Lattices and Cryptographic Applications

Prof. dr. ir. Frederik Vercauteren

COSIC, KU Leuven

Computer Algebra for Cryptography (B-KUL-H0E74A)

2022-2023

Definition and first facts

Lattice Problems

Lattice Reduction Algorithms

Lattices: a fantastic tool for

- Cryptanalysis (see Lecture 5):
  - Merkle–Hellman knapsack cryptosystem, RSA with small decryption exponent, ...
- Building cryptosystems (see Lecture 6)
  - post-quantum cryptography (3 out of 4 NIST standards!), fully homomorphic encryption, ...
- Computer algebra in general:
  - factoring polynomials, computing minimal polynomials, finding small roots of polynomials (Coppersmith's algorithm), ...
- Number theory (Minkowski's geometry of numbers °1910):
  - Lagrange's four-square theorem, finiteness of ideal class groups, Diophantine approximation, ...
- Sphere packings, ...

First definition of a lattice

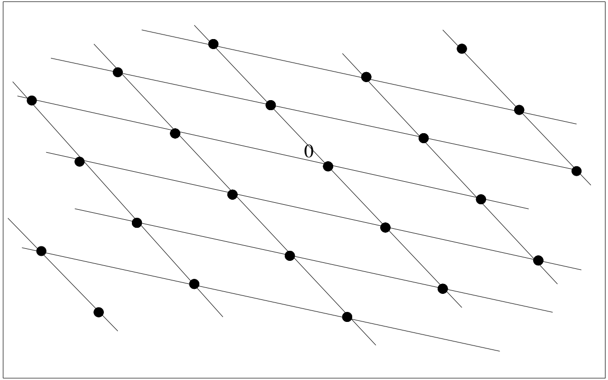
A lattice L is a **discrete subgroup** of $\mathbb{R}^n$:

1. $\mathbf{b} \in L \Rightarrow -\mathbf{b} \in L$
2. $\mathbf{b}_1, \mathbf{b}_2 \in L \Rightarrow \mathbf{b}_1 + \mathbf{b}_2 \in L$
3. Discrete: each point is contained in small open ball containing no other point

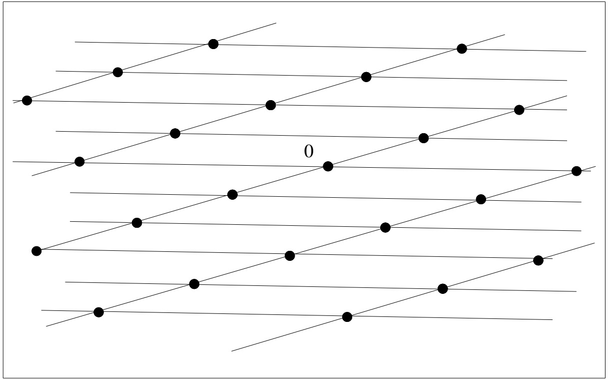
Examples:

- Integer lattice $\mathbb{Z}^d \subset \mathbb{R}^n$ with $d \leq n$
- Any subgroup of $\mathbb{Z}^d \subset \mathbb{R}^n$ with $d \leq n$
- Counterexample: the two (one dimensional) vectors $\{1, \sqrt{2}\}$ do not span lattice
  - Fun fact: can use lattices to show that this is not a lattice (see later)!

A 2-dimensional lattice



The same 2-dimensional lattice



Second definition of a lattice

A set $L \subset \mathbb{R}^n$ is a lattice if and only if there exist $\mathbb{R}$-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{R}^n$ such that

$$L = L(\mathbf{b}_1, \ldots, \mathbf{b}_d) := \left\{ \sum_{i=1}^{d} x_i \mathbf{b}_i \;\middle|\; x_i \in \mathbb{Z} \right\}$$

- Lattice dimension: $d = \dim(L)$
- Embedding dimension: n
- $\mathbf{b}_1, \ldots, \mathbf{b}_d$ is a lattice basis (not unique)

Example use of first definition

Let $A = (a_{i,j})_{i,j} \in \mathbb{Z}^{m \times n}$ and consider the **homogeneous** system of equations:

$$\begin{cases} a_{1,1}x_1 & + & a_{1,2}x_2 & + & \dots & + & a_{1,n}x_n & = & 0 \\ a_{2,1}x_1 & + & a_{2,2}x_2 & + & \dots & + & a_{2,n}x_n & = & 0 \\ \vdots & & \vdots & & \dots & & \vdots & & \vdots \\ a_{m,1}x_1 & + & a_{m,2}x_2 & + & \dots & + & a_{m,n}x_n & = & 0 \end{cases}.$$

- Set of solutions $(x_1, \dots, x_n)$ is a lattice L
- If the rows of A are linearly independent, $\dim(L) = n - m$

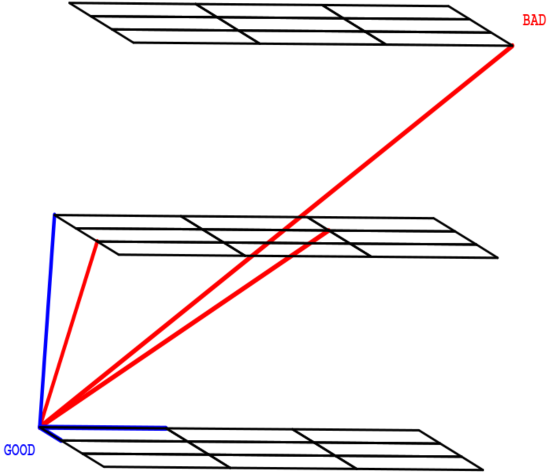
Changing basis of a lattice

- If $\mathbf{b}_1, \ldots, \mathbf{b}_d$ and $\mathbf{b}'_1, \ldots, \mathbf{b}'_d$ are bases of the same lattice, then there exist matrices $U = (u_{i,j})_{i,j}, U' = (u'_{i,j})_{i,j} \in \mathbb{Z}^{d \times d}$ such that

