An LLL algorithm for module lattices

Changmin Lee[1], **Alice Pellet-Mary**[2], Damien Stehlé[1]
and Alexandre Wallet[3]

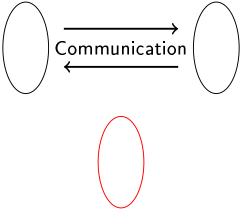
[1] ENS de Lyon, [2] KU Leuven, [3] NTT Tokyo

Séminaire Grace,
December 19, 2019

https://eprint.iacr.org/2019/1035

Cryptography and hard problems

Cryptographic primitives
(e.g. confidential communication)



Eavesdropper

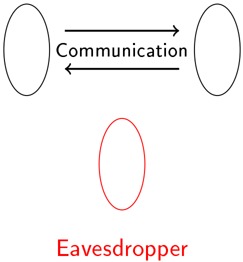
reduction ⟵

(Supposedly) intractable problems

factorisation
discrete logarithm
lattice problems
...

Cryptography and hard problems

Cryptographic primitives
(e.g. confidential communication)

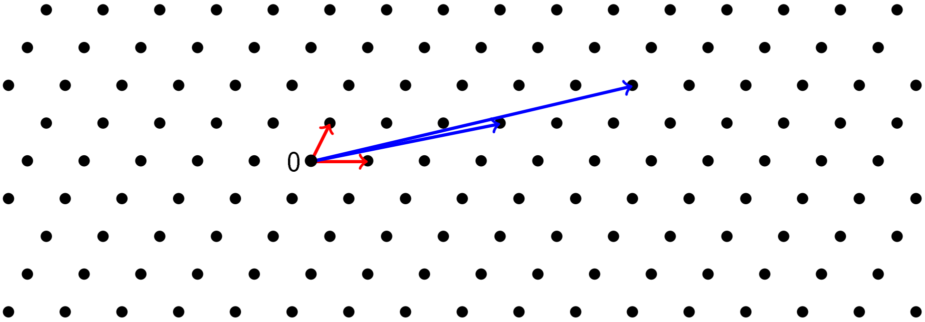


reduction

(Supposedly) intractable problems
in a quantum world

~~factorisation~~
~~discrete logarithm~~
lattice problems
...

Lattices

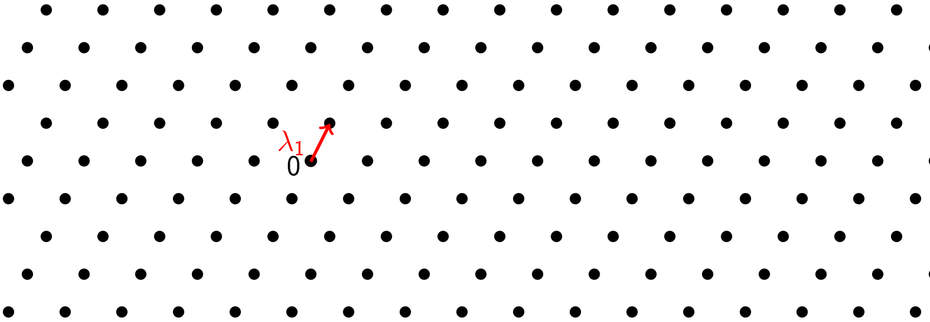


Lattice

A (full-rank) lattice L is a subset of $\mathbb{R}^n$ of the form $L = \{Bx \mid x \in \mathbb{Z}^n\}$, with $B \in \mathbb{R}^{n \times n}$ invertible. B is a basis of L.

$\begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix}$ and $\begin{pmatrix} 17 & 10 \\ 4 & 2 \end{pmatrix}$ are two bases of the above lattice.

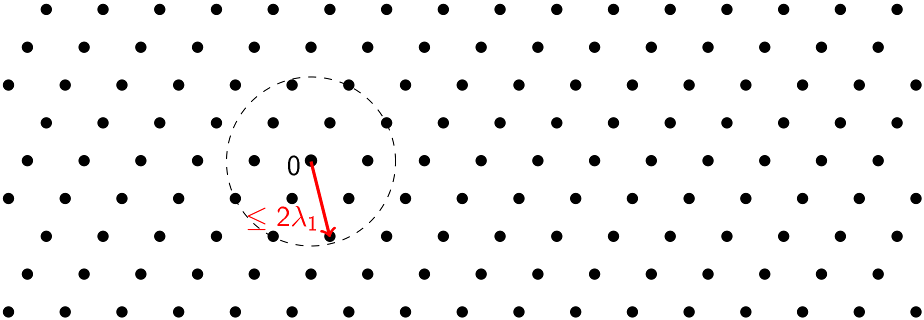
Lattice problems



Shortest Vector Problem (SVP)

Find a shortest (in Euclidean norm) non-zero vector.
Its Euclidean norm is denoted λ_1.

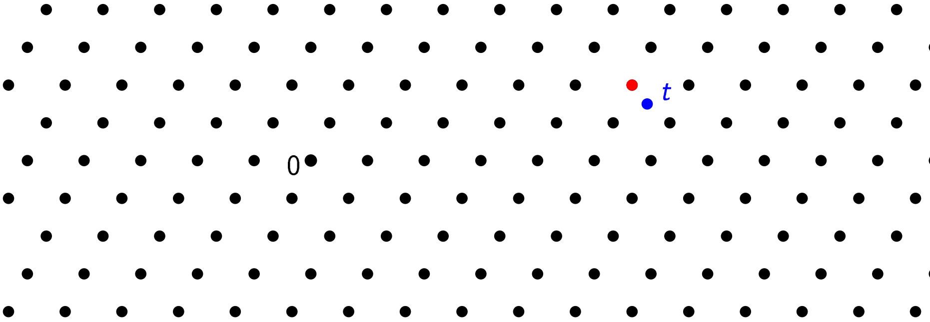
Lattice problems



Approximate Shortest Vector Problem (approx-SVP)

Find a short (in Euclidean norm) non-zero vector.
(e.g. of norm $\leq 2\lambda_1$).

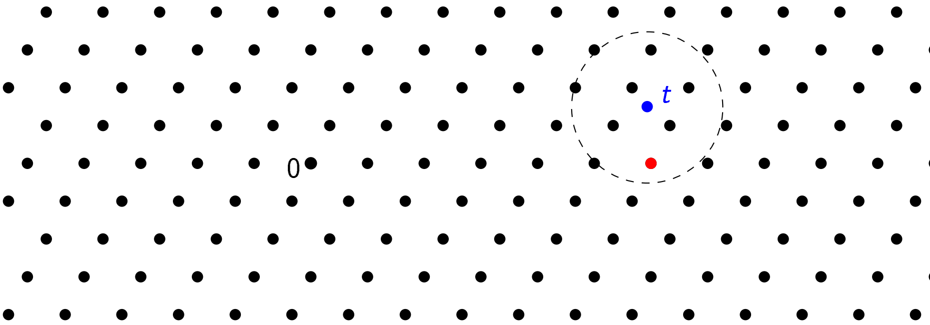
Lattice problems



Closest Vector Problem (CVP)

Given a target point t, find a point of the lattice closest to t.

