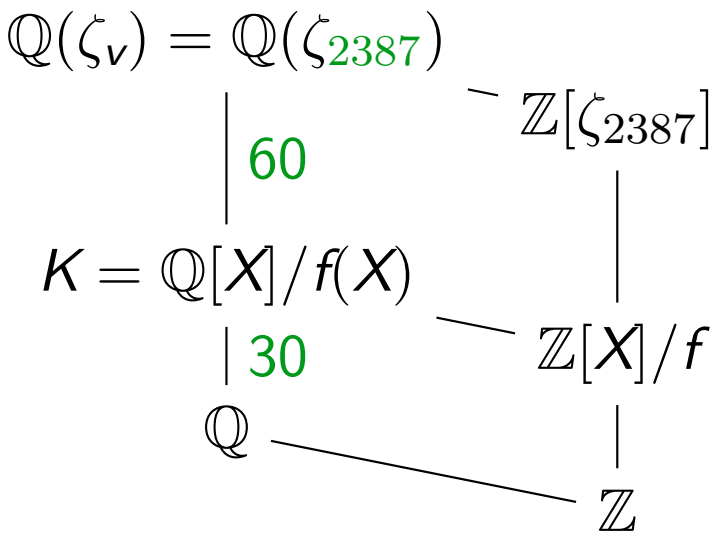


Ex.: $f = \Phi_v(X)$

$\sigma : K \to \mathbb{C}, \quad X \mapsto$ root of f

$\sigma_i : X \mapsto \zeta_v^i$ for $\gcd(v, i) = 1$

(Abelian) number fields



Ex.: $f = \Phi_v(X)$

$\sigma : K \to \mathbb{C}, \quad X \mapsto$ root of f

$\sigma_i : X \mapsto \zeta_v^i$ for $\gcd(v, i) = 1$

Trace

$$\begin{aligned} \mathrm{Tr} \quad : \quad & K \to \mathbb{Q} \\ & \alpha \mapsto \sum_{\sigma} \sigma(\alpha) \end{aligned}$$

Positive definite bilinear form

$$\begin{aligned} T(\alpha, \beta) &= \mathrm{Tr}(\alpha \cdot \overline{\beta}) \\ T(\alpha, \alpha) &= \sum \sigma(\alpha)\overline{\sigma(\alpha)} \end{aligned}$$

Ideal factorisation

$$\chi\overline{\chi} = n \quad \Rightarrow \quad \mathfrak{a}\overline{\mathfrak{a}} = (n) \text{ with } \mathfrak{a} = (\chi)$$

Ex.:

$$(n) = \mathfrak{p}\overline{\mathfrak{p}}\mathfrak{q}\overline{\mathfrak{q}}$$
$$\Rightarrow \quad \mathfrak{a} = \mathfrak{p}\mathfrak{q};\ \mathfrak{p}\overline{\mathfrak{q}};\quad \overline{\mathfrak{p}}\mathfrak{q};\ \overline{\mathfrak{p}}\overline{\mathfrak{q}}$$

Look for generator χ of $\mathfrak{p}\mathfrak{q}$ or $\mathfrak{p}\overline{\mathfrak{q}}$.

Ideal factorisation

$$\chi\overline{\chi} = n \quad \Rightarrow \quad \mathfrak{a}\overline{\mathfrak{a}} = (n) \text{ with } \mathfrak{a} = (\chi)$$

Ex.:

$$(n) = \mathfrak{p}\overline{\mathfrak{p}}\mathfrak{q}\overline{\mathfrak{q}}$$
$$\Rightarrow \quad \mathfrak{a} = \mathfrak{p}\mathfrak{q};\ \mathfrak{p}\overline{\mathfrak{q}};\quad \overline{\mathfrak{p}}\mathfrak{q};\ \overline{\mathfrak{p}}\overline{\mathfrak{q}}$$

Look for generator χ of $\mathfrak{p}\mathfrak{q}$ or $\mathfrak{p}\overline{\mathfrak{q}}$.

Heuristic: χ is "small"
LLL finds element with small T-norm in the lattice $\mathfrak{a}$ of dimension $\deg(K)$.