$$\mathbf{b}'_i = u_{i,1}\mathbf{b}_1 + \ldots + u_{i,d}\mathbf{b}_d, \qquad \mathbf{b}_i = u'_{i,1}\mathbf{b}'_1 + \ldots + u'_{i,d}\mathbf{b}'_d \qquad \text{for all } i,$$

  so $UU' = U'U = \mathbb{I}_d$ and therefore $\det U = \pm 1$.
- **Unimodular transformations**: $d \times d$ integral matrices with det ± 1
- Building blocks:
  - permutation of vectors,
  - adding to a given basis vector another basis vector
- Good bases are made of **short** and nearly **orthogonal** vectors

Two bases of a 3-dimensional lattice



Lattice volume: $\mathrm{vol}(L)$

- If $L = L(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, then

$$\mathrm{vol}(L) = \mathrm{vol}(\Pi(\mathbf{b}_1, \ldots, \mathbf{b}_d)),$$

  the d-dimensional volume of the parallelepiped spanned by the $\mathbf{b}_i$'s.
- Special case: $d = n$, then

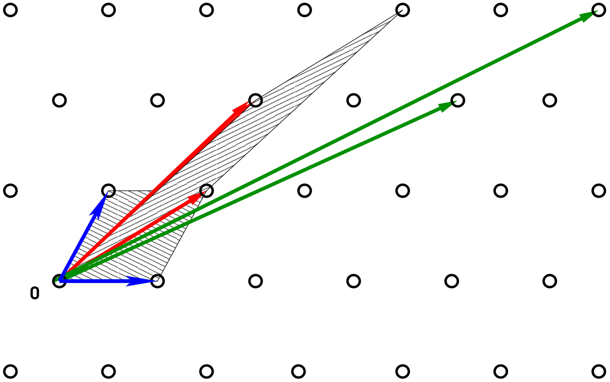
$$\mathrm{vol}(L) = \left| \det \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} \right|$$

- In general, $\mathrm{vol}(L) = (\det G(\mathbf{b}_1, \ldots, \mathbf{b}_d))^{1/2}$, where G is the Gram matrix of the $\mathbf{b}_i$'s:

$$G = (\langle \mathbf{b}_i, \mathbf{b}_j \rangle)_{i,j}$$

- The lattice volume is an **invariant**: independent of choice of basis!
- **Orthogonality defect** $\frac{\|\mathbf{b}_1\| \cdots \|\mathbf{b}_d\|}{\mathrm{vol}(L)}$ measures the quality of a basis $\mathbf{b}_1, \ldots, \mathbf{b}_d$ of L

Lattice volume: vol(L)



How to compute vol(L)?

- Easy if basis is known, i.e. compute using definition.
- MAGMA :

```
> M := Matrix(Integers(), 2, 3, [1, 7, -2,
                                  4, -1, 1]);
> L := Lattice(M); // builds lattice from rows of M
> Determinant(L);
947
```

- Warnings:
  - `Determinant(L)` returns $\det G(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, so volume squared
  - `Lattice(M)` automatically performs LLL lattice reduction (see later), avoid by using `LatticeWithBasis(M)`

How to compute vol(L)?

- Sometimes useful: if $L_1 \subset L_2$ and L_2/L_1 finite then

$$\mathrm{vol}(L_1) = \mathrm{vol}(L_2) \times [L_2 : L_1]$$

- Example usage: take $a_1, \ldots, a_n \in \mathbb{Z}$ and $N \in \mathbb{Z}_{\geq 2}$ with $\gcd(a_1, \ldots, a_n, N) = 1$ and consider

$$L = \{\, (x_1, \ldots, x_n) \in \mathbb{Z}^n \,|\, a_1x_1 + a_2x_2 + \ldots + a_nx_n \equiv 0 \mod N \,\}$$

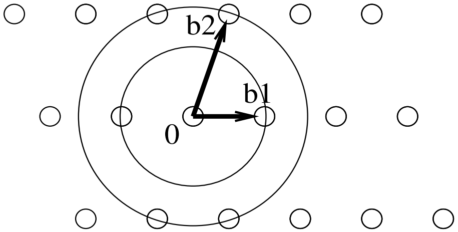
  - **Exercise:** this is a lattice of dimension n
  - L is the kernel of

$$\phi : \mathbb{Z}^n \to \mathbb{Z}/N\mathbb{Z} : (x_1, \ldots, x_n) \mapsto \sum a_i x_i \bmod N$$

  - **Exercise:** ϕ is surjective, so $\mathbb{Z}^n/L \cong \mathbb{Z}/N\mathbb{Z}$
  - Conclusion: $\mathrm{vol}(L) = \mathrm{vol}(\mathbb{Z}^n) \cdot [\mathbb{Z}^n : L] = 1 \cdot N = N$

Lattice minima: $\lambda_i(L)$

- $\lambda_1(L) :=$ the length of a shortest non-zero vector in L
  - Shortest non-zero vector is never unique since $||v|| = ||-v||$
  - Number can grow exponentially with dimension ("kissing number")
- More generally, for $1 \leq i \leq d$, we define $\lambda_i(L)$ as the minimum radius r for which $B(\mathbf{0}, r)$ contains i *linearly independent* lattice vectors
- Can take linearly independent vectors in L reaching λ_i's, but not necessarily possible to find a **basis** of such vectors (see later)!