Lattice problems

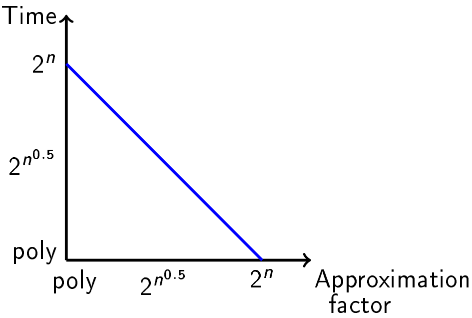


Approximate Closest Vector Problem (approx-CVP)

Given a target point t, find a point of the lattice close to t.

Hardness of lattice problems

Best Time/Approximation trade-off for SVP and CVP (even quantumly):
BKZ algorithm [Sch87,SE94]

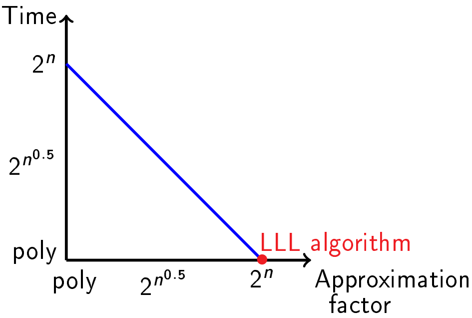


[Sch87] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. TCS.

[SE94] C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. Mathematical programming.

Hardness of lattice problems

Best Time/Approximation trade-off for SVP and CVP (even quantumly):
BKZ algorithm [Sch87,SE94]



[LLL82] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen.

Structured lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using structured lattices

Structured lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using structured lattices

Example: NIST post-quantum standardization process

- 26 candidates (2nd round)
- 12 lattice-based
- 11 using structured lattices

Structured lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using structured lattices

Example: NIST post-quantum standardization process

- 26 candidates (2nd round)
- 12 lattice-based
- 11 using structured lattices

	Frodo (lvl 1) (unstructured lattices)	Kyber (lvl 1) (structured lattices)
public key size (in Bytes)	9 616	800
ciphertexts size (in Bytes)	9 720	736

Ideal lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using structured lattices

Ideal lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using lattices with a structured basis

$$M_{\mathbf{a}} = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix}$$

Ideal lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using lattices with a structured basis

$$M_{\mathbf{a}} = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix}$$

multiplication by
$a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$
mod $X^n - 1$

Ideal lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using lattices with a structured basis

$$M_{\mathbf{a}} = \begin{pmatrix} a_0 & -a_{n-1} & \cdots & -a_1 \\ a_1 & a_0 & \cdots & -a_2 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix}$$

multiplication by
$a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$
mod $X^n + 1$
$(n = 2^\ell)$

Ideal lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using lattices with a structured basis

$$M_{\mathbf{a}} = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_1 + a_2 \\ a_1 & a_0 + a_{n-1} & \cdots & a_2 + a_3 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \dots & a_0 + a_{n-1} \end{pmatrix}$$

multiplication by
$a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$
mod $X^n - X - 1$
(n prime)

Ideal lattices

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using lattices with a structured basis

$$M_{\mathbf{a}} = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_1 + a_2 \\ a_1 & a_0 + a_{n-1} & \cdots & a_2 + a_3 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \dots & a_0 + a_{n-1} \end{pmatrix}$$

multiplication by
$a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$
mod $X^n - X - 1$
(n prime)

basis of a (principal) ideal lattice

Module lattices

Ring R

- $R = \mathbb{Z}[X]/P(X)$ with P irreducible, degree d
- $M_{\mathbf{a}}$ = multiplication by $\mathbf{a}$ in R

Module lattices

Ring R

- $R = \mathbb{Z}[X]/P(X)$ with P irreducible, degree d
- $M_{\mathbf{a}}$ = multiplication by $\mathbf{a}$ in R

$$d \updownarrow \begin{pmatrix} M_{\mathbf{a}_{11}} & M_{\mathbf{a}_{12}} & \cdots & M_{\mathbf{a}_{1k}} \\ M_{\mathbf{a}_{21}} & M_{\mathbf{a}_{22}} & \cdots & M_{\mathbf{a}_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ M_{\mathbf{a}_{k1}} & M_{\mathbf{a}_{k2}} & \cdots & M_{\mathbf{a}_{kk}} \end{pmatrix} \updownarrow dk$$

Module lattices

Ring R

- $R = \mathbb{Z}[X]/P(X)$ with P irreducible, degree d
- $M_{\mathbf{a}}$ = multiplication by $\mathbf{a}$ in R

$$d \left\{ \begin{pmatrix} M_{\mathbf{a}_{11}} & M_{\mathbf{a}_{12}} & \cdots & M_{\mathbf{a}_{1k}} \\ M_{\mathbf{a}_{21}} & M_{\mathbf{a}_{22}} & \cdots & M_{\mathbf{a}_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ M_{\mathbf{a}_{k1}} & M_{\mathbf{a}_{k2}} & \cdots & M_{\mathbf{a}_{kk}} \end{pmatrix} \right\} dk$$

Is SVP still hard when restricted to module lattices?

Module lattices

Ring R

- $R = \mathbb{Z}[X]/P(X)$ with P irreducible, degree d
- $M_{\mathbf{a}}$ = multiplication by $\mathbf{a}$ in R

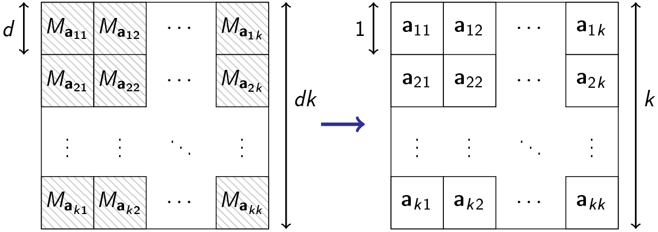
$$
\underbrace{\begin{pmatrix} M_{\mathbf{a}_{11}} & M_{\mathbf{a}_{12}} & \cdots & M_{\mathbf{a}_{1k}} \\ M_{\mathbf{a}_{21}} & M_{\mathbf{a}_{22}} & \cdots & M_{\mathbf{a}_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ M_{\mathbf{a}_{k1}} & M_{\mathbf{a}_{k2}} & \cdots & M_{\mathbf{a}_{kk}} \end{pmatrix}}_{d \text{ per block},\ dk} \longrightarrow \underbrace{\begin{pmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} & \cdots & \mathbf{a}_{1k} \\ \mathbf{a}_{21} & \mathbf{a}_{22} & \cdots & \mathbf{a}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{k1} & \mathbf{a}_{k2} & \cdots & \mathbf{a}_{kk} \end{pmatrix}}_{1 \text{ per entry},\ k}
$$

Is SVP still hard when restricted to module lattices?

