o=true; for(int z=0;o;z++) o=false; for(int j=0;j<=

Algebraic techniques for Algebraic lattices

Thomas Espitau

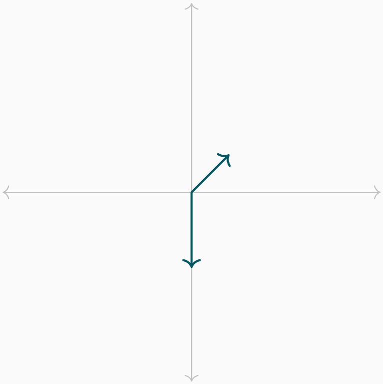
March 2, 2020, Simons Institute

NTT

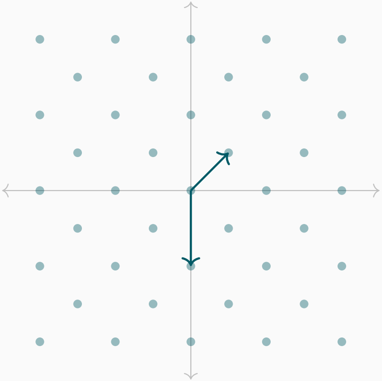
Lattices and LLL

What is this all about?

What is this all about?



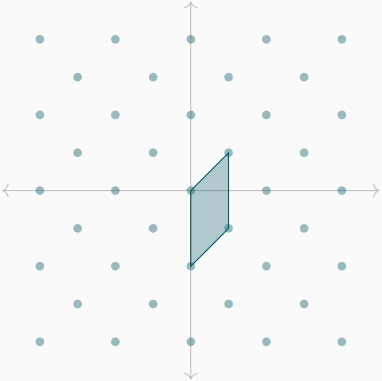
What is this all about?



Lattice

A (Euclidean) lattice Λ is a *discrete* subgroup of an Euclidean space (say $\mathbf{R}^n$).

What is this all about?



Lattice

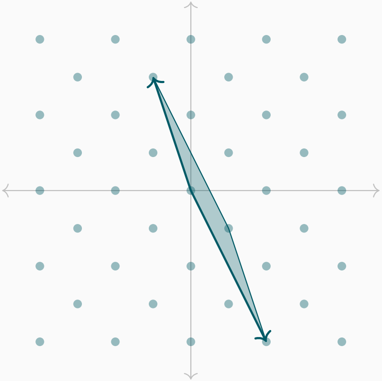
A (Euclidean) lattice Λ is a *discrete* subgroup of an Euclidean space (say $\mathbf{R}^n$).

The covolume $\mathrm{covol}(\Lambda)$ of Λ is the quantity

$$\mathrm{covol}(\Lambda) = \sqrt{\det \langle v_i, v_j \rangle}$$

Corresponds to the volume of the fundamental domain $\{\sum x_i v_i | 0 \leqslant x_i < 1\}$.

What is this all about?



Lattice

A (Euclidean) lattice Λ is a *discrete* subgroup of an Euclidean space (say $\mathbf{R}^n$).

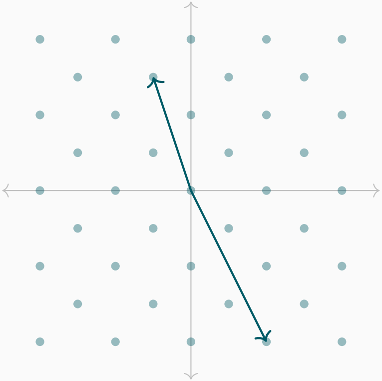
The covolume $\mathrm{covol}(\Lambda)$ of Λ is the quantity

$$\mathrm{covol}(\Lambda) = \sqrt{\det \langle v_i, v_j \rangle}$$

Corresponds to the volume of the fundamental domain $\{\sum x_i v_i | 0 \leqslant x_i < 1\}$.

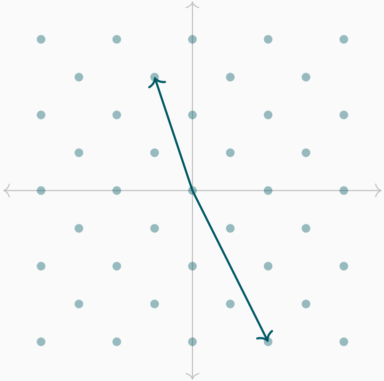
Independent of the basis

What is this all about?



How to get a shorter basis?

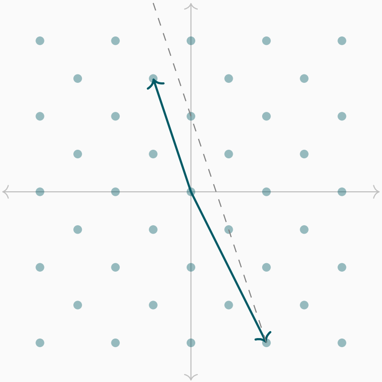
What is this all about?



How to get a shorter basis?

→ Use the shortest vector to reduce the longest one.

What is this all about?

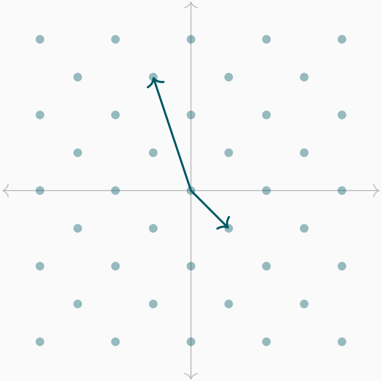


How to get a shorter basis?

$\rightarrow$ Use the shortest vector to reduce the longest one.

1. Take the *shortest* element in the coset

What is this all about?

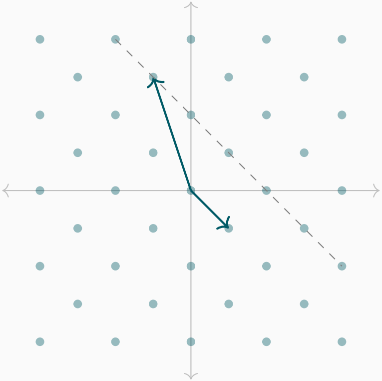


How to get a shorter basis?

$\rightarrow$ Use the shortest vector to reduce the longest one.

1. Take the *shortest* element in the coset

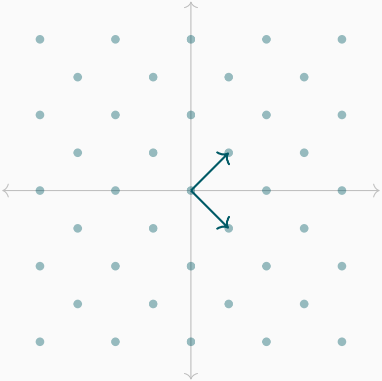
What is this all about?



How to get a shorter basis?

→ Use the shortest vector to reduce the longest one.

1. Take the *shortest* element in the coset
2. Repeat

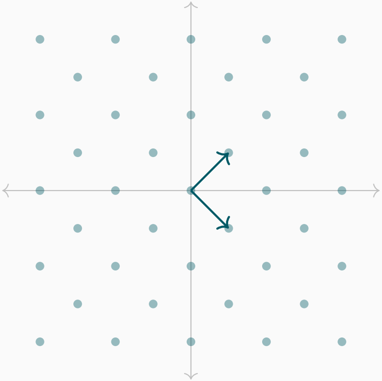


How to get a shorter basis?

→ Use the shortest vector to reduce the longest one.

1. Take the *shortest* element in the coset
2. Repeat

What is this all about?

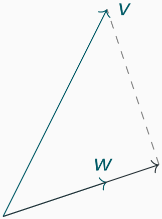


How to get a shorter basis?

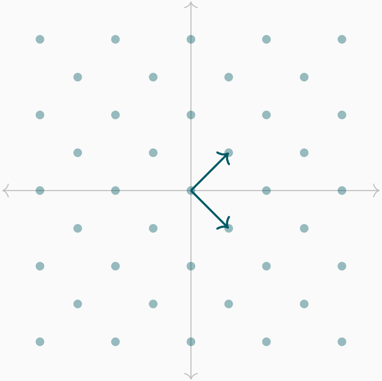
$\rightarrow$ Use the shortest vector to reduce the longest one.

An effective way of computing this element:

1. Orthogonal projection $\frac{\langle w,v\rangle}{\langle w,w\rangle} w$



What is this all about?

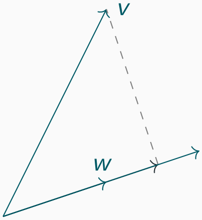


How to get a shorter basis?

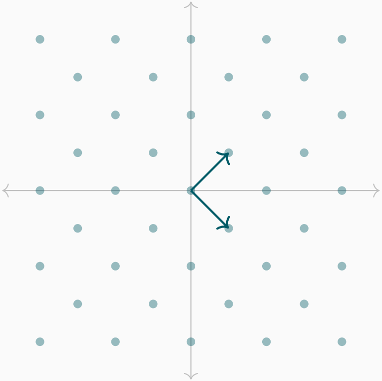
$\rightarrow$ Use the shortest vector to reduce the longest one.

An effective way of computing this element:

1. Orthogonal projection $\quad \frac{\langle w,v\rangle}{\langle w,w\rangle} w$
2. Round $\quad \left\lceil \frac{\langle w,v\rangle}{\langle w,w\rangle} \right\rfloor w$



What is this all about?



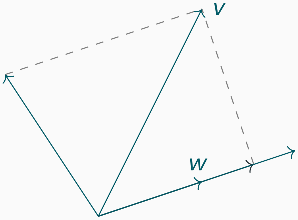
How to get a shorter basis?

$\rightarrow$ Use the shortest vector to reduce the longest one.

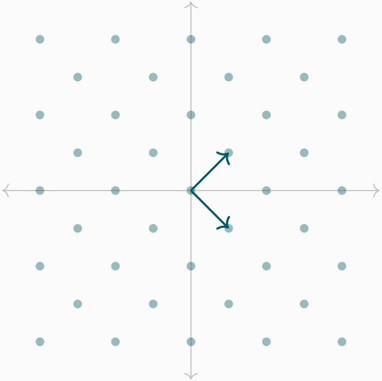
An effective way of computing this element:

1. Orthogonal projection $\frac{\langle w,v\rangle}{\langle w,w\rangle}w$
2. Round $\left\lceil\frac{\langle w,v\rangle}{\langle w,w\rangle}\right\rfloor w$
3. Substract

$$v - \left\lceil\frac{\langle w,v\rangle}{\langle w,w\rangle}\right\rfloor w$$



What is this all about?



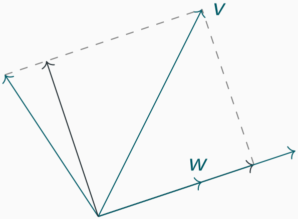
How to get a shorter basis?