Gaussian heuristic

- Assume L full rank lattice in $\mathbb{R}^d$
- **Lemma:** for any $r > 0$, denote by $s_L(r)$ the number of $\mathbf{x} \in L$ with $||\mathbf{x}|| < r$, then

$$\lim_{r\to\infty} \frac{s_L(r)}{r^d v_d} = \frac{1}{\mathrm{vol}(L)}$$

  with

$$v_d = \frac{\pi^{d/2}}{\Gamma(1 + d/2)}$$

  the volume of the unit ball in $\mathbb{R}^d$ (we have $v_d = \pi^{d/2}/(d/2)!$ if d is even).
- **Gaussian heuristic**: for a measurable set $C \subset \mathbb{R}^d$, the number of lattice points inside C is roughly $\mathrm{vol}(C)/\mathrm{vol}(L)$
- Gaussian heuristic + Stirling: we expect $\lambda_1(L) \approx \sqrt{\frac{d}{2\pi e}}\mathrm{vol}(L)^{1/d}$

Minkowski's convex body theorem

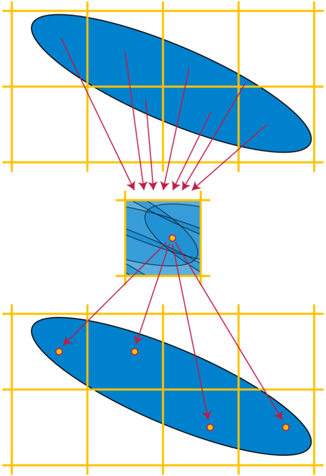
- **Theorem (Minkowski, 1889)**: For any full rank lattice L and any measurable convex set C symmetric about the origin:

$$\text{if } \mathrm{vol}(C) > 2^d \mathrm{vol}(L) \text{ then } (C \cap L) \setminus \{\mathbf{0}\} \neq \emptyset$$

- Proof is based on continuous version of **pigeon-hole principle**: if $N+1$ pigeons are put in N cages, there is at least 1 cage with at least 2 pigeons

- In our case: for any lattice L and any measurable set C with $\mathrm{vol}(C) > \mathrm{vol}(L)$, there exists two distinct points $\mathbf{z}_1, \mathbf{z}_2 \in C$ such that $\mathbf{z}_1 - \mathbf{z}_2 \in L$

- Sometimes called **Blichfeldt's lemma**



Minkowski's convex body theorem: proof

- Consider scaled lattice $2L$
- Note: $\mathrm{vol}(2L) = 2^d \mathrm{vol}(L)$, so $\mathrm{vol}(C) > \mathrm{vol}(2L)$
- Pick distinct points $\mathbf{z}_1, \mathbf{z}_2 \in C$ such that
$$\mathbf{z}_1 - \mathbf{z}_2 \in 2L$$
- But then
$$\frac{1}{2}\mathbf{z}_1 + \frac{1}{2}(-\mathbf{z}_2) \in L$$
so theorem follows because of convexity and symmetry.
- **Exercise**: give examples showing the necessity of the convexity and symmetry assumptions

Consequences

- Consequence:
$$\lambda_1(L) \leq 2\left(\frac{\mathrm{vol}(L)}{v_d}\right)^{1/d} \leq \sqrt{d}\,\mathrm{vol}(L)^{1/d}$$
- Can be generalized to bound geometric mean of $\lambda_i(L)$'s:
$$\left(\prod_{i=1}^{k}\lambda_i(L)\right)^{1/k} \leq 2\left(\frac{\mathrm{vol}(L)}{v_d}\right)^{1/d} \leq \sqrt{d}\,\mathrm{vol}(L)^{1/d}$$
- **Minkowski's second theorem**: for $k = d$ also have lower bound $2\left(\frac{\mathrm{vol}(L)}{d!v_d}\right)^{1/d}$
- **Ajtai**: for "random" lattices L the expected successive minima are
$$\lambda_1(L) \approx \lambda_2(L) \approx \ldots \approx \lambda_d(L) \approx \sqrt{\frac{d}{2\pi e}}\mathrm{vol}(L)^{1/d}$$
(compare with Gaussian heuristic)

Application 1

- Goal: write $r \in \mathbb{F}_p$ as fraction $\frac{x}{y}$ with x, y small
- Strategy: solutions to $x - ry = 0 \bmod p$ form lattice L with volume p
- Minkowski: should be possible to find non-zero $(x, y) \in L$ with

$$||(x,y)|| \leq 2\sqrt{\frac{p}{\pi}} < \sqrt{p}$$

- Magma :

```
> p := NextPrime(2^128 : Proof := false);  r := Random(p);
> M := Matrix(Integers(), 2, 2, [[ p, 0 ], [r, 1]]);
> M;
[340282366920938463463374607431768211507      0]
[278999982678262827282714582240079804296      1]
> ShortestVectors(L)
[
    ( 12577160292907500815 -10887493637333730493)
]
```

Application 2: Diophantine approximation

- Let $\alpha \in \mathbb{R} \setminus \mathbb{Q}$. Claim: can find integers m, n with n arbitrarily large such that

$$\left|\alpha - \frac{m}{n}\right| < \frac{1}{n^2}$$

- Proof: for any B we can consider lattice spanned by rows of

$$\begin{pmatrix} 1 & 0 \\ \alpha & B^{-2} \end{pmatrix}$$

- Minkowski: there exist $m, n \in \mathbb{Z}$ (can assume n positive) such that

$$||(m - n\alpha, -nB^{-2})|| = ||m(1,0) - n(\alpha, B^{-2})|| < B^{-1}$$

- Implies $|m - n\alpha| < B^{-1} < \frac{1}{n}$ (and can do this for infinitely many n because B can be taken arbitrarily large)
- **Exercise:** conclude that set $\mathbb{Z} \cdot 1 + \mathbb{Z} \cdot \sqrt{2}$ is not discrete (cf. slide 4)