Module lattices

Ring R

- $R = \mathbb{Z}[X]/P(X)$ with P irreducible, degree d
- $M_{\mathbf{a}}$ = multiplication by $\mathbf{a}$ in R



dimension $n = dk$ over $\mathbb{Z}$

dimension k over R
(k = rank of the module)

Is SVP still hard when restricted to module lattices?

Module lattices

Ring R

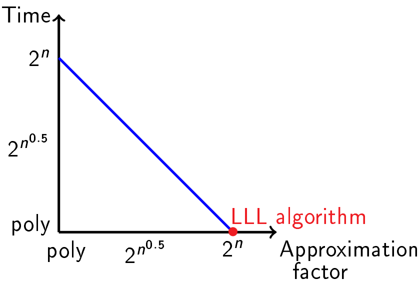
- $R = \mathbb{Z}[X]/P(X)$ with P irreducible, degree d
- $M_{\mathbf{a}}$ = multiplication by $\mathbf{a}$ in R

$$\begin{pmatrix} M_{\mathbf{a}_{11}} & M_{\mathbf{a}_{12}} & \cdots & M_{\mathbf{a}_{1k}} \\ M_{\mathbf{a}_{21}} & M_{\mathbf{a}_{22}} & \cdots & M_{\mathbf{a}_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ M_{\mathbf{a}_{k1}} & M_{\mathbf{a}_{k2}} & \cdots & M_{\mathbf{a}_{kk}} \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} & \cdots & \mathbf{a}_{1k} \\ \mathbf{a}_{21} & \mathbf{a}_{22} & \cdots & \mathbf{a}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{k1} & \mathbf{a}_{k2} & \cdots & \mathbf{a}_{kk} \end{pmatrix}$$

block size d, total size dk	block size 1, total size k
dimension $n = dk$ over $\mathbb{Z}$	dimension k over R (k = rank of the module)
Typically $500 \leq dk \leq 1000$	Typically $k \leq 10$

Is SVP still hard when restricted to module lattices?

Objective

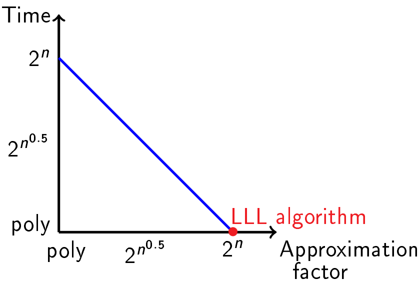


Lattice reduction over $\mathbb{Z}$

Module lattices

- large dimension over $\mathbb{Z}$
- small dimension over R

Objective



Lattice reduction over $\mathbb{Z}$

Module lattices

- large dimension over $\mathbb{Z}$
- small dimension over R

Can we extend the LLL algorithm to lattices over R?

Previous works and result

[Nap96] LLL for some specific number fields
no bound on quality / run-time

[Nap96] H. Napias. A generalization of the LLL-algorithm over Euclidean rings or orders. Journal de théorie des nombres de Bordeaux.

Previous works and result

[Nap96] LLL for some specific number fields
no bound on quality / run-time

[FP96] LLL for any number fields
no bound on quality / run-time
bound on run-time for specific number fields

[FP96] C. Fieker, M. E. Pohst. Lattices over number fields. ANTS.

Previous works and result

[Nap96]	LLL for some specific number fields no bound on quality / run-time
[FP96]	LLL for any number fields no bound on quality / run-time bound on run-time for specific number fields
[KL17]	LLL for norm-Euclidean fields bound on run-time but not on quality bound on quality for biquadratic fields

[KL17] T. Kim, C. Lee. Lattice reductions over euclidean rings with applications to cryptanalysis. IMACC.

Previous works and result

[Nap96] LLL for some specific number fields
no bound on quality / run-time

[FP96] LLL for any number fields
no bound on quality / run-time
bound on run-time for specific number fields

[KL17] LLL for norm-Euclidean fields
bound on run-time but not on quality
bound on quality for biquadratic fields

[LPSW19] LLL for any number field
bound on quality and run-time if oracle solving CVP in a fixed lattice (depending on R)

[LPSW19] C. Lee, A. Pellet-Mary, D. Stehlé, A. Wallet. An LLL algorithm for module lattices. To appear at Asiacrypt 2019.

Outline of the talk

1. Module lattices
2. The LLL algorithm
3. The Lagrange-Gauss algorithm
4. Computing the relaxed Euclidean division

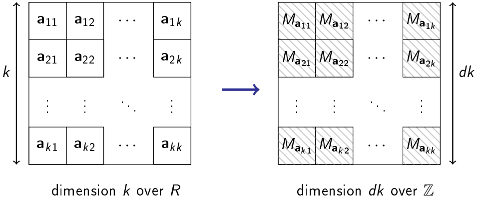
Outline of the talk

1. Module lattices
2. The LLL algorithm
3. The Lagrange-Gauss algorithm
4. Computing the relaxed Euclidean division

Canonical embedding

Reminder

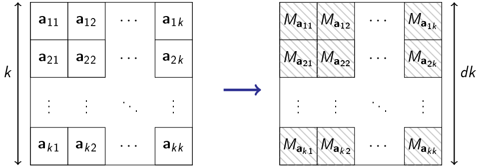
$R = \mathbb{Z}[X]/P(X)$



Canonical embedding

Reminder

$R = \mathbb{Z}[X]/P(X)$



dimension k over R dimension dk over $\mathbb{Z}$

Coefficient embedding

$$\begin{array}{rcl} \sigma : \qquad\qquad R & \to & \mathbb{R}^d \\ \mathbf{a} = a_0 + a_1 X + \cdots + a_{d-1}X^{d-1} & \mapsto & (a_0, a_1, \cdots, a_{d-1})^T \end{array}$$

$$\mathbf{a} \longrightarrow M_{\mathbf{a}} = \begin{pmatrix} | & | & & | \\ \sigma(\mathbf{a}) & \sigma(X\mathbf{a}) & \cdots & \sigma(X^{d-1}\mathbf{a}) \\ | & | & & | \end{pmatrix}$$