$\rightarrow$ Use the shortest vector to reduce the longest one.

An effective way of computing this element:

1. Orthogonal projection $\frac{\langle w,v\rangle}{\langle w,w\rangle} w$
2. Round $\left\lceil \frac{\langle w,v\rangle}{\langle w,w\rangle} \right\rfloor w$
3. Substract

$$v - \left\lceil \frac{\langle w,v\rangle}{\langle w,w\rangle} \right\rfloor w$$



In dim 2... The Lagrange-Gauss reduction algorithm

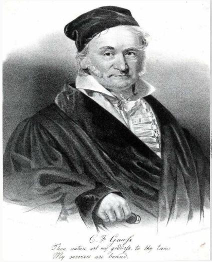
Gauss-Lagrange reduction

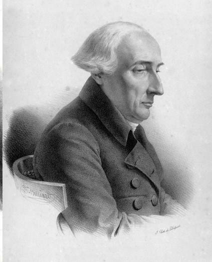
if $\|v\| < \|u\|$ **then return Gauss**(v, u);

$v' \leftarrow v - \left\lfloor \frac{\langle u,v \rangle}{\|u\|^2} \right\rceil u$;

if $\|v'\| < \|v\|$ **then return**
Gauss(u, v');

else return (u, v);





Gauss-Lagrange reduction

if $\|v\| < \|u\|$ **then return Gauss**(v, u);
$v' \leftarrow v - \left\lfloor \frac{\langle u,v \rangle}{\|u\|^2} \right\rceil u$;
if $\|v'\| < \|v\|$ **then return Gauss**(u, v');
else return (u, v);

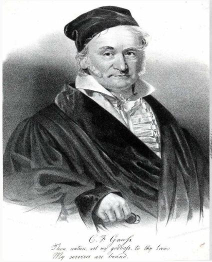
Properties of a Gauss-reduced basis (u, v)

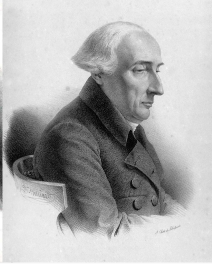
- $\|u\| \leqslant \|v\|$ and $|\langle u, v \rangle| \leqslant \frac{\|u\|^2}{2}$.

In dim 2... The Lagrange-Gauss reduction algorithm

Gauss-Lagrange reduction

if $\|v\| < \|u\|$ **then return** **Gauss**(v, u);
$v' \leftarrow v - \left\lfloor \frac{\langle u,v\rangle}{\|u\|^2} \right\rceil u$;
if $\|v'\| < \|v\|$ **then return** **Gauss**(u, v');
else return (u, v);





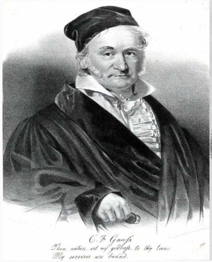
Properties of a Gauss-reduced basis (u, v)

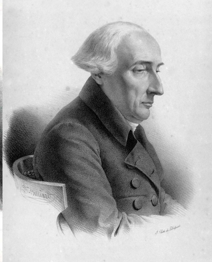
- $\|u\| \leqslant \|v\|$ and $|\langle u, v\rangle| \leqslant \frac{\|u\|^2}{2}$.
- u is a *shortest vector* of Λ

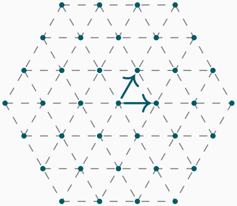
In dim 2... The Lagrange-Gauss reduction algorithm

Gauss-Lagrange reduction

if $\|v\| < \|u\|$ **then return** **Gauss**(v, u);
$v' \leftarrow v - \left\lfloor \frac{\langle u, v \rangle}{\|u\|^2} \right\rceil u$;
if $\|v'\| < \|v\|$ **then return** **Gauss**(u, v');
else return (u, v);







Properties of a Gauss-reduced basis (u, v)

- $\|u\| \leqslant \|v\|$ and $|\langle u, v\rangle| \leqslant \frac{\|u\|^2}{2}$.
- u is a *shortest vector* of Λ
- $\|u\|^2 \leqslant (4/3)\,\mathrm{covol}(\Lambda)$

And now what?

Minkowski theorem for first minima: For any lattice Λ of rank d,

$$\lambda_1(\Lambda) \leqslant \sqrt{d}\,\mathrm{covol}(\Lambda)^{\frac{1}{d}}$$

And now what?

Minkowski theorem for first minima: For any lattice Λ of rank d,

$$\lambda_1(\Lambda) \leqslant \sqrt{d}\,\mathrm{covol}(\Lambda)^{\frac{1}{d}}$$

And now what?

Finding a shortest/closest vector in a lattice is **hard**

And now what?

[LLL82] *There exists a* ***polynomial-time algorithm****, which given any lattice Λ, produces a vector in Λ of Euclidean length* ***at most*** *a factor of 2^d longer than a shortest vector.*

And now what?

[LLL82] *There exists a* ***polynomial-time algorithm****, which given any lattice Λ, produces a vector in Λ of Euclidean length* ***at most*** *a factor of 2^d longer than a shortest vector.*

- Simultaneous Diophantine approximation
  $\left| r_i - \frac{p_i}{q} \right| \leqslant \epsilon$

And now what?

[LLL82] *There exists a* ***polynomial-time algorithm****, which given any lattice Λ, produces a vector in Λ of Euclidean length* ***at most*** *a factor of 2^d longer than a shortest vector.*

- Simultaneous Diophantine approximation
  $\left|r_i - \frac{p_i}{q}\right| \leqslant \epsilon$
- Minimal polynomials of algebraic numbers
  $(r_i = r^i)$

And now what?

[LLL82] *There exists a* ***polynomial-time algorithm****, which given any lattice Λ, produces a vector in Λ of Euclidean length* ***at most*** *a factor of 2^d longer than a shortest vector.*

- Simultaneous Diophantine approximation
  $\left| r_i - \frac{p_i}{q} \right| \leqslant \epsilon$
- Minimal polynomials of algebraic numbers
  $(r_i = r^i)$
- Polynomial factorization over rationals
  Approximate a root r, find a minimal g vanishing at r.
- Cryptanalysis Knapsack problem , RSA for small public exponents, lattice-based cryptography...

And now what?

[LLL82] *There exists a* ***polynomial-time algorithm****, which given any lattice Λ, produces a vector in Λ of Euclidean length* ***at most*** *a factor of 2^d longer than a shortest vector.*

- Simultaneous Diophantine approximation $\left|r_i - \frac{p_i}{q}\right| \leqslant \epsilon$
- Minimal polynomials of algebraic numbers $(r_i = r^i)$
- Polynomial factorization over rationals Approximate a root r, find a minimal g vanishing at r.
- Cryptanalysis Knapsack problem , RSA for small public exponents, lattice-based cryptography...
- Computations in algebraic number theory (ideal computations, HNF, control of size of elements...)

Towards a polynomial-time reduction algorithm

Any basis $(v_1, \ldots, v_d)$ of a lattice Λ yields a filtration:

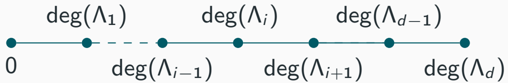
$$\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_{i-1} \subset \Lambda_i \subset \Lambda_{i+1} \subset \cdots \subset \Lambda_d = \Lambda$$

Towards a polynomial-time reduction algorithm

Any basis $(v_1, \ldots, v_d)$ of a lattice Λ yields a filtration:

$$\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_{i-1} \subset \Lambda_i \subset \Lambda_{i+1} \subset \cdots \subset \Lambda_d = \Lambda$$

Profile space

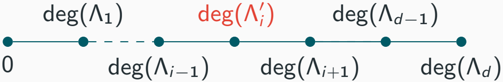


Any basis $(v_1, \ldots, v_d)$ of a lattice Λ yields a filtration:

$$\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_{i-1} \subset \Lambda_i \subset \Lambda_{i+1} \subset \cdots \subset \Lambda_d = \Lambda$$

Reduce the projected lattice with **Gauss algorithm**, **lift** and **replace**.

Profile space



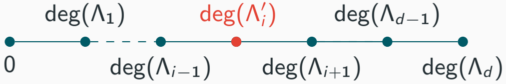
Towards a polynomial-time reduction algorithm

Any basis $(v_1, \ldots, v_d)$ of a lattice Λ yields a filtration:

$$\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_{i-1} \subset \Lambda_i \subset \Lambda_{i+1} \subset \cdots \subset \Lambda_d = \Lambda$$

Reduce the projected lattice with **Gauss algorithm**, **lift** and **replace**.

Profile space



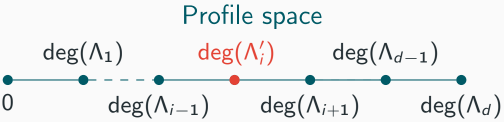
$\deg(\Lambda_i') \leqslant \deg(\Lambda_i)$

Towards a polynomial-time reduction algorithm

Any basis $(v_1, \ldots, v_d)$ of a lattice Λ yields a filtration:

$$\{0\} = \Lambda_0 \subset \Lambda_1 \subset \cdots \subset \Lambda_{i-1} \subset \Lambda_i \subset \Lambda_{i+1} \subset \cdots \subset \Lambda_d = \Lambda$$

Reduce the projected lattice with **Gauss algorithm**, **lift** and **replace**.