Application 3: finding minimal polynomials

- Given: approximation α of a root of some degree-d polynomial $f(x) \in \mathbb{Z}[x]$.
- Goal: reconstruct $f(x)$.
- Idea: for large N, consider lattice spanned by rows of

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 & N \\ 0 & 1 & \dots & 0 & \lfloor N\alpha \rfloor \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \lfloor N\alpha^d \rfloor \end{pmatrix},$$

then if $f(x) = a_0 + a_1 x + \ldots + a_d x^d$ for small a_i, then $(a_0, a_1, \ldots, a_d)L$ is likely a shortest vector since last entry is also very small.

Application 3: finding minimal polynomials

- Exampe:
```
> alp := 0.68232; d := 3; N := 100;
> M := Matrix(4, 5, [1,0,0,0,N,
                     0,1,0,0,Floor(N*alp),
                     0,0,1,0,Floor(N*alp^2),
                     0,0,0,1,Floor(N*alp^3)];
> L := Lattice(M)
> ShortestVectors(L);
[
    ( 1 -1  0 -1  0)
]
```
- So $\alpha = 0.68232\ldots$ is probably a root of $x^3 + x - 1$.
- Check:
```
> R<x> := PolynomialRing(RealField());
> Roots(x^3 + x - 1);
[ <0.682327803828019327369483739711, 1> ]
```

Definition and first facts

Lattice Problems

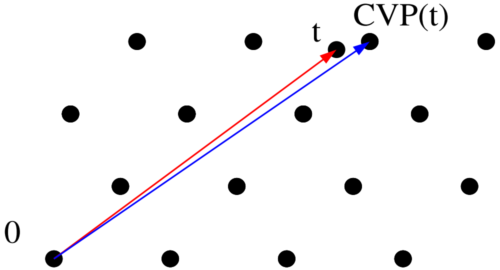
Lattice Reduction Algorithms

The shortest vector problem: SVP

- Given a basis of L, compute a vector of length $\lambda_1(L)$
- γ-SVP: Compute a vector of length $\leq \gamma \cdot \lambda_1(L)$.
- Expected solution: a vector of length $\approx \mathrm{vol}(L)^{1/d}$
- If $\lambda_1(L)$ is much shorter than this, then problem typically becomes easier

The closest vector problem: CVP

- Given a basis of L and a vector $\mathbf{t}$ of the embedding space, compute a lattice vector closest to $\mathbf{t}$
- γ-CVP: Given a basis of L and a target vector $\mathbf{t}$, compute a lattice vector $\mathbf{v}$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \gamma \cdot \min_{\mathbf{b} \in L} \|\mathbf{b} - \mathbf{t}\|$



More on CVP

- A "general" solution should be at distance $\mathrm{vol}(L)^{1/d}$ of $\mathbf{t}$
- Intuition of the difficulty: Consider $\mathbf{t} = (1/2, \ldots, 1/2)$ and slightly shake $\mathbb{Z}^d$. Which one of the 2^d vertices is the solution?
- CVP is considered harder than SVP

Hardness results

Solving these problems for a small γ is infeasible

- CVP: NP-hard under deterministic reductions, even with preprocessing (van Emde Boas, Micciancio)
- γ-SVP: NP-hard under randomized reductions for $\gamma < 2$, and for $\gamma = O(2^{\sqrt{\log n} - \varepsilon})$ under a reasonable assumption (Ajtai, Micciancio, Khot)
- γ-SVP: not NP-hard for $\gamma \geq \frac{\sqrt{n}}{\log n}$ under a reasonable assumption (Goldreich & Goldwasser)
- Random instances of n^c-SVP are not easier than worst-case instances when c is larger than some constant (Ajtai, Regev)

Definition and first facts

Lattice Problems

Lattice Reduction Algorithms

Definition of reduced basis?

- A reduced basis is made of "rather orthogonal and short vectors"
- Basis is reduced if lengths of vectors are "close" to the $\lambda_i(L)$'s
- Famously and unfortunately, a basis reaching the $\lambda_i(L)$'s is not always possible:
$$L = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$
Indeed, $\lambda_1(L) = \ldots = \lambda_5(L) = 2$, but any basis made of norm-2 vectors does not contain $(1, 1, 1, 1, 1)$
- No "best" definition.

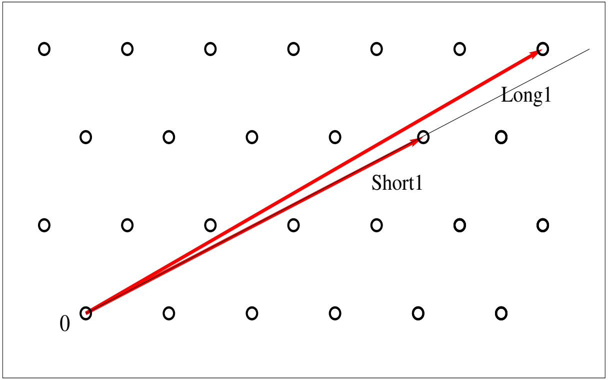
Definition of reduced basis?