Canonical embedding

Reminder

$R = \mathbb{Z}[X]/P(X)$

$\alpha_1, \cdots, \alpha_d$ roots of P

$$k \left\updownarrow \begin{array}{|c|c|c|c|} \hline \mathbf{a}_{11} & \mathbf{a}_{12} & \cdots & \mathbf{a}_{1k} \\ \hline \mathbf{a}_{21} & \mathbf{a}_{22} & \cdots & \mathbf{a}_{2k} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{a}_{k1} & \mathbf{a}_{k2} & \cdots & \mathbf{a}_{kk} \\ \hline \end{array} \right. \longrightarrow \left. \begin{array}{|c|c|c|c|} \hline M_{\mathbf{a}_{11}} & M_{\mathbf{a}_{12}} & \cdots & M_{\mathbf{a}_{1k}} \\ \hline M_{\mathbf{a}_{21}} & M_{\mathbf{a}_{22}} & \cdots & M_{\mathbf{a}_{2k}} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline M_{\mathbf{a}_{k1}} & M_{\mathbf{a}_{k2}} & \cdots & M_{\mathbf{a}_{kk}} \\ \hline \end{array} \right\updownarrow dk$$

dimension k over R dimension dk over $\mathbb{Z}$

Canonical embedding

$$\begin{array}{rrcl} \sigma : & R & \to & \mathbb{C}^d \\ & \mathbf{a} = a_0 + a_1 X + \cdots + a_{d-1} X^{d-1} & \mapsto & (\mathbf{a}(\alpha_1), \cdots, \mathbf{a}(\alpha_n))^T \end{array}$$

$$\mathbf{a} \longrightarrow M_{\mathbf{a}} = \begin{pmatrix} | & | & & | \\ \sigma(\mathbf{a}) & \sigma(X\mathbf{a}) & \cdots & \sigma(X^{d-1}\mathbf{a}) \\ | & | & & | \end{pmatrix}$$

Outline of the talk

High-level overview of LLL

LLL over $\mathbb{Z}$

- γ'-SVP in dim k $\leq$ 1-SVP in dim 2
  - $\gamma' = 2^{O(k)}$
  - poly time

[LLL82] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen.

High-level overview of LLL

LLL over $\mathbb{Z}$

- γ'-SVP in dim k
  $\leq$ 1-SVP in dim 2
  - $\gamma' = 2^{O(k)}$
  - poly time

- Lagrange-Gauss algo
  for SVP in dim 2
  - poly time

[LLL82] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen.

High-level overview of LLL

LLL over $\mathbb{Z}$

- γ'-SVP in dim k $\leq$ 1-SVP in dim 2
  - $\gamma' = 2^{O(k)}$
  - poly time
- Lagrange-Gauss algo for SVP in dim 2
  - poly time

LLL over $R = \mathbb{Z}[X]/(X^d + 1)$

- γ'-SVP in rank-k $\leq \gamma$-SVP in rank-2
  - $\gamma' = (\gamma d)^{O(k)}$
  - poly time

[LLL82] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen.

High-level overview of LLL

LLL over $\mathbb{Z}$

- γ'-SVP in dim k
  $\leq$ 1-SVP in dim 2
  - $\gamma' = 2^{O(k)}$
  - poly time
- Lagrange-Gauss algo for SVP in dim 2
  - poly time

LLL over $R = \mathbb{Z}[X]/(X^d + 1)$

- γ'-SVP in rank-k
  $\leq \gamma$-SVP in rank-2
  - $\gamma' = (\gamma d)^{O(k)}$
  - poly time
- Algorithm for γ-SVP in rank-2
  - $\gamma = 2^{(\log d)^{O(1)}}$
  - heuristic, quantum
  - poly time if oracle solving CVP in a fixed lattice
    (depending only on R)

[LLL82] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen.

High-level overview of LLL

LLL over $\mathbb{Z}$

- γ'-SVP in dim k $\leq$ 1-SVP in dim 2
  - $\gamma' = 2^{O(k)}$
  - poly time

- Lagrange-Gauss algo for SVP in dim 2
  - poly time

LLL over $R = \mathbb{Z}[X]/(X^d + 1)$

- γ'-SVP in rank-k $\leq \gamma$-SVP i
  -
  - time

needs QR-factorisation

- Algorithm for γ-SVP in rank-2
  - $\gamma = 2^{(\log d)^{O(1)}}$
  - heu
  - poly oracle solving CVP in a fixed lattice (depending only on R)

next section

[LLL82] A. K. Lenstra, H. W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen.

Inner product over R

For $\vec{a} = (a_1, \cdots, a_k) \in K^k$ and $\vec{b} = (b_1, \cdots, b_k) \in K^k$,

$$\boxed{\langle \vec{a}, \vec{b} \rangle_K = \sum_i a_i \overline{b_i} \in K}$$

Inner product over R

For $\vec{a} = (a_1, \cdots, a_k) \in K^k$ and $\vec{b} = (b_1, \cdots, b_k) \in K^k$,

$$\boxed{\langle \vec{a}, \vec{b} \rangle_K = \sum_i a_i \overline{b_i} \in K}$$

$\|\vec{a}\|_K := \sqrt{\langle \vec{a}, \vec{a} \rangle_K} \in K$ (or $K_{\mathbb{R}}$)

Inner product over R

For $\vec{a} = (a_1, \cdots, a_k) \in K^k$ and $\vec{b} = (b_1, \cdots, b_k) \in K^k$,

$$\boxed{\langle \vec{a}, \vec{b} \rangle_K = \sum_i a_i \overline{b_i} \in K}$$

$\|\vec{a}\|_K := \sqrt{\langle \vec{a}, \vec{a} \rangle_K} \in K$ (or $K_{\mathbb{R}}$)

Properties

- $\mathrm{Tr}(\|\vec{a}\|_K^2) = \|\sigma(\vec{a})\|_2^2$

$\mathrm{Tr}(x) = \sum_{i=1}^d \sigma(x)_i$

Inner product over R

For $\vec{a} = (a_1, \cdots, a_k) \in K^k$ and $\vec{b} = (b_1, \cdots, b_k) \in K^k$,

$$\boxed{\langle \vec{a}, \vec{b} \rangle_K = \sum_i a_i \overline{b_i} \in K}$$

$\|\vec{a}\|_K := \sqrt{\langle \vec{a}, \vec{a} \rangle_K} \in K$ (or $K_{\mathbb{R}}$)

Properties

- $\mathrm{Tr}(\|\vec{a}\|_K^2) = \|\sigma(\vec{a})\|_2^2$
- $\mathcal{N}(\|\vec{a}\|_K^2) = \Delta_K^{-1} \cdot \det(\mathcal{L}(\vec{a}))^2$