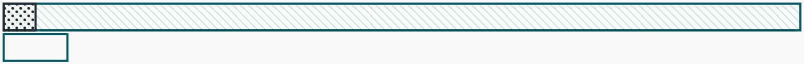


$$\deg(\Lambda'_i) \leqslant \deg(\Lambda_i)$$

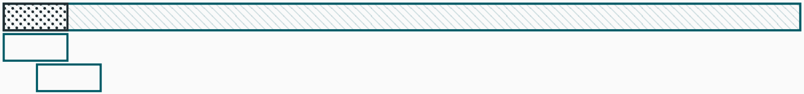
Gauss's reduction is a *local* tool for densifying the filtration

From local to global: an iterative strategy

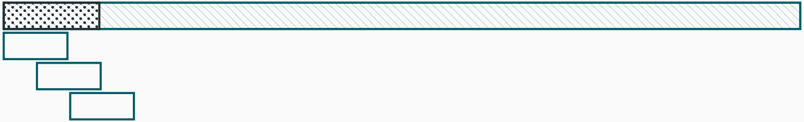
From local to global: an iterative strategy



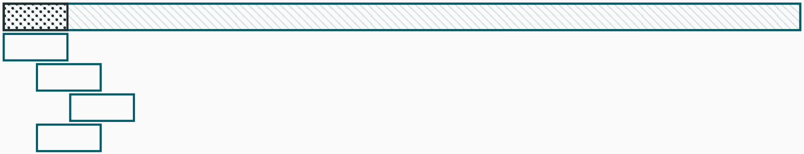
From local to global: an iterative strategy



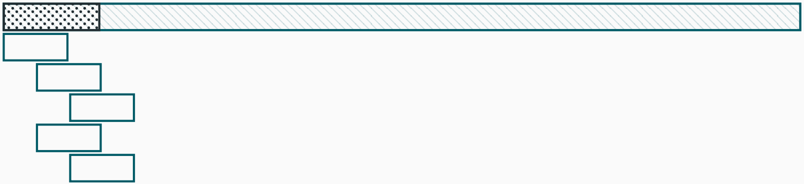
From local to global: an iterative strategy



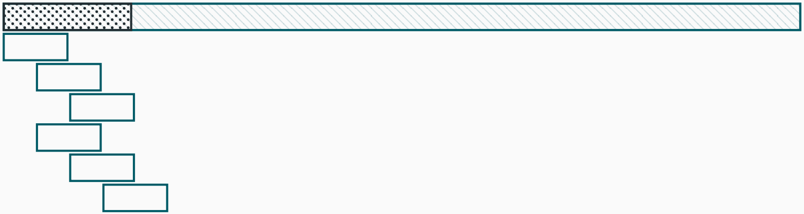
From local to global: an iterative strategy



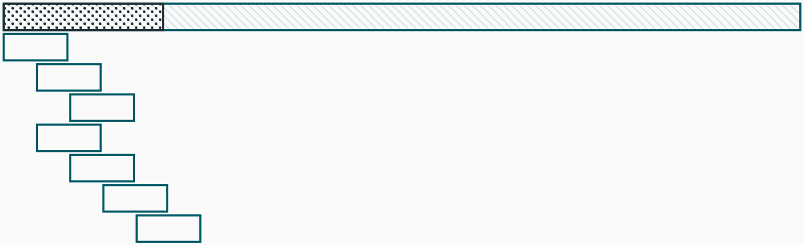
From local to global: an iterative strategy



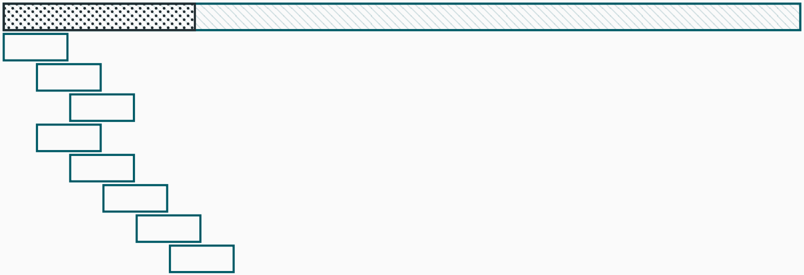
From local to global: an iterative strategy



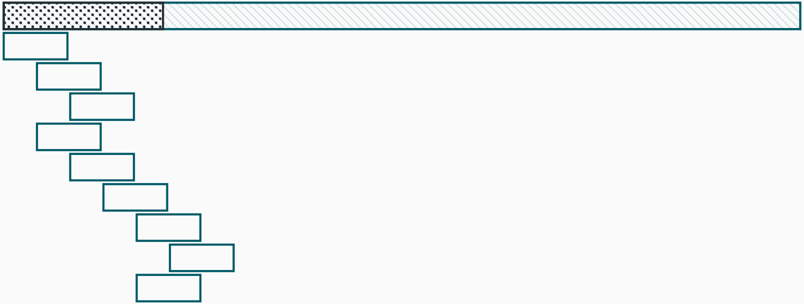
From local to global: an iterative strategy



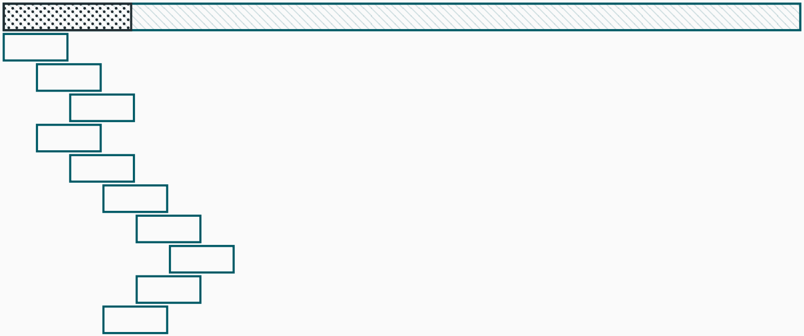
From local to global: an iterative strategy



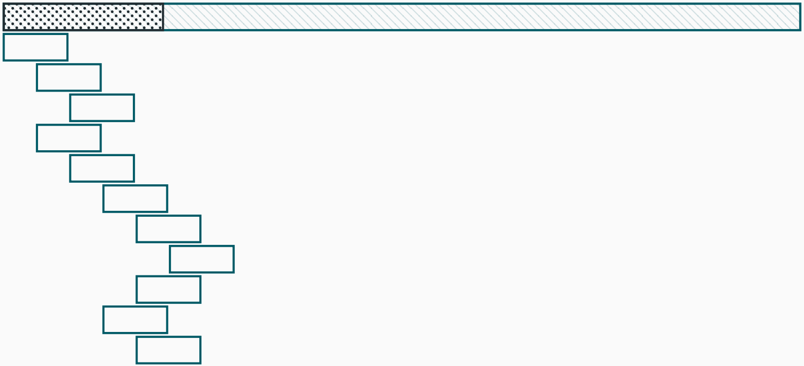
From local to global: an iterative strategy



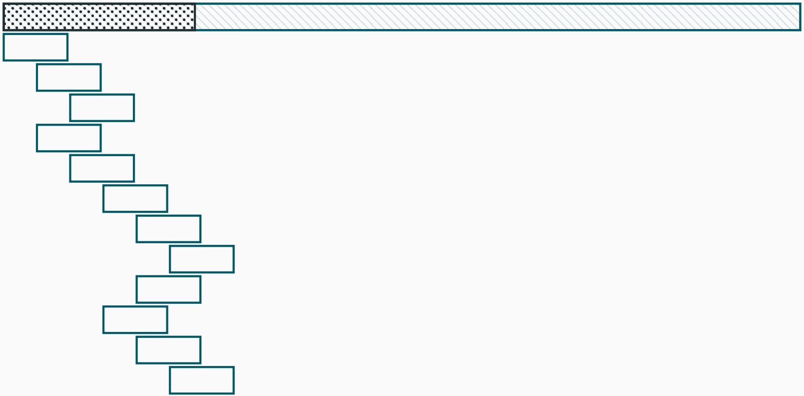
From local to global: an iterative strategy



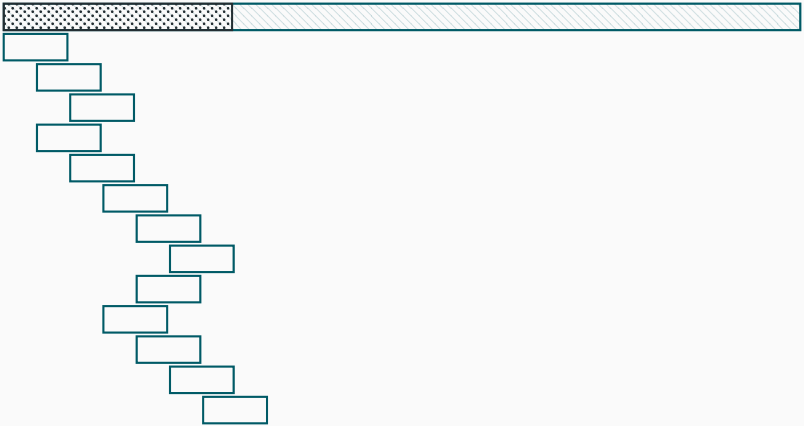
From local to global: an iterative strategy



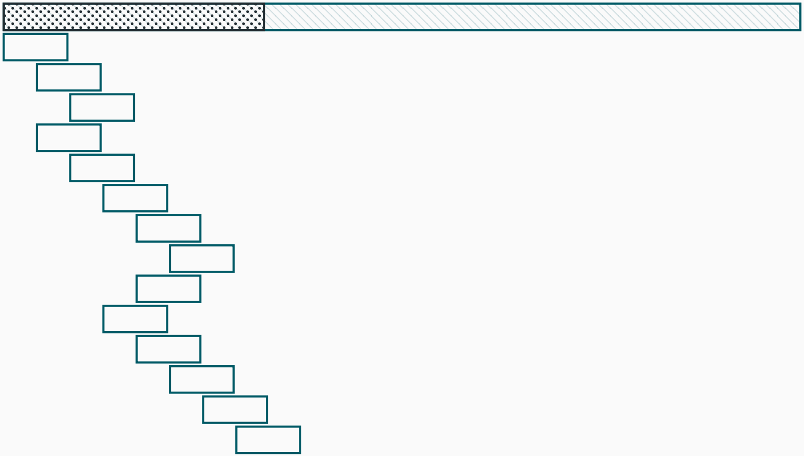
From local to global: an iterative strategy



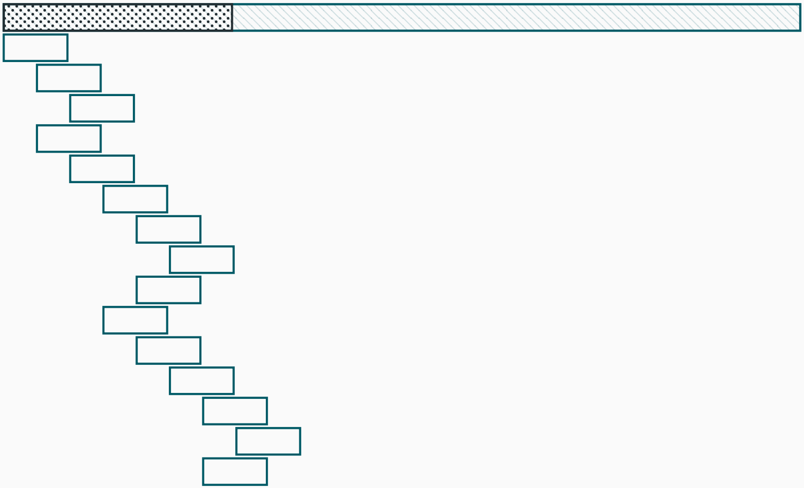
From local to global: an iterative strategy