- Common definitions make a trade-off between quality and computation time.
- Very strong notions, but hard to compute:
  - **Minkowski-reduced**: each $\mathbf{b}_i$ is shortest vector such that $(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ can be extended to basis
  - **HKZ** (Hermite-Korkine-Zolotarev): $\mathbf{b}_1$ is the shortest vector and projection of $(\mathbf{b}_2, \ldots, \mathbf{b}_d)$ on $\mathbf{b}_1^{\perp}$ is HKZ
- Weaker notion, but easier to compute:
  - **LLL** (Lenstra-Lenstra-Lovász)
- Trade-off:
  - **BKZ**$_k$ (Block Korkine-Zolotarev with block size k), specializes to LLL for $k = 2$ and to HKZ for $k = d$

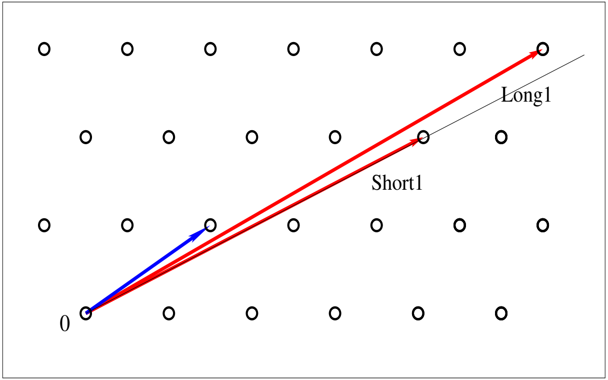
The 2-dimensional case

- Completely solved by Gauss (Lagrange?) algorithm
- Vectorial generalization of Euclid's algorithm
- Running time: $O(\log^2 B)$, where $B = \max(\|\mathbf{a}\|, \|\mathbf{b}\|)$
- Can be generalized up to dimension 4
- Algorithm: shorten the long vector by adding to it an integer multiple of the short one, while possible

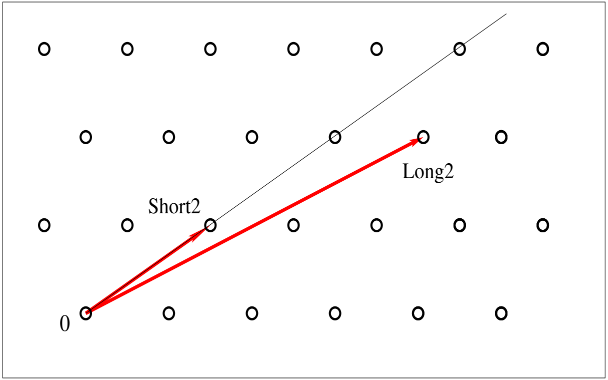
The 2-dimensional case



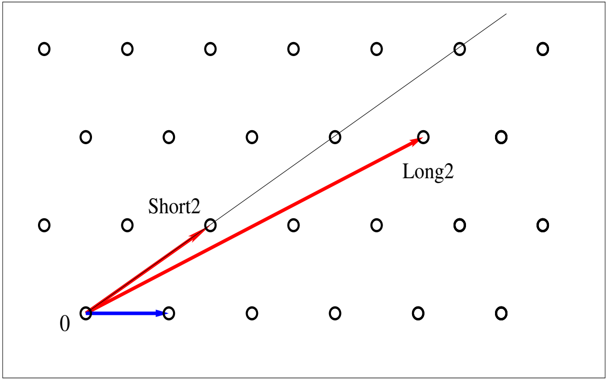
The 2-dimensional case



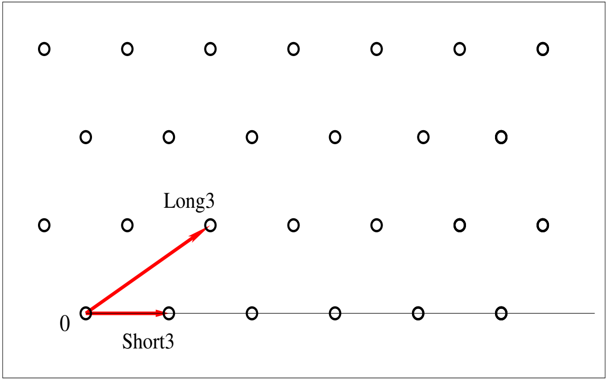
The 2-dimensional case



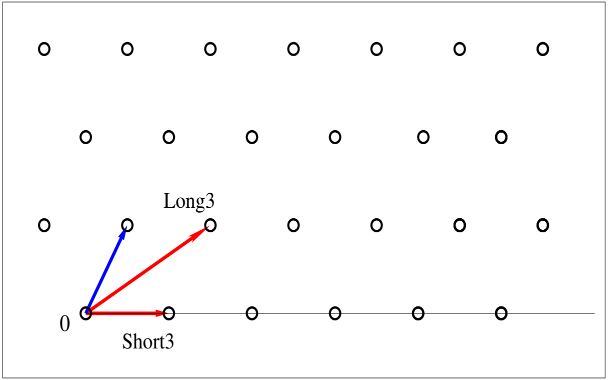
The 2-dimensional case



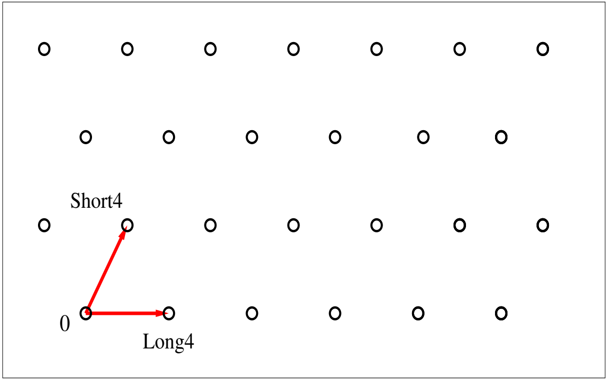
The 2-dimensional case



The 2-dimensional case



The 2-dimensional case



Gauss reduction for Euclidean norm

Input: $(\mathbf{a}, \mathbf{b})$ linearly independent.
Output: A Gauss reduced basis $(\mathbf{a}', \mathbf{b}')$
with $||\mathbf{a}'|| = \lambda_1(L)$ and $||\mathbf{b}'|| = \lambda_2(L)$