$\mathcal{N}(x) = \prod_{i=1}^d \sigma(x)_i$

QR factorization over R

Let $B = (b_1, \cdots, b_k) \in K^k$, define

$$b_i^* = b_i - \sum_{j<i} \mu_{ij} b_j^*, \;\; \text{with } \mu_{ij} = \frac{\langle b_i, b_j^* \rangle_K}{\langle b_j^*, b_j^* \rangle_K}$$

QR factorization over R

Let $B = (b_1, \cdots, b_k) \in K^k$, define

$$b_i^* = b_i - \sum_{j<i} \mu_{ij} b_j^*, \quad \text{with } \mu_{ij} = \frac{\langle b_i, b_j^* \rangle_K}{\langle b_j^*, b_j^* \rangle_K}$$

QR-factorisation: $B = QR$, with

- $r_{ii} = \|b_i^*\|_K$, $r_{ij} = \mu_{ji} r_{ii}$ for $i < j$ and $r_{ij} = 0$ otherwise
- columns of Q are $b_i^* / \|b_i^*\|_K$

QR factorization over R

Let $B = (b_1, \cdots, b_k) \in K^k$, define

$$b_i^* = b_i - \sum_{j<i} \mu_{ij} b_j^*, \ \text{ with } \mu_{ij} = \frac{\langle b_i, b_j^* \rangle_K}{\langle b_j^*, b_j^* \rangle_K}$$

QR-factorisation: $B = QR$, with

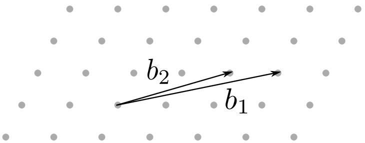
- $r_{ii} = \|b_i^*\|_K$, $r_{ij} = \mu_{ji} r_{ii}$ for $i < j$ and $r_{ij} = 0$ otherwise
- columns of Q are $b_i^* / \|b_i^*\|_K$

Properties

- $\overline{Q}^T Q = I_k$
- $\det_K(B) = \prod_i r_{ii}$ and $\det(\mathcal{L}(B)) = \det(\mathcal{L}(R))$
- $\forall v \in \mathcal{L}(B), \ \mathcal{N}(\|v\|_K) \geq \min \mathcal{N}(r_{ii})$

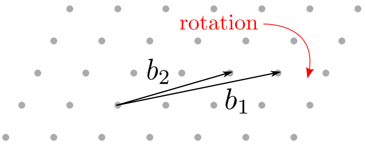
Outline of the talk

Lagrange-Gauss algorithm (over $\mathbb{Z}$)



$$M = \begin{pmatrix} 10 & 7 \\ 2 & 2 \end{pmatrix}$$

Lagrange-Gauss algorithm (over $\mathbb{Z}$)

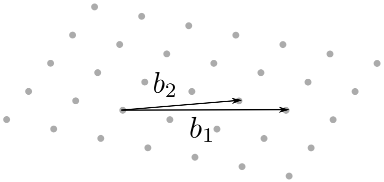


rotation

$$M = \begin{pmatrix} 10 & 7 \\ 2 & 2 \end{pmatrix}$$

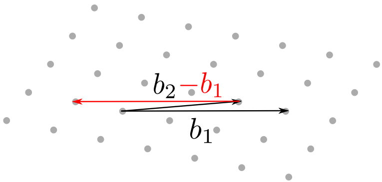
Compute QR factorization

Lagrange-Gauss algorithm (over $\mathbb{Z}$)



$$M = \begin{pmatrix} 10.2 & 7.3 \\ 0 & 0.6 \end{pmatrix}$$

Lagrange-Gauss algorithm (over $\mathbb{Z}$)

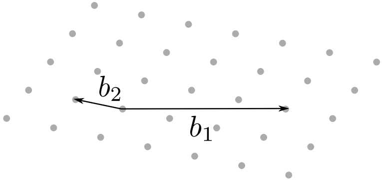


reduce b_2 with b_1

$$M = \begin{pmatrix} 10.2 & 7.3 \\ 0 & 0.6 \end{pmatrix}$$

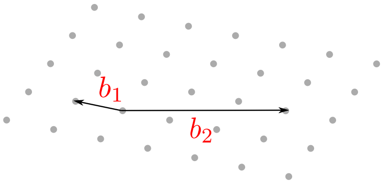
"Euclidean division" (over $\mathbb{R}$)
of 7.3 by 10.2

Lagrange-Gauss algorithm (over $\mathbb{Z}$)



$$M = \begin{pmatrix} 10.2 & -2.9 \\ 0 & 0.6 \end{pmatrix}$$

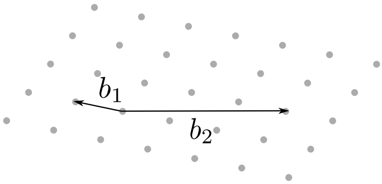
Lagrange-Gauss algorithm (over $\mathbb{Z}$)



swap

$$M = \begin{pmatrix} -2.9 & 10.2 \\ 0.6 & 0 \end{pmatrix}$$

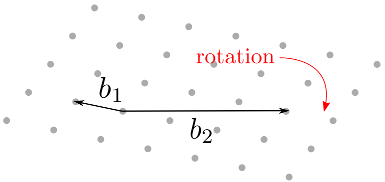
Lagrange-Gauss algorithm (over $\mathbb{Z}$)



start again

$$M = \begin{pmatrix} -2.9 & 10.2 \\ 0.6 & 0 \end{pmatrix}$$

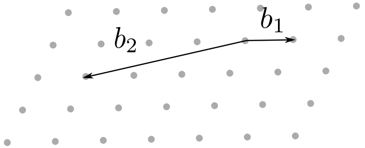
Lagrange-Gauss algorithm (over $\mathbb{Z}$)



rotation

$$M = \begin{pmatrix} -2.9 & 10.2 \\ 0.6 & 0 \end{pmatrix}$$

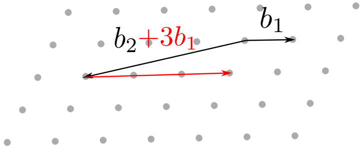
Lagrange-Gauss algorithm (over $\mathbb{Z}$)



rotation

$$M = \begin{pmatrix} 3 & -10 \\ 0 & -2 \end{pmatrix}$$

Lagrange-Gauss algorithm (over $\mathbb{Z}$)

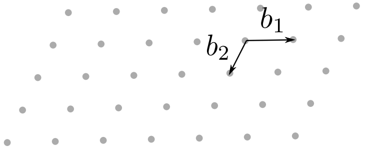


reduce b_2 with b_1

$$M = \begin{pmatrix} 3 & -10 \\ 0 & -2 \end{pmatrix}$$

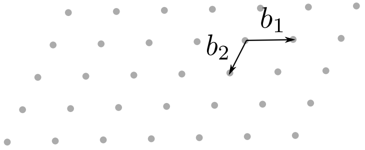
"Euclidean division" (over $\mathbb{R}$) of -10 by 3