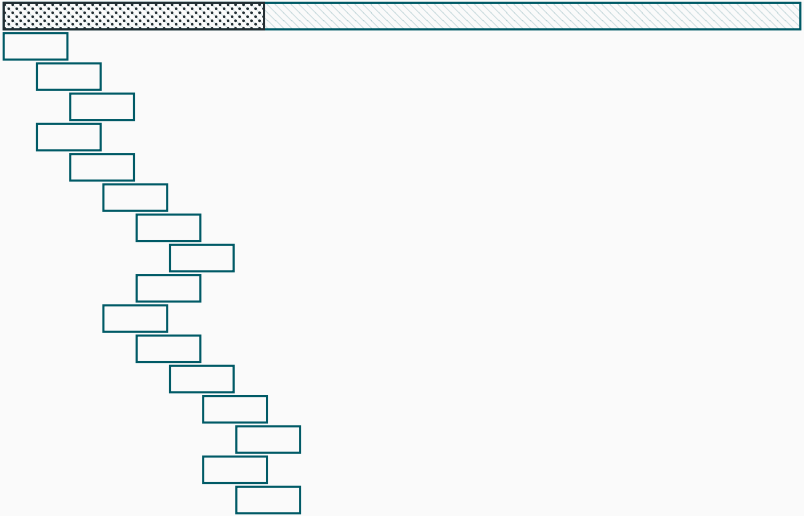
From local to global: an iterative strategy



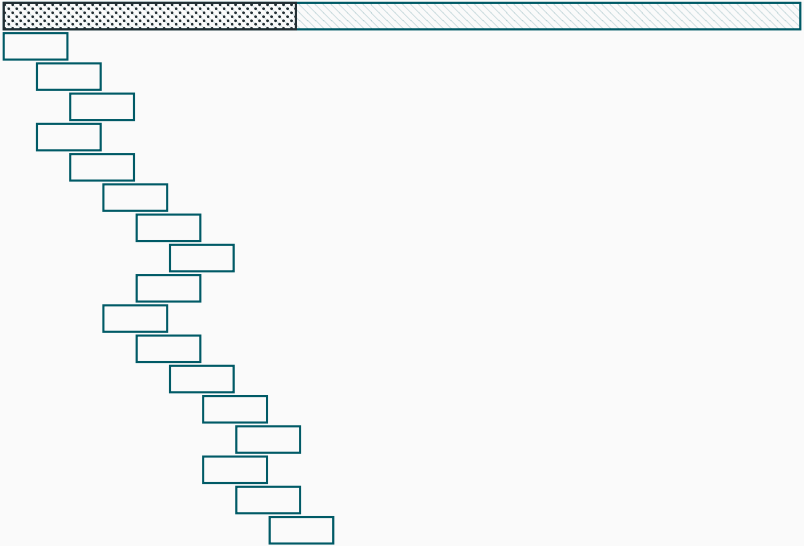
From local to global: an iterative strategy



From local to global: an iterative strategy



From local to global: an iterative strategy



Quantitatively...

- Algorithmic tools: QR decomposition, Size-reduction
- Original analysis:

$$O\left(d^6 B^3\right)$$

- If very precautious one can use floating-point representation
- L^2 [Nguyen-Stehlé:2009]:

$$O\left(d^5 B(d+B)\right)$$

- [Neumaier-Stehlé:2016] (recursive strategy):

$$O\left(d^{4+\epsilon} B^{1+\epsilon}\right)$$

Generalization: towards algebraic lattices

Number fields and algebraic lattices

Number field

- Finite extension of $\mathbf{Q}$:

$$L \cong {}^{\mathbf{Q}[X]}\!/\!_{(P)}$$

- Ring of integers:

$$\mathcal{O}_L = \{\alpha \mid \exists R \in \mathbf{Z}[X] \text{ monic }, R(\alpha) = 0\}$$

Number fields and algebraic lattices

Number field

- Finite extension of $\mathbf{Q}$:

$$L \cong {}^{\mathbf{Q}[X]}\!\diagup_{(P)}$$

- Ring of integers:

$$\mathcal{O}_L = \{\alpha \mid \exists R \in \mathbf{Z}[X] \text{ monic }, R(\alpha) = 0\}$$

For instance:

$$\mathcal{O}_{\mathbf{Q}} = \mathbf{Z}$$
$$\mathcal{O}_{\mathbf{Q}(i)} = \mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$$
$$\mathcal{O}_{\mathbf{Q}(j)} = \mathbf{Z}[i] = \{a + jb \mid a, b \in \mathbf{Z}\}$$

Number fields and algebraic lattices

Number field

- Finite extension of $\mathbf{Q}$:

$$L \cong \mathbf{Q}[X] / (P)$$

- Ring of integers:

$$\mathcal{O}_L = \{\alpha \mid \exists R \in \mathbf{Z}[X] \text{ monic }, R(\alpha) = 0\}$$

For instance:

$$\mathcal{O}_{\mathbf{Q}} = \mathbf{Z}$$

$$\mathcal{O}_{\mathbf{Q}(i)} = \mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$$

$$\mathcal{O}_{\mathbf{Q}(j)} = \mathbf{Z}[i] = \{a + jb \mid a, b \in \mathbf{Z}\}$$

Lattice

A (Euclidean) lattice Λ is a *discrete* subgroup of a Euclidean space (say $\mathbf{R}^n$).

Number fields and algebraic lattices

Number field

- Finite extension of $\mathbf{Q}$:

$$L \cong {\mathbf{Q}[X]}/{(P)}$$

- Ring of integers:

$$\mathcal{O}_L = \{\alpha \mid \exists R \in \mathbf{Z}[X] \text{ monic }, R(\alpha) = 0\}$$

For instance:

$$\mathcal{O}_{\mathbf{Q}} = \mathbf{Z}$$

$$\mathcal{O}_{\mathbf{Q}(i)} = \mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$$

$$\mathcal{O}_{\mathbf{Q}(j)} = \mathbf{Z}[i] = \{a + jb \mid a, b \in \mathbf{Z}\}$$

Lattice

A (Euclidean) lattice Λ is a free $\mathbf{Z}$-module of finite rank, endowed with an inner product on $\Lambda \otimes_{\mathbf{Z}} \mathbf{R}$.

Number fields and algebraic lattices

Number field

- Finite extension of $\mathbf{Q}$:

$$L \cong \mathbf{Q}[X] \diagup (P)$$

- Ring of integers:

$$\mathcal{O}_L = \{\alpha \mid \exists R \in \mathbf{Z}[X] \text{ monic }, R(\alpha) = 0\}$$

For instance:

$$\mathcal{O}_{\mathbf{Q}} = \mathbf{Z}$$

$$\mathcal{O}_{\mathbf{Q}(i)} = \mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$$

$$\mathcal{O}_{\mathbf{Q}(j)} = \mathbf{Z}[i] = \{a + jb \mid a, b \in \mathbf{Z}\}$$

Lattice

An (algebraic) lattice Λ is a free $\mathcal{O}_L$-module of finite rank, endowed with an inner product on $\Lambda \otimes_{\mathcal{O}_L} \mathbf{R}$.

Number fields and algebraic lattices

Number field

- Finite extension of $\mathbf{Q}$:

$$L \cong \mathbf{Q}[X]/(P)$$

- Ring of integers:

$$\mathcal{O}_L = \{\alpha \mid \exists R \in \mathbf{Z}[X] \text{ monic }, R(\alpha) = 0\}$$

For instance:

$$\mathcal{O}_{\mathbf{Q}} = \mathbf{Z}$$

$$\mathcal{O}_{\mathbf{Q}(i)} = \mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$$

$$\mathcal{O}_{\mathbf{Q}(j)} = \mathbf{Z}[i] = \{a + jb \mid a, b \in \mathbf{Z}\}$$

Lattice

An (algebraic) lattice Λ is a **free** $\mathcal{O}_L$-module of finite rank, endowed with an inner product on $\Lambda \otimes_{\mathcal{O}_L} \mathbf{R}$.

Number fields and algebraic lattices

Number field

- Finite extension of $\mathbf{Q}$:

$$L \cong \mathbf{Q}[X] \big/ (P)$$

- Ring of integers:

$$\mathcal{O}_L = \{\alpha \mid \exists R \in \mathbf{Z}[X] \text{ monic }, R(\alpha) = 0\}$$

For instance:

$$\mathcal{O}_{\mathbf{Q}} = \mathbf{Z}$$

$$\mathcal{O}_{\mathbf{Q}(i)} = \mathbf{Z}[i] = \{a + ib \mid a, b \in \mathbf{Z}\}$$

$$\mathcal{O}_{\mathbf{Q}(j)} = \mathbf{Z}[i] = \{a + jb \mid a, b \in \mathbf{Z}\}$$

Lattice

An (algebraic) lattice Λ is a free $\mathcal{O}_L$-module of finite rank, endowed with an inner product on $\Lambda \otimes_{\mathcal{O}_L} \mathbf{R}$.

Natural Hermitian structure

- $[L:\mathbf{Q}]$ embeddings $L \to \mathbf{C}$
- Archimedean embedding Σ:

$$\begin{array}{ccc} L \otimes_{\mathbf{Q}} \mathbf{R} & \to & \mathbf{R}^r \times \mathbf{C}^c \\ x & \mapsto & (\sigma(x))_{\sigma : L \to \mathbf{C}} \end{array}$$

- *Transport* the Hermitian structure to $L \otimes \mathbf{R}$:

$$\langle a, b \rangle_\Sigma = \sum_{\sigma : L \to \mathbf{C}} \overline{\sigma(a)}\sigma(b).$$

- For any $x = (x_1, \ldots, x_d) \in (L \otimes \mathbf{R})^d$ and $y = (y_1, \ldots, y_d) \in (L \otimes \mathbf{R})^d$:

$$\langle x, y \rangle = \sum_{i=1}^{d} \langle x_i, y_i \rangle_\Sigma,$$

Lattice

An (algebraic) lattice Λ is a free $\mathcal{O}_L$-module of finite rank, endowed with an inner product on $\Lambda \otimes_{\mathcal{O}_L} \mathbf{R}$.

Natural Hermitian structure

- $[L:\mathbf{Q}]$ embeddings $L \to \mathbf{C}$
- Archimedean embedding Σ:

$$\begin{array}{ccc} L \otimes_{\mathbf{Q}} \mathbf{R} & \to & \mathbf{R}^r \times \mathbf{C}^c \\ x & \mapsto & (\sigma(x))_{\sigma : L \to \mathbf{C}} \end{array}$$

- *Transport* the Hermitian structure to $L \otimes \mathbf{R}$:

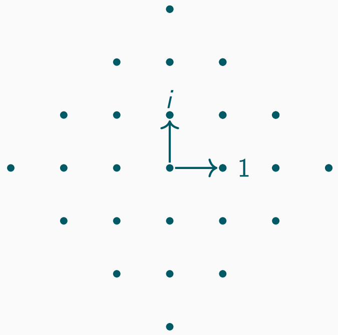
$$\langle a, b \rangle_\Sigma = \sum_{\sigma : L \to \mathbf{C}} \overline{\sigma(a)}\sigma(b).$$

- For any $x = (x_1, \ldots, x_d) \in (L \otimes \mathbf{R})^d$ and $y = (y_1, \ldots, y_d) \in (L \otimes \mathbf{R})^d$:

$$\langle x, y \rangle = \sum_{i=1}^{d} \langle x_i, y_i \rangle_\Sigma,$$

Lattice

An (algebraic) lattice Λ is a free $\mathcal{O}_L$-module of finite rank, endowed with an inner product on $\Lambda \otimes_{\mathcal{O}_L} \mathbf{R}$.