1. Repeat
2. $\quad \mathbf{r} := \mathbf{b} - q\mathbf{a}$ with $q = \lceil \frac{\langle \mathbf{b},\mathbf{a} \rangle}{\langle \mathbf{a},\mathbf{a} \rangle} \rfloor$
3. $\quad \mathbf{b} := \mathbf{a}$
4. $\quad \mathbf{a} := \mathbf{r}$
5. as long as $||\mathbf{a}|| < ||\mathbf{b}||$
6. $\mathbf{a}' := \mathbf{b}$, $\mathbf{b}' := \mathbf{a}$

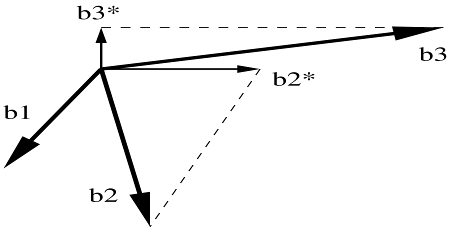
Algorithms in low dimension

- Suppose we want a d-dimensional HKZ-reduced basis
- For small d, exponential algorithms remain feasible
- Algorithms: Kannan, Ajtai-Kumar-Sivakumar
- SVP and CVP can be solved efficiently in practice up to dimension $\approx 45 - 50$

Gram–Schmidt orthogonalization

- Iterative process orthogonalizing $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$.
- $\mathbf{b}_i^*$ is the component of $\mathbf{b}_i$ orthogonal to $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$:

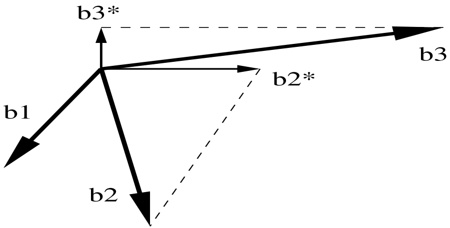
$$\mathbf{b}_1^* = \mathbf{b}_1, \; \mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*, \; \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}$$



Gram–Schmidt orthogonalization

- Iterative process orthogonalizing $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$.
- $\mathbf{b}_i^*$ is the component of $\mathbf{b}_i$ orthogonal to $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$:

$$\mathbf{b}_1^* = \mathbf{b}_1, \ \mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*, \ \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2}$$



- Note matrix of base change has det 1, so volume does not change

Gram–Schmidt orthogonalization

- Let $(\mathbf{b}_1^*, \ldots, \mathbf{b}_d^*)$ be the GS orthogonalization of $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, then

$$\lambda_1(L(\mathbf{b}_1, \ldots, \mathbf{b}_d)) \geq \min_{i=1,\ldots,d} ||\mathbf{b}_i^*|| \,.$$

- Write $\mathbf{v} = \sum_{i=1}^{d} x_i \mathbf{b}_i$ with $x_i \in \mathbb{Z}$ not all zero
- Let j be maximal such that $x_j \neq 0$, then

$$|\langle \mathbf{v}, \mathbf{b}_j^* \rangle| = |x_j| \cdot ||\mathbf{b}_j^*||^2$$

- Cauchy–Schwarz: $|\langle \mathbf{v}, \mathbf{b}_j^* \rangle| \leq ||\mathbf{v}|| \cdot ||\mathbf{b}_j^*||$
- Conclusion: $||\mathbf{v}|| \geq |x_j| \cdot ||\mathbf{b}_j^*|| \geq \min_{i=1,\ldots,d} ||\mathbf{b}_i^*||$
- Try to balance the norms of the Gram–Schmidt vectors ...

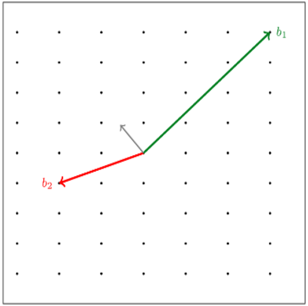
Size reduction

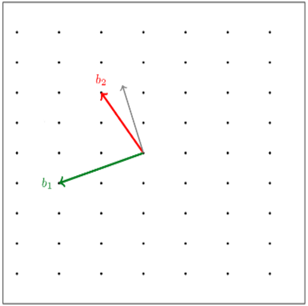
- Integral version of Gram–Schmidt.
- Given $(\mathbf{b}_1, \ldots, \mathbf{b}_k)$, we add to $\mathbf{b}_k$ an integer linear relation of the previous vectors such that: $\forall i < k, \ |\mu_{k,i}| \leq 1/2$

1. Compute the $\mu_{k,i}$'s for $i < k$.
2. For $i = (k-1)$ to 1,
3. $\quad x_i := \lceil \mu_{k,i} \rfloor$, $\mathbf{b}_k := \mathbf{b}_k - x_i \mathbf{b}_i$,
4. $\quad$ For $j = 1$ to i, $\mu_{k,j} := \mu_{k,j} - x_i \mu_{i,j}$.