Lagrange-Gauss algorithm (over $\mathbb{Z}$)



$$M = \begin{pmatrix} 3 & -1 \\ 0 & -2 \end{pmatrix}$$

Lagrange-Gauss algorithm (over $\mathbb{Z}$)

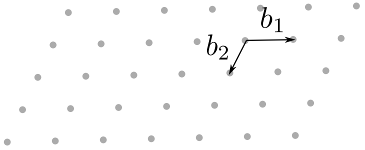


$$M = \begin{pmatrix} 3 & -1 \\ 0 & -2 \end{pmatrix}$$

For the Lagrange-Gauss algorithm over R, we need

- Rotation
- Euclidean division

Lagrange-Gauss algorithm (over $\mathbb{Z}$)



$$M = \begin{pmatrix} 3 & -1 \\ 0 & -2 \end{pmatrix}$$

For the Lagrange-Gauss algorithm over R, we need

- Rotation $\Rightarrow$ ok
- Euclidean division $\Rightarrow$?

Euclidean division

Over $\mathbb{Z}$

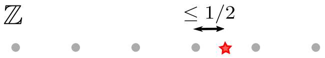
Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target $-b/a$.

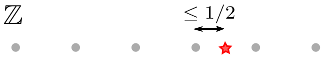


Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target $-b/a$.



Over R

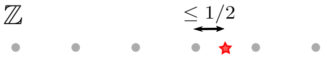
CVP in R with target $-b/a$
$\Rightarrow$ output $r \in R$

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

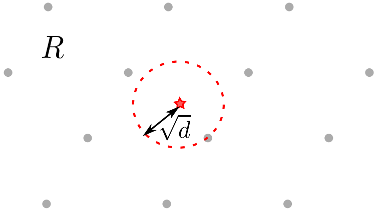
CVP in $\mathbb{Z}$ with target $-b/a$.



Over R

CVP in R with target $-b/a$
$\Rightarrow$ output $r \in R$

Difficulty: Typically
$\|b + ra\| \approx \sqrt{d} \cdot \|a\| \gg \|a\|$.



Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target $-b/a$.

Over R

CVP in R with target $-b/a$
$\Rightarrow$ output $r \in R$

Difficulty: Typically
$\|b + ra\| \approx \sqrt{d} \cdot \|a\| \gg \|a\|$.

Relax the requirement

Find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$

Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target $-b/a$.

Over R

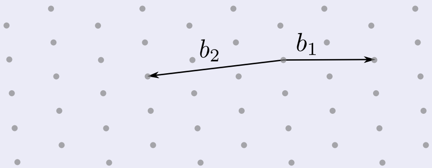
CVP in R with target $-b/a$
$\Rightarrow$ output $r \in R$

Difficulty: Typically
$\|b + ra\| \approx \sqrt{d} \cdot \|a\| \gg \|a\|$.

Relax the requirement

Find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$



Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target $-b/a$.

Over R

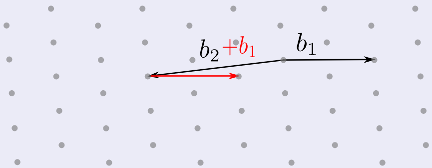
CVP in R with target $-b/a$
$\Rightarrow$ output $r \in R$

Difficulty: Typically
$\|b + ra\| \approx \sqrt{d} \cdot \|a\| \gg \|a\|$.

Relax the requirement

Find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$



Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target $-b/a$.

Over R

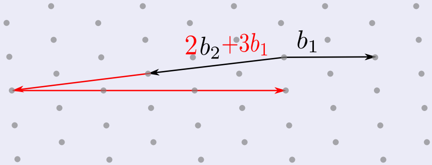
CVP in R with target $-b/a$
$\Rightarrow$ output $r \in R$

Difficulty: Typically
$\|b + ra\| \approx \sqrt{d} \cdot \|a\| \gg \|a\|$.

Relax the requirement

Find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$



Euclidean division

Over $\mathbb{Z}$

Input: $a, b \in \mathbb{Z}$, $a \neq 0$
Output: $r \in \mathbb{Z}$
such that $|b + ra| \leq |a|/2$

CVP in $\mathbb{Z}$ with target $-b/a$.

Over R

CVP in R with target $-b/a$
$\Rightarrow$ output $r \in R$

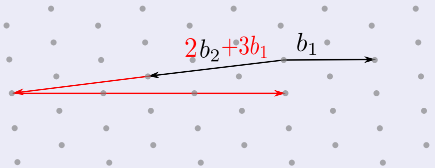
Difficulty: Typically
$\|b + ra\| \approx \sqrt{d} \cdot \|a\| \gg \|a\|$.

Relax the requirement

Find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$

$\Rightarrow$ sufficient for Gauss' algo



Outline of the talk

The Log space

Reminder: $\sigma(r) = (r(\alpha_1), \cdots, r(\alpha_d))^T$

(multiplication is coefficient-wise)

The Log space

Reminder: $\sigma(r) = (r(\alpha_1), \cdots, r(\alpha_d))^T$

(multiplication is coefficient-wise)

$$\boxed{\mathrm{Log}(r) = (\log|r(\alpha_1)|, \cdots, \log|r(\alpha_d)|)^T}$$

The Log space

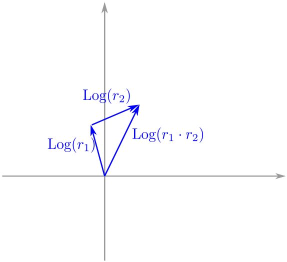
Reminder: $\sigma(r) = (r(\alpha_1), \cdots, r(\alpha_d))^T$

(multiplication is coefficient-wise)

$$\boxed{\mathrm{Log}(r) = (\log|r(\alpha_1)|, \cdots, \log|r(\alpha_d)|)^T}$$

Properties of Log

- $\mathrm{Log}(r_1 \cdot r_2) = \mathrm{Log}(r_1) + \mathrm{Log}(r_2)$



The Log space

Reminder: $\sigma(r) = (r(\alpha_1), \cdots, r(\alpha_d))^T$

(multiplication is coefficient-wise)

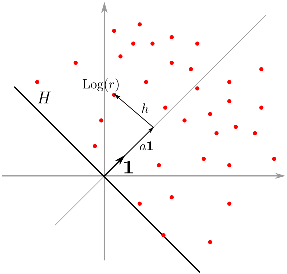
$$\boxed{\mathrm{Log}(r) = (\log|r(\alpha_1)|, \cdots, \log|r(\alpha_d)|)^T}$$