$\mathbf{Z}[i]$

Natural Hermitian structure

- $[L:\mathbf{Q}]$ embeddings $L \to \mathbf{C}$
- Archimedean embedding Σ:

$$\begin{array}{ccc} L \otimes_{\mathbf{Q}} \mathbf{R} & \to & \mathbf{R}^r \times \mathbf{C}^c \\ x & \mapsto & (\sigma(x))_{\sigma: L \to \mathbf{C}} \end{array}$$

- *Transport* the Hermitian structure to $L \otimes \mathbf{R}$:

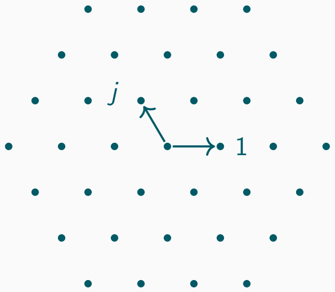
$$\langle a, b \rangle_\Sigma = \sum_{\sigma: L \to \mathbf{C}} \overline{\sigma(a)} \sigma(b).$$

- For any $x = (x_1, \ldots, x_d) \in (L \otimes \mathbf{R})^d$ and $y = (y_1, \ldots, y_d) \in (L \otimes \mathbf{R})^d$:

$$\langle x, y \rangle = \sum_{i=1}^{d} \langle x_i, y_i \rangle_\Sigma,$$

Lattice

An (algebraic) lattice Λ is a free $\mathcal{O}_L$-module of finite rank, endowed with an inner product on $\Lambda \otimes_{\mathcal{O}_L} \mathbf{R}$.



$\mathbf{Z}[j]$

Reduction of algebraic lattices

Techniques for the reduction of algebraic lattices

Techniques for the reduction of algebraic lattices

Module-SVP to Module-SVP type

QR-decomposition

Need an oracle...

Techniques for the reduction of algebraic lattices

Rank recursivity

Controlled precision

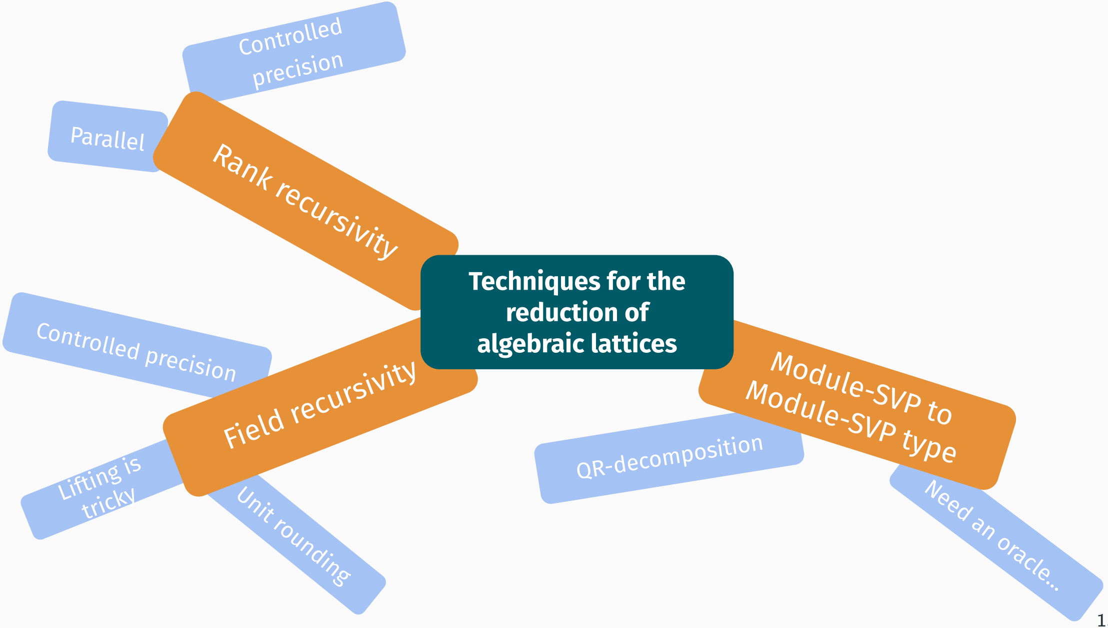
Parallel

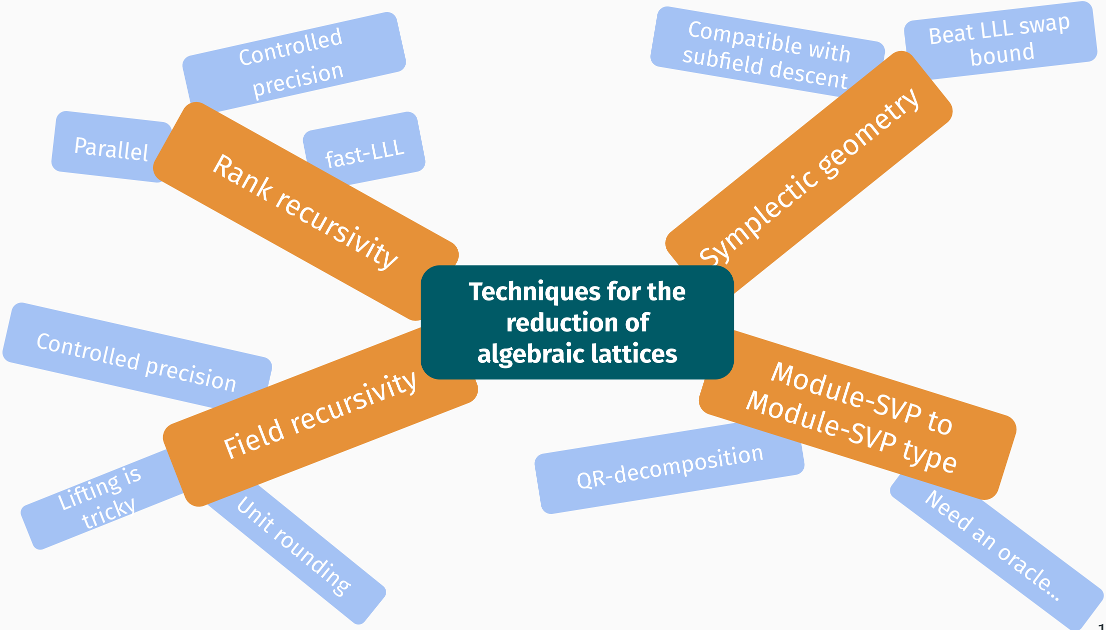
fast-LLL

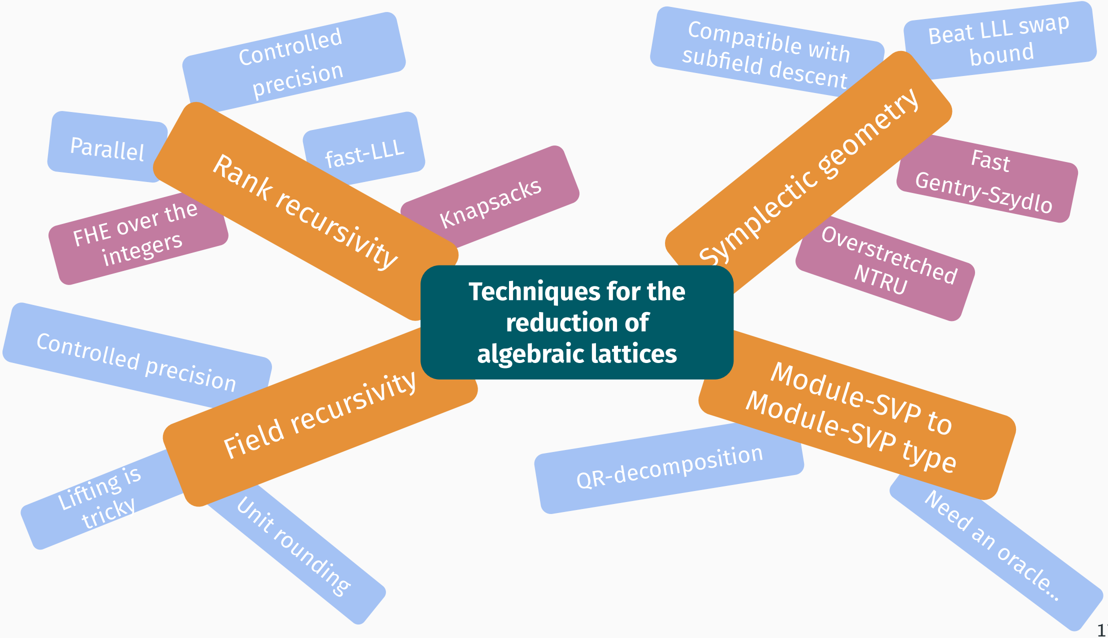
Module-SVP to Module-SVP type

QR-decomposition

Need an oracle...


Techniques for the reduction of algebraic lattices
Rank recursivity
Controlled precision
Parallel
Field recursivity
Controlled precision
Lifting is tricky
Unit rounding
Module-SVP to Module-SVP type
QR-decomposition
Need an oracle...


Techniques for the reduction of algebraic lattices
Rank recursivity
Controlled precision
Parallel
fast-LLL
Symplectic geometry
Compatible with subfield descent
Beat LLL swap bound
Field recursivity
Controlled precision
Lifting is tricky
Unit rounding
Module-SVP to Module-SVP type
QR-decomposition
Need an oracle...


Techniques for the reduction of algebraic lattices
Rank recursivity
Controlled precision
Parallel
fast-LLL
Knapsacks
FHE over the integers
Field recursivity
Controlled precision
Lifting is tricky
Unit rounding
Symplectic geometry
Compatible with subfield descent
Beat LLL swap bound
Fast Gentry-Szydlo
Overstretched NTRU
Module-SVP to Module-SVP type
QR-decomposition
Need an oracle...

Base reduction

Enhancing reduction over Z

$\rightarrow$ **Idea**: Use parallelisation and recursion on the rank.