- Size reduction depends on the order of the input vectors and does not change this order

Size reduction





- Left: $\mathbf{b}_1, \mathbf{b}_2$ is size reduced
- Right: swapping roles of $\mathbf{b}_1, \mathbf{b}_2$ and size-reducing improves basis ($=$ Gauss)
- Observe: better balance of Gram–Schmidt vectors

Algorithms in high dimension: LLL

- LLL algorithm (Lenstra, Lenstra, Lovász - 1982)
- Gives an LLL-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ with:

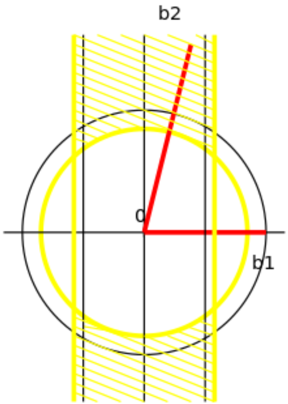
$$\begin{aligned} \|\mathbf{b}_1\| &\leq c^d \cdot \mathrm{vol}(L)^{1/d} \\ \|\mathbf{b}_i\| &\leq c^{2d} \cdot \lambda_i(L) \end{aligned}$$

  where $c \approx 1.075$. In practice $c \approx 1.02$
- Time: $O(d^5 n \log^3 B)$, with $B = \max_{i \leq d} \|\mathbf{b}_i^{init}\|$
- With floating-point arithmetic: $O(d^4 n (d + \log B) \log B)$

The LLL-reduction (1982)

- Size reduced + condition on projection of pairs of basis vectors (local condition).
- $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is LLL-reduced iff ($\delta \in]1/4, 1]$):
  1. $\forall i > j, |\mu_{i,j}| \leq 1/2$ [Size]
  2. $\forall i, \delta \|\mathbf{b}_{i-1}^*\|^2 \leq \|\mathbf{b}_i^* + \mu_{i,i-1}\mathbf{b}_{i-1}^*\|^2$ [Lovász]
- 2 means: in $(\mathbf{b}_1, \ldots, \mathbf{b}_{i-2})^\perp$, $\mathbf{b}_{i-1}$ is approx. shorter than $\mathbf{b}_i$
- Only depends on projection of $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ on $(\mathbf{b}_{i-1}^*, \mathbf{b}_i^*)$
- 1 and 2 imply $\|\mathbf{b}_i^*\| \geq \sqrt{\delta - \frac{1}{4}} \cdot \|\mathbf{b}_{i-1}^*\|$
- Lengths of the Gram–Schmidt vectors cannot drop too fast

The Lovasz condition



Picture created by Damien Stehlé

Why is the LLL-reduction interesting?

- We can compute it in polynomial time
- The output vectors are not too long ($\delta = 1 - \varepsilon$):

$$\begin{aligned} \|\mathbf{b}_1\| &\leq c^d \cdot \mathrm{vol}(L)^{1/d} \\ \|\mathbf{b}_i\| &\leq c^{2d} \cdot \lambda_i(L) \end{aligned}$$

  where $c = (4/3 + \varepsilon)^{1/4} \approx 1.075$. In practice $c \approx 1.02$
- Note: still exponential factor difference with shortest vector

The LLL algorithm

Input: $\mathbf{b}_1, \ldots, \mathbf{b}_d$ linearly independent.
Output: An L^3-reduced basis of $L(\mathbf{b}_1, \ldots, \mathbf{b}_d)$.

1. [GS] Compute the $\mu_{i,j}$'s and $\|\mathbf{b}_i^*\|^2$'s.
2. $\kappa := 2$. While $\kappa \leq d$,
3. [Size?] Size-reduce $\mathbf{b}_\kappa$ with respect to $\mathbf{b}_1, \ldots, \mathbf{b}_{\kappa-1}$.
4. [Lovász?] If $(\delta - \mu_{\kappa,\kappa-1}^2)\|\mathbf{b}_{\kappa-1}^*\|^2 \leq \|\mathbf{b}_\kappa^*\|^2$, then $\kappa := \kappa + 1$.
5. Else swap $\mathbf{b}_{\kappa-1}$ and $\mathbf{b}_\kappa$, $\kappa := \max(\kappa - 1, 2)$.

LLL-reduction: more general

- A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is (δ, η)-LLL-reduced if:
  1. $\forall i > j, \ |\mu_{i,j}| \leq \eta$
  2. $\forall i, \ \delta \cdot \|\mathbf{b}_{i-1}^*\|^2 \leq \|\mathbf{b}_i^* + \mu_{i,i-1}\mathbf{b}_{i-1}^*\|^2$,
     where $\delta \in (0.25, 1)$ and $\eta \in (0.5, \sqrt{\delta})$
- Often $(\delta, \eta) = (0.999, 0.501)$

Properties of LLL-reduced bases

1. $\|b_1\| \leq \left(\delta - \eta^2\right)^{-(d-1)/4} \cdot (\mathrm{vol} L)^{1/d}$
2. $\|b_1\| \leq (\delta - \eta^2)^{-(d-1)/2} \cdot \lambda_1(L)$
3. $\prod_{i=1}^{d} \|b_i\| \leq (\delta - \eta^2)^{-d(d-1)/4} \cdot (\mathrm{vol} L)$
4. $\forall j < i,\ \|b_j^*\| \leq (\delta - \eta^2)^{(j-i)/2} \cdot \|b_i^*\|$

Algorithms in high dimension

- LLL + HKZ reduction = Schnorr's Block-Korkine-Zolotarev algorithm
- Examples: LLL == BKZ_2, HKZ == BKZ_d
- BKZ_k costs $\approx k^{O(k)}$ and gives $\gamma = k^{O(n/k)}$ for SVP $\Rightarrow$ Best γ for deterministic polynomial time: $2^{O\left(k\frac{(\log\log k)^2}{\log k}\right)}$
- BKZ is feasible for $k \leq 45$ to 50