Ideal factorisation

$$\chi\overline{\chi} = n \quad \Rightarrow \quad \mathfrak{a}\overline{\mathfrak{a}} = (n) \text{ with } \mathfrak{a} = (\chi)$$

Ex.:

$$(n) = \mathfrak{p}\overline{\mathfrak{p}}\mathfrak{q}\overline{\mathfrak{q}}$$
$$\Rightarrow \quad \mathfrak{a} = \mathfrak{p}\mathfrak{q};\ \mathfrak{p}\overline{\mathfrak{q}};\quad \overline{\mathfrak{p}}\mathfrak{q};\ \overline{\mathfrak{p}}\overline{\mathfrak{q}}$$

Look for generator χ of $\mathfrak{p}\mathfrak{q}$ or $\mathfrak{p}\overline{\mathfrak{q}}$.

Heuristic: χ is "small"
LLL finds element with small T-norm in the lattice $\mathfrak{a}$ of dimension $\deg(K)$.

More advanced algorithm:
Compute class group and generalised discrete logarithm in it.
subexponential

Solving norm equations

History

Given $\mathfrak{a}$ and n with $\mathfrak{a}\overline{\mathfrak{a}} = (n)$, output χ s.t. $\chi\overline{\chi} = n$ or failure.

- Gentry–Szydlo (2002)
  - algorithm for $f = X^v - 1$
  - breaks lattice based cryptosystems in practice
- Lenstra–Silverberg (2014)
  - deterministic polynomial time complexity
- Kirchner (2016)
  - generalisation to CM number fields
  - claim of polynomial complexity doubtful
  - code not available
- E.–Schmidt (2017)
  - generalisation to abelian number fields
  - polynomial complexity very probable

Ideas

Given $\mathfrak{a}$ and $w \in K$ with $\mathfrak{a}\overline{\mathfrak{a}} = (w)$, output χ s.t. $\chi\overline{\chi} = w$.

First idea: Use adapted T-norm

$$\begin{aligned} T_w(x, y) &= \mathrm{Tr}(x\overline{y}/w) \in \mathbb{Z} \text{ for } x, y \in \mathfrak{a} \\ T_w(\chi, \chi) &= \deg(K) \end{aligned}$$

Ideas

Given $\mathfrak{a}$ and $w \in K$ with $\mathfrak{a}\overline{\mathfrak{a}} = (w)$, output χ s.t. $\chi\overline{\chi} = w$.

First idea: Use adapted T-norm

$$\begin{aligned} T_w(x, y) &= \mathrm{Tr}(x\overline{y}/w) \in \mathbb{Z} \text{ for } x, y \in \mathfrak{a} \\ T_w(\chi, \chi) &= \deg(K) \end{aligned}$$

Second (rough) idea:

Choose (totally split) large prime P and let $e = P - 1$.

Compute

$$\mathfrak{a}^e = (\chi^e) \text{ with } \mathfrak{a}^e\overline{\mathfrak{a}}^e = (w^e) \quad \text{and } \chi^e \equiv 1 \pmod{P}$$

Ideas

Given $\mathfrak{a}$ and $w \in K$ with $\mathfrak{a}\overline{\mathfrak{a}} = (w)$, output χ s.t. $\chi\overline{\chi} = w$.

First idea: Use adapted T-norm

$$\begin{aligned} T_w(x, y) &= \mathrm{Tr}(x\overline{y}/w) \in \mathbb{Z} \text{ for } x, y \in \mathfrak{a} \\ T_w(\chi, \chi) &= \deg(K) \end{aligned}$$

Second (rough) idea:

Choose (totally split) large prime P and let $e = P - 1$.