Enhancing reduction over Z

Orthogonalize (M = QR)

for $j = 1$ **to** d **do**
  $Q_j \leftarrow M_j - \sum_{i=1}^{j-1} \frac{\langle M_j, Q_i \rangle}{\langle Q_i, Q_i \rangle} Q_i$
end for
return $R = \left(\frac{\langle Q_i, M_j \rangle}{\| Q_i \|} \right)_{1 \leqslant i < j \leqslant d}$;

$\rightarrow$ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

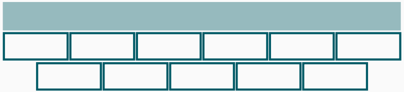
- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices

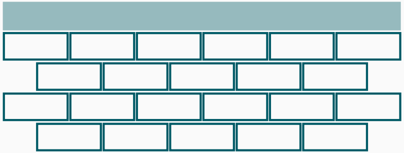
Enhancing reduction over Z



→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices

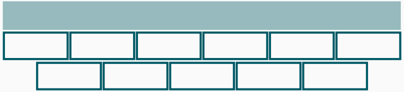
Enhancing reduction over Z



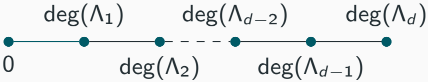
→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of QR decomposition
- Reduction is done by reducing rank 2 projected sublattices

Enhancing reduction over Z



A round of local reductions acts as a *discretized* Laplacian operator on the profile "space":



- Reminiscent of the *diffusion property* of the solution of the heat equation

$$\frac{\partial u}{\partial t} = \alpha \Delta u$$

- *Characteristic time* is **quadratic in the diameter** of the space.

$\rightarrow$ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of QR decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

$\rightarrow$ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of QR decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

$\rightarrow$ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of *QR* decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

Enhancing reduction over Z

→ **Idea**: Use parallelisation and recursion on the rank.

- Work on filtrations/R-part of QR decomposition
- Reduction is done by reducing rank 2 projected sublattices
- Operations are *local*: possible to recurse on blocks

On the complexity of the reduction

Complexity [E-Kirchner-Fouque 2019]

Let A be a matrix of dimension d with entries in $\mathbf{Z}$, with $\kappa(A) \leq 2^B$ such that $B \geq d$. Our reduction algorithm finds an integer vector x with

$$\|Ax\| \leq 2^{d/2} |\det A|^{1/d}.$$

Further, the *heuristic* running time is

$$\mathrm{O}\left(\frac{d^\omega}{(\omega-2)^2} \cdot \frac{B}{\log B} + d^2 B \log B\right).$$

Playing with number fields

$$\begin{tikzcd}
\mathsf{L} & \mathcal{O}_{\mathsf{L}} \arrow[l, hook'] \\
\mathsf{K} \arrow[u, hook] & \mathcal{O}_{\mathsf{K}} \arrow[l, hook'] \arrow[u, hook] \\
\mathsf{Q} \arrow[u, hook] & \mathsf{Z} \arrow[l, hook'] \arrow[u, hook]
\end{tikzcd}$$

Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of QR decomposition

Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of *QR* decomposition ✓

Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of *QR* decomposition ✓
- Reduction can be done with the parallel structure

Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of QR decomposition $\checkmark$
- Reduction can be done with the parallel structure $\checkmark$

Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of QR decomposition $\checkmark$
- Reduction can be done with the parallel structure $\checkmark$

Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of QR decomposition ✓
- Reduction can be done with the parallel structure ✓
- Size-reduction?

Adapting to number fields

→ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of QR decomposition ✓
- Reduction can be done with the parallel structure ✓
- Size-reduction?

- Over $\mathbf{Z}$: requires integral rounding

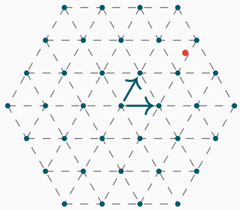
Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of *QR* decomposition ✓
- Reduction can be done with the parallel structure ✓
- Size-reduction?

- Over $\mathbf{Z}$: requires integral rounding

- Translated over $\mathcal{O}_K$: find the closest element in this ring: instance of CVP



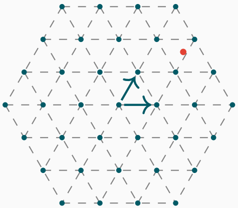
Adapting to number fields

→ **Idea**: Use the same structure as over **Z**, but exploit the algebraic specificities.

- Work on filtrations/R-part of *QR* decomposition ✓
- Reduction can be done with the parallel structure ✓
- Size-reduction?

- Over **Z**: requires integral rounding

- Translated over $\mathcal{O}_K$: find the closest element in this ring: instance of CVP



- Approx-CVP suffices

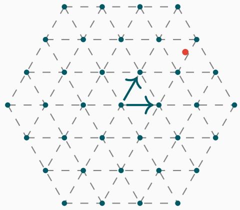
Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of *QR* decomposition ✓
- Reduction can be done with the parallel structure ✓
- Size-reduction?

- Over $\mathbf{Z}$: requires integral rounding

- Translated over $\mathcal{O}_K$: find the closest element in this ring: instance of CVP



- Approx-CVP suffices : just do the coefficient-wise rounding!

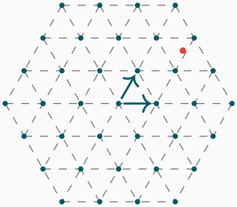
Adapting to number fields

$\rightarrow$ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of *QR* decomposition ✓
- Reduction can be done with the parallel structure ✓
- Size-reduction? ✓

- Over $\mathbf{Z}$: requires integral rounding

- Translated over $\mathcal{O}_K$: find the closest element in this ring: instance of CVP



- Approx-CVP suffices : just do the coefficient-wise rounding!

Adapting to number fields

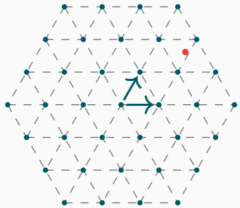
→ **Idea**: Use the same structure as over $\mathbf{Z}$, but exploit the algebraic specificities.

- Work on filtrations/R-part of *QR* decomposition
- Reduction can be done with the parallel structure
- Size-reduction?

  Hack: Use **units** to decrease the condition number and *lower the precision*.

- Over $\mathbf{Z}$: requires integral rounding

- Translated over $\mathcal{O}_K$: find the closest element in this ring: instance of CVP



- Approx-CVP suffices : just do the coefficient-wise rounding!

General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] \\
\vdots \arrow[u, hook] \\
\mathbf{Z} \arrow[u, hook]
\end{tikzcd}
$$

General recursive strategy

$$\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] \\
\vdots \arrow[u, hook] \\
\mathbf{Z} \arrow[u, hook]
\end{tikzcd}$$

General recursive strategy

$$\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dashed, "\text{reduce rk } 2"] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & & \\
\vdots \arrow[u, hook] & & \\
\mathbf{Z} \arrow[u, hook] & &
\end{tikzcd}$$

General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dashed, "\text{reduce rk 2}"] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[dl, dashed, bend right, "\text{descend}"'] \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} & \\
\vdots \arrow[u, hook] & & \\
\mathbf{Z} \arrow[u, hook] & &
\end{tikzcd}
$$

General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dashed, "\text{reduce rk } 2"] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[dl, dashed, bend right, "\text{descend}"'] \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} \arrow[r, dashed, no head] & \alpha'_1\mathcal{O}_{K_{h-1}} \oplus \alpha'_2\mathcal{O}_{K_{h-1}} \\
\vdots \arrow[u, hook] & & \\
\mathbf{Z} \arrow[u, hook] & &
\end{tikzcd}
$$

General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[rr, dashed, hook, "\text{reduce rk } 2"] & & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[dll, dashed, bend right, two heads, "\text{descend}"'] \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} \arrow[rr, dashed, dash] & & \alpha'_1\mathcal{O}_{K_{h-1}} \oplus \alpha'_2\mathcal{O}_{K_{h-1}} \arrow[dll, dotted, hook, two heads] \\
\vdots \arrow[u, hook] & \vdots \arrow[rr, dotted, dash] & & \vdots \arrow[dll, dotted, bend right, two heads, "\text{descend}"'] \\
\mathbf{Z} \arrow[u, hook] & M_{\mathbf{Z}} = \beta_1\mathbf{Z} \oplus \cdots \oplus \beta_k\mathbf{Z} & &
\end{tikzcd}
$$

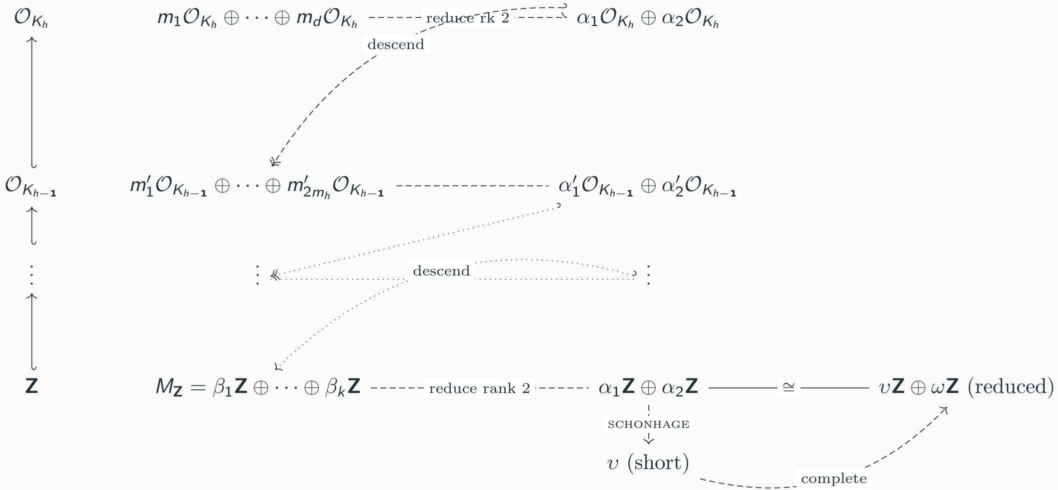
General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dash, dashed, "\text{reduce rk } 2" description] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[dl, dashed, bend right, "\text{descend}" description] \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} \arrow[r, dash, dashed] & \alpha'_1\mathcal{O}_{K_{h-1}} \oplus \alpha'_2\mathcal{O}_{K_{h-1}} \arrow[dl, dotted] \\
\vdots \arrow[u, hook] & \vdots \arrow[r, dash, dotted] & \vdots \arrow[dl, dotted, bend right, "\text{descend}" description] \\
\mathbf{Z} \arrow[u, hook] & M_{\mathbf{Z}} = \beta_1\mathbf{Z} \oplus \cdots \oplus \beta_k\mathbf{Z} \arrow[r, dash, dashed, "\text{reduce rank } 2" description] & \alpha_1\mathbf{Z} \oplus \alpha_2\mathbf{Z}
\end{tikzcd}
$$