Algorithms in high dimension

- LLL + HKZ reduction = Schnorr's Block-Korkine-Zolotarev algorithm
- Examples: LLL == BKZ_2, HKZ == BKZ_d
- BKZ_k costs $\approx k^{O(k)}$ and gives $\gamma = k^{O(n/k)}$ for SVP $\Rightarrow$ Best γ for deterministic polynomial time: $2^{O\left(k\frac{(\log\log k)^2}{\log k}\right)}$
- BKZ is feasible for $k \leq 45$ to 50
- MAGMA : `HKZ(L);` `BKZ(L, k);`

Example of LLL

- Lattice L given by the rows of

$$\begin{pmatrix} 853539607829 & 0 & 0 & 0 & 0 \\ 512469270672 & 1 & 0 & 0 & 0 \\ 487596978484 & 0 & 1 & 0 & 0 \\ 112511841846 & 0 & 0 & 1 & 0 \\ 24050211137 & 0 & 0 & 0 & 1 \end{pmatrix}$$

- After call to LLL, L is given by better basis

$$\begin{pmatrix} 62 & -115 & -56 & -26 & -19 \\ 27 & -37 & 163 & -156 & 59 \\ 4 & 82 & 65 & -97 & -198 \\ -256 & -133 & -75 & -82 & 89 \\ 91 & 151 & -174 & -255 & 48 \end{pmatrix}$$

Algorithms for the CVP

- γ-CVP: Given a basis of L and a target vector $\mathbf{t}$, compute a lattice vector $\mathbf{v}$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \gamma \cdot \min_{\mathbf{b} \in L} \|\mathbf{b} - \mathbf{t}\|$
- Babai's rounding algorithm
- Babai's nearest plane algorithm
- Embedding technique

Babai's rounding algorithm

- Compute a reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_d$ of L
- Write t in this basis

$$t = \sum_{i=1}^{d} x_i \mathbf{b}_i$$

- Define $v = \sum_{i=1}^{d} \lceil x_i \rfloor \mathbf{b}_i$
- Theorem: assume the basis is LLL-reduced for $\delta = 3/4$, then

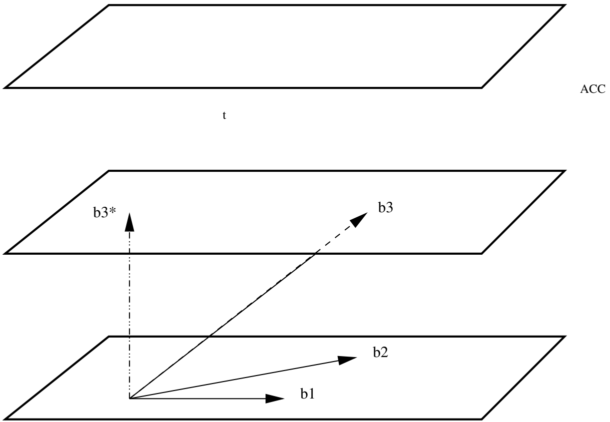
$$||\mathbf{v} - \mathbf{t}|| \leq (1 + 2d\,(9/2)^{d/2})||\mathbf{b} - \mathbf{t}||$$

for all $\mathbf{b} \in L$

Babai's nearest plane algorithm

- Compute a reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_d$ of L
- Let $H = \sum_{i=1}^{d-1} \mathbb{R}\mathbf{b}_i$ the hyperplane and $L' = L \cap H$
- Find $\mathbf{u} \in L$ such that distance of $\mathbf{t}$ to $\mathbf{u} + H$ is minimal
- Let $\mathbf{t}'$ be the orthogonal projection of $\mathbf{t} - \mathbf{u}$ on H
- Find $\mathbf{v}' \in L'$ close to $\mathbf{t}'$
- Return $\mathbf{v} = \mathbf{v}' + \mathbf{u}$

Babai's nearest plane algorithm



Babai's nearest plane algorithm

- Write $\mathbf{t} = \sum_i^d x_i \mathbf{b}_i^*$
- Then can take: $\mathbf{u} = \lceil x_d \rfloor \mathbf{b}_d$
- And then: $\mathbf{t}' = \sum_i^{d-1} (x_i - \lceil x_d \rfloor \mu_{d,i}) \mathbf{b}_i^*$
- Repeat and return sum of all $\mathbf{u}$'s
- Theorem: assume the basis is LLL-reduced for $\delta = 3/4$, then

$$||\mathbf{v} - \mathbf{t}|| \leq 2^{d/2} ||\mathbf{b} - \mathbf{t}||$$

for all $\mathbf{b} \in L$

Embedding technique

- Heuristic only: in general no proven bounds on approximation
- Idea: construct lattice L' of dimension $d+1$ with basis vectors

$$\begin{pmatrix} \mathbf{b}_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} \mathbf{b}_d \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{t} \\ 1 \end{pmatrix}$$

- Let $\mathbf{v} = \sum_{i=0}^{d} x_i \mathbf{b}_i$ be solution to the CVP
- The vector $(\mathbf{t} - \mathbf{v}, 1)$ is short vector in L'
- Simply run SVP-solver on L'

Reading material

- Chapter's 16 to 19 of Steven Galbraith's book:
  `www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html`
- Public key cryptanalysis by Phong Q. Nguyen:
  see Toledo
- Hermite's constant and lattice algorithms by Phong Q. Nguyen:
  `www.di.ens.fr/~pnguyen/Nguyen_HermiteConstant.pdf`