Let $\mathbf{1} = (1, \cdots, 1)$ and $H = \mathbf{1}^\perp$

Properties of Log

$\mathrm{Log}\, r = h + a\mathbf{1}$, with $h \in H$

- $\mathrm{Log}(r_1 \cdot r_2) = \mathrm{Log}(r_1) + \mathrm{Log}(r_2)$
- $a \geq 0$ if $r \in R$



The Log space

Reminder: $\sigma(r) = (r(\alpha_1), \cdots, r(\alpha_d))^T$

(multiplication is coefficient-wise)

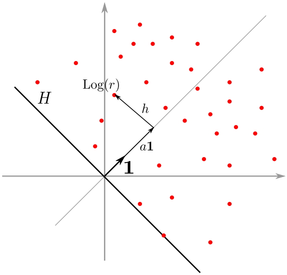
$$\boxed{\mathrm{Log}(r) = (\log|r(\alpha_1)|, \cdots, \log|r(\alpha_d)|)^T}$$

Let $\mathbf{1} = (1, \cdots, 1)$ and $H = \mathbf{1}^\perp$

Properties of Log

$\mathrm{Log}\, r = h + a\mathbf{1}$, with $h \in H$

- $\mathrm{Log}(r_1 \cdot r_2) = \mathrm{Log}(r_1) + \mathrm{Log}(r_2)$
- $a \geq 0$ if $r \in R$
- $\|r\| \simeq 2^{\|\mathrm{Log}\, r\|_\infty}$



Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$

Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa + yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$

Difficulty: Log works well with $\times$, but not with $+$

Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa - yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$

Difficulty: Log works well with $\times$, but not with $+$

Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa - yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$

Difficulty: Log works well with $\times$, but not with $+$

Solution: If $\|\operatorname{Log}(u) - \operatorname{Log}(v)\| \leq \varepsilon$
then $\|u - v\| \lesssim \varepsilon \cdot \min(\|u\|, \|v\|)$
(requires to extend Log to take arguments into account)

Using the Log space

Objective: find $x, y \in R$ such that

- $\|xa - yb\| \leq \|a\|/2$
- $\|y\| \leq \mathrm{poly}(d)$

Difficulty: Log works well with $\times$, but not with $+$

Solution: If $\|\operatorname{Log}(u) - \operatorname{Log}(v)\| \leq \varepsilon$
then $\|u - v\| \lesssim \varepsilon \cdot \min(\|u\|, \|v\|)$
(requires to extend Log to take arguments into account)

New objective

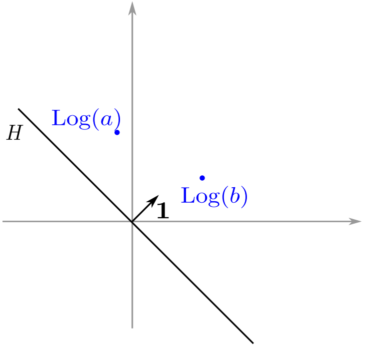
Find $x, y \in R$ such that

- $\|\operatorname{Log}(xa) - \operatorname{Log}(yb)\| \leq \varepsilon$
- $\|\operatorname{Log}(y)\|_\infty \leq O(\log d)$

Idea

Objective: find $x, y \in R$ s.t.

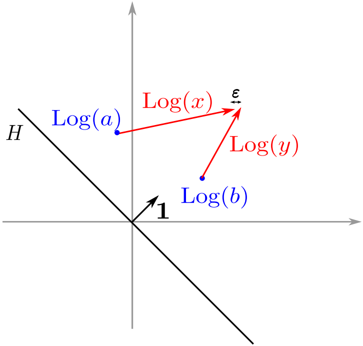
- $\| \mathrm{Log}(xa) - \mathrm{Log}(yb) \| \le \varepsilon$
- $\| \mathrm{Log}(y) \|_\infty \le O(\log d)$



Idea

Objective: find $x, y \in R$ s.t.

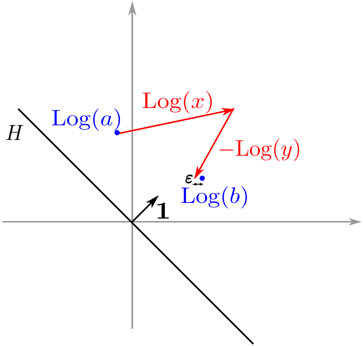
- $\| \mathrm{Log}(xa) - \mathrm{Log}(yb) \| \leq \varepsilon$
- $\| \mathrm{Log}(y) \|_\infty \leq O(\log d)$



Idea

Objective: find $x, y \in R$ s.t.

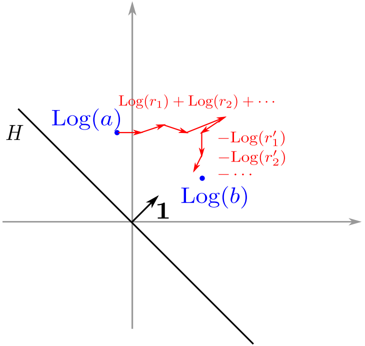
- $\|(\mathrm{Log}(x) - \mathrm{Log}(y)) - \mathrm{Log}(b/a)\| \leq \varepsilon$
- $\|\mathrm{Log}(y)\|_\infty \leq O(\log d)$



Idea

Objective: find $x, y \in R$ s.t.

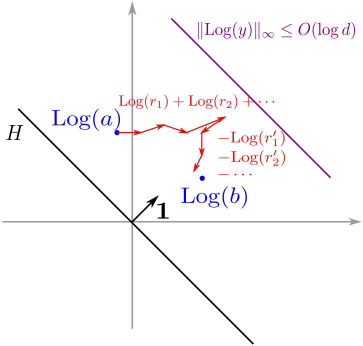
- $\|(\mathrm{Log}(x) - \mathrm{Log}(y)) - \mathrm{Log}(b/a)\| \leq \varepsilon$
- $\|\mathrm{Log}(y)\|_\infty \leq O(\log d)$



Idea

Objective: find $x, y \in R$ s.t.

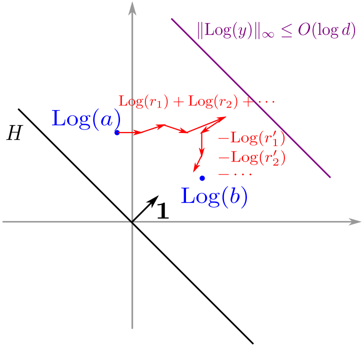
- $\|(\mathrm{Log}(x) - \mathrm{Log}(y)) - \mathrm{Log}(b/a)\| \leq \varepsilon$
- $\|\mathrm{Log}(y)\|_\infty \leq O(\log d)$



Idea

Objective: find $x, y \in R$ s.t.