General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dashed, "\text{reduce rk } 2"] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[dl, dashed, bend right, "\text{descend}"'] \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} \arrow[r, dashed] & \alpha'_1\mathcal{O}_{K_{h-1}} \oplus \alpha'_2\mathcal{O}_{K_{h-1}} \arrow[dl, dotted, "\text{descend}"] \\
\vdots \arrow[u, hook] & \vdots \arrow[r, dotted] & \vdots \arrow[dl, dotted, bend right] \\
\mathbf{Z} \arrow[u, hook] & M_{\mathbf{Z}} = \beta_1\mathbf{Z} \oplus \cdots \oplus \beta_k\mathbf{Z} \arrow[r, dashed, "\text{reduce rank } 2"] & \alpha_1\mathbf{Z} \oplus \alpha_2\mathbf{Z} \arrow[d, dashed, "\text{SCHONHAGE}"] \\
 & & v \text{ (short)}
\end{tikzcd}
$$

General recursive strategy



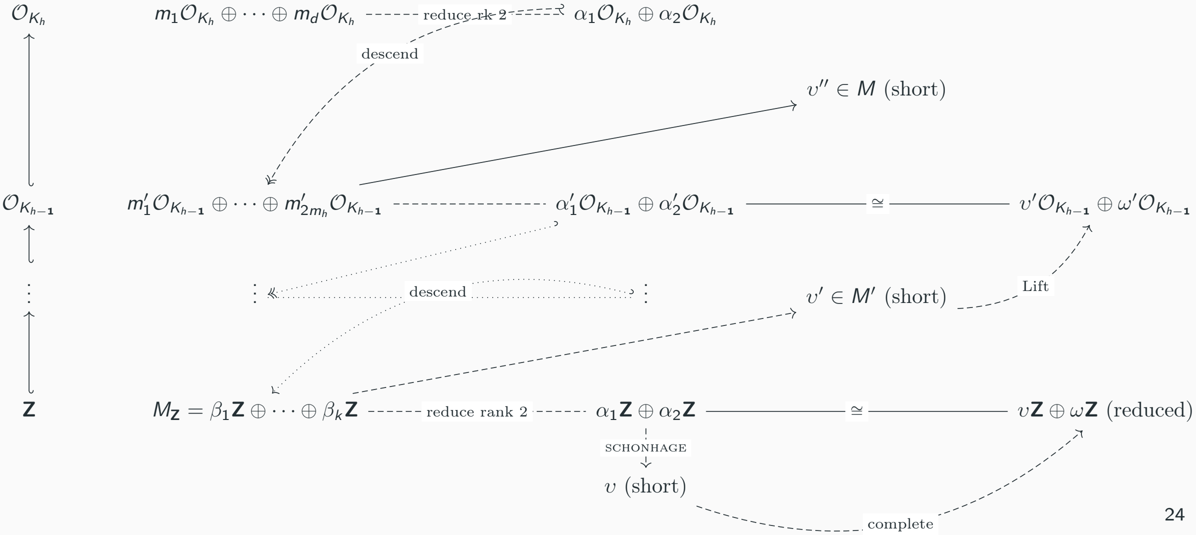
General recursive strategy

$$\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dashed, "\text{reduce rk } 2"] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[dl, dashed, bend right, "\text{descend}"'] & & \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} \arrow[r, dash, dashed] & \alpha'_1\mathcal{O}_{K_{h-1}} \oplus \alpha'_2\mathcal{O}_{K_{h-1}} \arrow[dl, dotted] & & \\
\vdots \arrow[u, hook] & \vdots & \vdots \arrow[l, dotted, "\text{descend}"] \arrow[dl, dotted] & & \upsilon' \in M' \ \text{(short)} \\
\mathbf{Z} \arrow[u, hook] & M_{\mathbf{Z}} = \beta_1\mathbf{Z} \oplus \cdots \oplus \beta_k\mathbf{Z} \arrow[r, dashed, "\text{reduce rank } 2"] \arrow[urrr, dashed] & \alpha_1\mathbf{Z} \oplus \alpha_2\mathbf{Z} \arrow[rr, dash, "\cong" description] \arrow[d, dashed, "\textsc{Schonhage}" description] & & \upsilon\mathbf{Z} \oplus \omega\mathbf{Z} \ \text{(reduced)} \\
& & \upsilon \ \text{(short)} \arrow[urr, dashed, bend right, "\text{complete}"'] & &
\end{tikzcd}$$

General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dashed, "\text{reduce rk 2}"] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[dl, dashed, bend right, "\text{descend}"'] & \\
\mathcal{O}_{K_{h-1}} \arrow[u, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} \arrow[r, dash, dashed] & \alpha'_1\mathcal{O}_{K_{h-1}} \oplus \alpha'_2\mathcal{O}_{K_{h-1}} \arrow[r, dash, "\cong"] \arrow[dl, dotted] & \upsilon'\mathcal{O}_{K_{h-1}} \oplus \omega'\mathcal{O}_{K_{h-1}} \\
\vdots \arrow[u, hook] & \vdots & \vdots \arrow[l, dotted, "\text{descend}"'] \arrow[dl, dotted, bend right] & \upsilon' \in M' \ (\text{short}) \arrow[u, dashed, bend right, "\text{Lift}"'] \\
\mathbf{Z} \arrow[u, hook] & M_{\mathbf{Z}} = \beta_1\mathbf{Z} \oplus \cdots \oplus \beta_k\mathbf{Z} \arrow[r, dashed, "\text{reduce rank 2}"] \arrow[urr, dashed] & \alpha_1\mathbf{Z} \oplus \alpha_2\mathbf{Z} \arrow[r, dash, "\cong"] \arrow[d, dashed, "\textsc{Schonhage}"] & \upsilon\mathbf{Z} \oplus \omega\mathbf{Z} \ (\text{reduced}) \\
 & & \upsilon \ (\text{short}) \arrow[ur, dashed, bend right, "\text{complete}"'] &
\end{tikzcd}
$$

General recursive strategy



General recursive strategy

$$
\begin{tikzcd}
\mathcal{O}_{K_h} & m_1\mathcal{O}_{K_h} \oplus \cdots \oplus m_d\mathcal{O}_{K_h} \arrow[r, dashed, "\text{reduce rk 2}"] \arrow[dd, dashed, bend right, "\text{descend}"'] & \alpha_1\mathcal{O}_{K_h} \oplus \alpha_2\mathcal{O}_{K_h} \arrow[rr, no head, "\cong"] & & \upsilon''\mathcal{O}_{K_h} \oplus \omega''\mathcal{O}_{K_h} \\
 & & & \upsilon'' \in M \ (\text{short}) \arrow[ur, dashed, bend right, "\text{Lift}"'] & \\
\mathcal{O}_{K_{h-1}} \arrow[uu, hook] & m'_1\mathcal{O}_{K_{h-1}} \oplus \cdots \oplus m'_{2m_h}\mathcal{O}_{K_{h-1}} \arrow[r, dashed, no head] \arrow[urr] & \alpha'_1\mathcal{O}_{K_{h-1}} \oplus \alpha'_2\mathcal{O}_{K_{h-1}} \arrow[rr, no head, "\cong"] \arrow[dl, dotted, "\text{descend}"] & & \upsilon'\mathcal{O}_{K_{h-1}} \oplus \omega'\mathcal{O}_{K_{h-1}} \\
\vdots \arrow[u, hook] & \vdots \arrow[d, dotted] & \vdots \arrow[l, dotted] & \upsilon' \in M' \ (\text{short}) \arrow[ur, dashed, bend right, "\text{Lift}"'] & \\
\mathbf{Z} \arrow[u, hook] & M_{\mathbf{Z}} = \beta_1\mathbf{Z} \oplus \cdots \oplus \beta_k\mathbf{Z} \arrow[r, dashed, "\text{reduce rank 2}"] \arrow[urr, dashed] & \alpha_1\mathbf{Z} \oplus \alpha_2\mathbf{Z} \arrow[rr, no head, "\cong"] \arrow[d, dashed, "\text{SCHONHAGE}"] & & \upsilon\mathbf{Z} \oplus \omega\mathbf{Z} \ (\text{reduced}) \\
 & & \upsilon \ (\text{short}) \arrow[urr, dashed, bend right, "\text{complete}"'] & &
\end{tikzcd}
$$

On the complexity of the reduction

Complexity [E-Kirchner-Fouque 2019]

Let f be a log-smooth integer. The complexity of the algorithm **Reduce** on rank two modules over $K = \mathbf{Q}[x]/\Phi_f(x)$, represented as a matrix M whose number of bits in the input coefficients is uniformly bounded by $B > n$, is *heuristically* a $\tilde{\mathrm{O}}(n^2 B)$ with $n = \varphi(f)$. The first column of the reduced matrix has its coefficients uniformly bounded by $2^{\tilde{\mathrm{O}}(n)} \operatorname{covol}(M)^{\frac{1}{2n}}$.

Faster with symplectic symmetries

$$\begin{array}{c}\mathrm{Sp}_{\omega}(2, \mathsf{L} \otimes \mathsf{R}) \\ \big\downarrow \\ \mathrm{Sp}_{\omega \cdot \tau}(2[\mathsf{L} : \mathsf{K}], \mathsf{K} \otimes \mathsf{R})\end{array}$$

A primer on symplectic geometry

A primer on symplectic geometry

Euclidean space	Symplectic space

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$

Symplectic space

- Antisymmetric bilinear Form ω

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

□□□□□□□□□□□□

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$

□□□□□□□□□□□□

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$

A primer on symplectic geometry

Euclidean space

- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$

A primer on symplectic geometry

Euclidean space

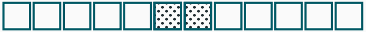
- Symmetric bilinear Form $\langle \cdot, \cdot \rangle$
- Transformation group: $O_n(\mathbf{R})$
- *Nice* bases: Orthonormal bases

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Symplectic space

- Antisymmetric bilinear Form ω
- Transformation group: $\mathrm{Sp}_\omega(\mathbf{R})$
- *Nice* bases: Darboux bases

$$\begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}$$



Going further with symplectic symmetries

$$\begin{tikzcd}
\mathbf{K}_h & \mathcal{O}_{\mathbf{K}_h} \arrow[l, hook'] \\
\mathbf{K}_{h-1} \arrow[u, hook, "r"] & \mathcal{O}_{\mathbf{K}_{h-1}} \arrow[l, hook'] \arrow[u, hook] \\
\vdots \arrow[u, hook] & \vdots \arrow[u, hook] \\
\mathbf{Q} \arrow[u, hook] & \mathbf{Z} \arrow[l, hook'] \arrow[u, hook]
\end{tikzcd}$$

Going further with symplectic symmetries

J_h

$$\begin{tikzcd}
\mathbf{K}_h & \mathcal{O}_{\mathbf{K}_h} \arrow[l, hook'] \\
\mathbf{K}_{h-1} \arrow[u, hook, "r"] & \mathcal{O}_{\mathbf{K}_{h-1}} \arrow[l, hook'] \arrow[u, hook] \\
\vdots \arrow[u, hook] & \vdots \arrow[u, hook] \\
\mathbf{Q} \arrow[u, hook] & \mathbf{Z} \arrow[l, hook'] \arrow[u, hook]
\end{tikzcd}$$

$J_h \in \bigwedge^2(K_h^2)$ is the determinant form:

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0$$

M is J_h-symplectic iff $\det M = 1$.