Compute

$$\mathfrak{a}^e = (\chi^e) \text{ with } \mathfrak{a}^e\overline{\mathfrak{a}}^e = (w^e) \quad \text{and } \chi^e \equiv 1 \pmod{P}$$

"Ideal hopping" and frequent LLL reductions to compute

$$\delta = \chi^e \cdot \varepsilon \in K$$

with ε small and

$$\delta' = \delta \bmod P = \varepsilon \bmod P.$$

$$\chi^e = \delta\big(\mathrm{lift}(\delta')\big)^{-1}$$

Algorithm — Initialisation

$$e = \sum_{i=0}^{r} e^{(i)} 2^{r-i} = e_0 e_1 e_2 \ldots e_{r-1} e_r$$

$$e_k = \sum_{i=0}^{k} e^{(i)} 2^{k-i} = \left\lfloor e/2^{r-k} \right\rfloor = e_0 e_1 e_2 \ldots e_{k-1} e_k$$

Invariants:

$$\begin{aligned} \mathfrak{a}_k &= (\chi_k) \\ w_k &= \chi_k \overline{\chi}_k \\ \delta_k &= \chi^{e_k} \overline{\chi}_k \\ \delta'_k &= \delta_k \bmod P \end{aligned}$$

Initialisation $k = 0$:

$$\begin{aligned} \mathfrak{a}_0 &= \mathfrak{a} \\ w_0 &= w \\ \delta_0 &= w \\ \delta'_k &= w \bmod P \end{aligned}$$

Algorithm — Step $k-1 \to k$ for $e^{(k)} = 0$

Square!

$$\mathfrak{a}_{k-1} = (\chi_{k-1}),\ w_{k-1} = \chi_{k-1}\overline{\chi}_{k-1},\ \delta_{k-1} = \chi^{e_{k-1}}\overline{\chi}_{k-1},\ \delta'_{k-1} = \delta_{k-1} \bmod P$$

$$\begin{aligned}
\mathfrak{b}_k &= \mathfrak{a}_{k-1}^2 \\
\beta_k &= \chi_{k-1}^2 \\
u_k &= \beta_k\overline{\beta}_k = w_{k-1}^2 \\
\gamma_k &= \text{small element in } \mathfrak{b}_k \text{ w.r.t. } \mathrm{Tr}(x\bar{y}/u_k) \quad \leftarrow \text{ LLL} \\
\mathfrak{a}_k &= \overline{(\gamma_k)\mathfrak{b}_k^{-1}} \\
\chi_k &= \overline{\gamma_k\beta_k^{-1}} \\
w_k &= (\gamma_k\overline{\gamma}_k)(\beta_k\overline{\beta}_k)^{-1} = \gamma_k\overline{\gamma}_k/u_k \\
\delta_k &= \delta_{k-1}^2 w_{k-1}^{-2}\gamma_k \quad \leftarrow \text{ in factored form!} \\
\delta'_k &= (\delta'_{k-1})^2 w_{k-1}^{-2}\gamma_k \bmod P \quad \in \mathbb{Z}/p\mathbb{Z}[X]
\end{aligned}$$

Algorithm — Step $k-1 \to k$ for $e^{(k)} = 1$

Square — then multiply!

Algorithm — The End

$$\psi = \chi^{P-1} = \delta_r\big(\operatorname{lift}(\delta'_r)\big)^{-1}$$

Algorithm — The End

$$\psi = \chi^{P-1} = \delta_r\big(\operatorname{lift}(\delta_r')\big)^{-1}$$

Choose second prime P', compute

$$\psi' = \chi^{P'-1}$$

$$d = u(P-1) - v(P'-1)$$

$$\chi^d = \psi^u(\psi')^{-v}$$

and take a d-th root in K.

Solving norm equations

Implementation

- About 1100 lines in PARI/GP: `http://pari.math.u-bordeaux.fr/`
- It works!

```
? test_random()
P 630169, P' = P + 4774
Step 1
Time for G:   2.2
Time for LLL: 2.4
Small element: [-184, -104, -92, -148, -192, -182, -178, ...]~
Step 2
Double, norm 1
Step 3
Double, norm 1
...
delta 1
Mat([[-184, -104, -92, -148, -192, -182, -178, ...]~, 4774])
Cumulated core time:     47
```

TODO

- Find example where LLL does not succeed immediately.
- Larger examples, require lower running times
  - PARI/GP not optimised for abelian fields
  - integral basis takes ages
  - use embedding into $\mathbb{Q}(\zeta)$ of degree 1800?
  - but then computation of multiplication tensor too costly...
  - work with polynomials, lazy reduction and do everything by hand!
- Prove polynomial complexity.
- Apply to finite geometry setting!