- $\|(\mathrm{Log}(x) - \mathrm{Log}(y)) - \mathrm{Log}(b/a)\| \leq \varepsilon$
- $\|\mathrm{Log}(y)\|_\infty \leq O(\log d)$



Solve **exact** CVP in L with target t

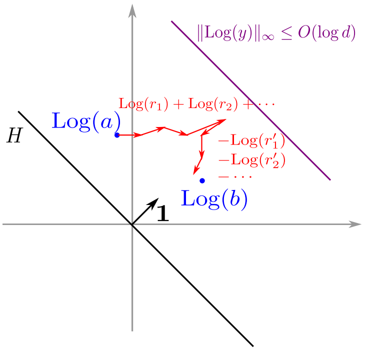
$$L = \begin{pmatrix} \mathrm{Log}\, r_1 & \cdots & \mathrm{Log}\, r_{d^2} \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}, \quad t = \begin{pmatrix} \mathrm{Log}(b/a) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(L is fixed and independent of a and b)

Idea

Objective: find $x, y \in R$ s.t.

- $\|(\mathrm{Log}(x) - \mathrm{Log}(y)) - \mathrm{Log}(b/a)\| \leq \varepsilon$
- $\|\,\mathrm{Log}(y)\|_\infty \leq O(\log d)$



Solve **exact** CVP in L with target t with an oracle

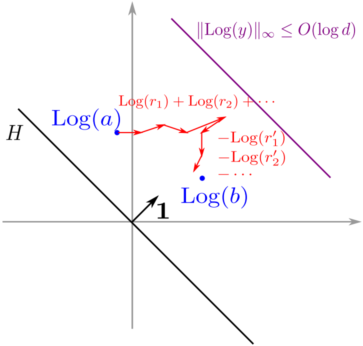
$$L = \begin{pmatrix} \mathrm{Log}\, r_1 & \cdots & \mathrm{Log}\, r_{d^2} \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}, \quad t = \begin{pmatrix} \mathrm{Log}(b/a) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(L is fixed and independent of a and b)

Idea

Objective: find $x, y \in R$ s.t.

- $\|(\mathrm{Log}(x) - \mathrm{Log}(y)) - \mathrm{Log}(b/a)\| \leq \varepsilon$
- $\|\mathrm{Log}(y)\|_\infty \leq O(\log d)$



Solve **exact** CVP in L with target t with an oracle

$$L = \begin{pmatrix} \mathrm{Log}\, r_1 & \cdots & \mathrm{Log}\, r_{d^2} \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}, \quad t = \begin{pmatrix} \mathrm{Log}(b/a) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

(L is fixed and independent of a and b)

Complexity

Poly time
(with the oracle)

Under the carpet

- Heuristics
  - maths justification
  - numerical experiments (in very small dimension)
- Any module / ideal
  - use pseudo-basis
  - add units and class group to L (cf [Buc88])
  - decompose ideals in class group $\Rightarrow$ quantum algo
- Full LLL algo over R
  - Lovász' swap condition
  - switch between $\mathcal{N}(\cdot)$ and $\|\cdot\|$
  - handling bit sizes

[Buc88] J. Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. Séminaire de théorie des nombres.

Summary and impact

LLL algorithm for power-of-two cyclotomic fields

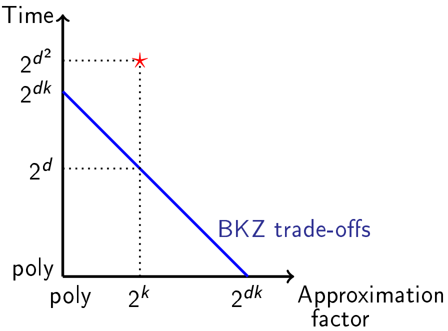
- Approx: quasi-poly(d)$^{O(k)} = 2^{(\log d)^{O(1)} \cdot k}$
- Time: quantum polynomial time
  if oracle solving CVP in L (of dim $O(d^{2+\varepsilon})$)

Summary and impact

LLL algorithm for power-of-two cyclotomic fields

- Approx: quasi-poly(d)$^{O(k)} = 2^{(\log d)^{O(1)} \cdot k}$
- Time: quantum polynomial time
  if oracle solving CVP in L (of dim $O(d^{2+\varepsilon})$)

In practice? $\Rightarrow$ replace the oracle by a CVP solver

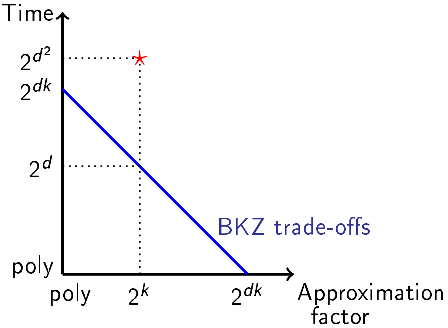


Summary and impact

LLL algorithm for power-of-two cyclotomic fields

- Approx: quasi-poly(d)$^{O(k)} = 2^{(\log d)^{O(1)} \cdot k}$
- Time: quantum polynomial time
  if oracle solving CVP in L (of dim $O(d^{2+\varepsilon})$)

In practice? $\Rightarrow$ replace the oracle by a CVP solver



$\Rightarrow$ theoretical result (not practical)

Conclusion

Open problems:

- Better understanding of the lattice L
  - reduce its dimension to $\tilde{O}(d)$?
  - prove the heuristics?
  - better CVP solver for L?

[MS19] T. Mukherjee, N. Stephens-Davidowitz. Lattice Reduction for Modules, or How to Reduce ModuleSVP to ModuleSVP, ePrint.

Conclusion

Open problems:

- Better understanding of the lattice L
  - reduce its dimension to $\tilde{O}(d)$?
  - prove the heuristics?
  - better CVP solver for L?
- Generalizing LLL to all the BKZ trade-offs?
  - reduction rank k to rank β (cf [MS19])
  - sieving/enumeration in modules?

[MS19] T. Mukherjee, N. Stephens-Davidowitz. Lattice Reduction for Modules, or How to Reduce ModuleSVP to ModuleSVP, ePrint.

Conclusion

Open problems:

- Better understanding of the lattice L
  - reduce its dimension to $\tilde{O}(d)$?
  - prove the heuristics?
  - better CVP solver for L?
- Generalizing LLL to all the BKZ trade-offs?
  - reduction rank k to rank β (cf [MS19])
  - sieving/enumeration in modules?

Thank you

[MS19] T. Mukherjee, N. Stephens-Davidowitz. Lattice Reduction for Modules, or How to Reduce ModuleSVP to ModuleSVP, ePrint.