Going further with symplectic symmetries

$$
\begin{tikzcd}
J_h \arrow[d, dotted, "\tau"] & \mathbf{K}_h & \mathcal{O}_{\mathbf{K}_h} \arrow[l, hook'] \\
J_h^{(1)} & \mathbf{K}_{h-1} \arrow[u, hook, "r"] & \mathcal{O}_{\mathbf{K}_{h-1}} \arrow[l, hook'] \arrow[u, hook] \\
 & \vdots \arrow[u, hook] & \vdots \arrow[u, hook] \\
 & \mathbf{Q} \arrow[u, hook] & \mathbf{Z} \arrow[l, hook'] \arrow[u, hook]
\end{tikzcd}
$$

$J_h \in \bigwedge^2(K_h^2)$ is the determinant form:

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0$$

$\rightarrow$ Descend the form J_h in $J_h^{(1)}$ to K_{h-1} by composition with a non-trivial linear form $\tau : K_h \rightarrow K_{h-1}$

Going further with symplectic symmetries

$$\begin{tikzcd}
J_h \arrow[d, dotted, "\tau"] & \mathbf{K}_h & \mathcal{O}_{\mathbf{K}_h} \arrow[l, hook'] \\
J_h^{(1)} \arrow[d, dotted] & \mathbf{K}_{h-1} \arrow[u, hook, "r"] & \mathcal{O}_{\mathbf{K}_{h-1}} \arrow[l, hook'] \arrow[u, hook] \\
{} & \vdots \arrow[u, hook] & \vdots \arrow[u, hook] \\
 & \mathbf{Q} \arrow[u, hook] & \mathbf{Z} \arrow[l, hook'] \arrow[u, hook]
\end{tikzcd}$$

$J_h \in \bigwedge^2(K_h^2)$ is the determinant form:

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0$$

$\rightarrow$ Descend the form J_h in $J_h^{(1)}$ to K_{h-1} by composition with a non-trivial linear form $\tau : K_h \rightarrow K_{h-1}$

Going further with symplectic symmetries

$J_h \in \bigwedge^2(K_h^2)$ is the determinant form:

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0$$

$\rightarrow$ Descend the form J_h in $J_h^{(1)}$ to K_{h-1} by composition with a non-trivial linear form $\tau : K_h \rightarrow K_{h-1}$

$$\begin{tikzcd}
J_h \arrow[d, dotted, "\tau"] & \mathbf{K}_h & \mathcal{O}_{\mathbf{K}_h} \arrow[l, hook'] \\
J_h^{(1)} \arrow[d, dotted] & \mathbf{K}_{h-1} \arrow[u, hook, "r"] & \mathcal{O}_{\mathbf{K}_{h-1}} \arrow[l, hook'] \arrow[u, hook] \\
\vdots \arrow[d, dotted] & \vdots \arrow[u, hook] & \vdots \arrow[u, hook] \\
J_h^h & \mathbf{Q} \arrow[u, hook] & \mathbf{Z} \arrow[l, hook'] \arrow[u, hook]
\end{tikzcd}$$

Going further with symplectic symmetries

$J_h \in \bigwedge^2(K_h^2)$ is the determinant form:

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0$$

$\rightarrow$ Descend the form J_h in $J_h^{(1)}$ to K_{h-1} by composition with a non-trivial linear form $\tau : K_h \rightarrow K_{h-1}$

$$\begin{tikzcd}
\mathrm{Sp}_{J_h}(2, \mathbf{K}_h) \arrow[d, hook] & J_h \arrow[d, dotted, "\tau"] & \mathbf{K}_h & \mathcal{O}_{\mathbf{K}_h} \arrow[l, hook'] \\
\mathrm{Sp}_{J_h^{(1)}}(2r, \mathbf{K}_{h-1}) & J_h^{(1)} \arrow[d, dotted] & \mathbf{K}_{h-1} \arrow[u, hook, "r"] & \mathcal{O}_{\mathbf{K}_{h-1}} \arrow[l, hook'] \arrow[u, hook] \\
 & \vdots \arrow[d, dotted] & \vdots \arrow[u, hook] & \vdots \arrow[u, hook] \\
 & J_h^h & \mathbf{Q} \arrow[u, hook] & \mathbf{Z} \arrow[l, hook'] \arrow[u, hook]
\end{tikzcd}$$

Compatibility

Let M be a 2×2 matrix over K_h which is J_h-symplectic, then its descent $M' \in K_{h-1}^{2d_h \times 2d_h}$ is J_h'-symplectic.

Going further with symplectic symmetries

$J_h \in \bigwedge^2(K_h^2)$ is the determinant form:

$$J_h\left(\begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}\right) = x_0 y_1 - x_1 y_0$$

$\rightarrow$ Descend the form J_h in $J_h^{(1)}$ to K_{h-1} by composition with a non-trivial linear form $\tau : K_h \rightarrow K_{h-1}$

$$\begin{tikzcd}
\mathrm{Sp}_{J_h}(2, \mathbf{K}_h) \arrow[d, hook] & J_h \arrow[d, dashed, "\tau"] & \mathbf{K}_h & \mathcal{O}_{\mathbf{K}_h} \arrow[l, hook'] \\
\mathrm{Sp}_{J_h^{(1)}}(2r, \mathbf{K}_{h-1}) \arrow[d, hook] & J_h^{(1)} \arrow[d, dashed] & \mathbf{K}_{h-1} \arrow[u, "r"] & \mathcal{O}_{\mathbf{K}_{h-1}} \arrow[l, hook'] \arrow[u] \\
\vdots \arrow[d, hook] & \vdots \arrow[d, dashed] & \vdots \arrow[u] & \vdots \arrow[u] \\
\mathrm{Sp}_{J_h^{(h)}}(2n, \mathbf{Q}) & J_h^h & \mathbf{Q} \arrow[u] & \mathbf{Z} \arrow[l, hook'] \arrow[u]
\end{tikzcd}$$

Compatibility

Let M be a 2×2 matrix over K_h which is J_h-symplectic, then its descent $M' \in K_{h-1}^{2d_h \times 2d_h}$ is J_h'-symplectic.

Improved complexity

Improved complexity [E-Kirchner-Fouque 2019]

Select an integer f a power of $q = \mathrm{O}(\log f)$ and let $n = \varphi(f)$. The complexity for reducing matrices M with condition number lower than 2^B, of dimension two over $L = \mathbf{Q}[x]/\Phi_f(x)$ with B the number of bits in the input coefficients is *heuristically*

$$\tilde{\mathrm{O}}\left(n^{2+\varepsilon(q)}B\right) + n^{\mathrm{O}(\log\log n)}, \qquad \varepsilon(q) = \frac{\log(1/2 + 1/2q)}{\log q} < 0$$

and the first column of the reduced matrix has coefficients bounded by

$$2^{\tilde{\mathrm{O}}(n)}\left|N_{K_h/\mathbf{Q}}(\det M)\right|^{\frac{1}{2n}}.$$

Open problems

- Getting further using higher-order symplectic structures

Exploitation of the symplectic symmetries:

1. **Decrease** the complexity, but...
2. **Increase** the approximation factor

Open problems

- Getting further using higher-order symplectic structures

Exploitation of the symplectic symmetries:

1. **Decrease** the complexity, but...
2. **Increase** the approximation factor

Seek for transformations preserving arbitrary non-degenerate alternate forms (for example the *volume form*)

Open problems

- Getting further using higher-order symplectic structures

Exploitation of the symplectic symmetries:

1. **Decrease** the complexity, but...
2. **Increase** the approximation factor

Seek for transformations preserving arbitrary non-degenerate alternate forms (for example the *volume form*)

Problems:

- Non uniqueness of higher-order symplectic structures (no Darboux' structure theorem)
- Find a descent compatible with this additional structure

Open problems

- Getting further using higher-order symplectic structures

Exploitation of the symplectic symmetries:

1. **Decrease** the complexity, but...
2. **Increase** the approximation factor

Seek for transformations preserving arbitrary non-degenerate alternate forms (for example the *volume form*)

Problems:

- Non uniqueness of higher-order symplectic structures (no Darboux' structure theorem)
- Find a descent compatible with this additional structure

Open problems

- Getting further using higher-order symplectic structures
- Get rid of the heuristics: reduce *projective* modules

Open problems

- Getting further using higher-order symplectic structures

- Get rid of the heuristics: reduce *projective* modules

Over $\mathcal{O}_L$ a projective module is of the shape: $\alpha_1\mathfrak{a}_1 \oplus \cdots \oplus \alpha_n\mathfrak{a}_n$

- Need to adapt the lifting to ideals [Cohen]

Open problems

- Getting further using higher-order symplectic structures
- Get rid of the heuristics: reduce *projective* modules

Over $\mathcal{O}_L$ a projective module is of the shape: $\alpha_1\mathfrak{a}_1 \oplus \cdots \oplus \alpha_n\mathfrak{a}_n$

- Need to adapt the lifting to ideals [Cohen]
- Requires computations with ideals: bottleneck is now the ideal multiplication algorithm

Open problems

- Getting further using higher-order symplectic structures
- Get rid of the heuristics: reduce *projective* modules

Over $\mathcal{O}_L$ a projective module is of the shape: $\alpha_1 \mathfrak{a}_1 \oplus \cdots \oplus \alpha_n \mathfrak{a}_n$

- Need to adapt the lifting to ideals [Cohen]
- Requires computations with ideals: bottleneck is now the ideal multiplication algorithm
- 2-elements representation: multiplying $\mathfrak{a} = \alpha_1 \mathcal{O}_L + \alpha_2 \mathcal{O}_L, \mathfrak{b} = \beta_1 \mathcal{O}_L + \beta_2 \mathcal{O}_L$ consists in the reduction of the ideal generated by $(\alpha_i \beta_j)_{1 \leqslant i,j \leqslant 2}$ (module spanned by 4 elements)

Open problems

- Getting further using higher-order symplectic structures

- Get rid of the heuristics: reduce *projective* modules

Over $\mathcal{O}_L$ a projective module is of the shape: $\alpha_1\mathfrak{a}_1 \oplus \cdots \oplus \alpha_n\mathfrak{a}_n$

- Need to adapt the lifting to ideals [Cohen]
- Requires computations with ideals: bottleneck is now the ideal multiplication algorithm

Cross recursive algorithms: reduction and ideal multiplication

Thank you !