FUNDAMENTAL LATTICE ALGORITHMS

PHONG NGUYEN

http://www.di.ens.fr/~pnguyen

September 2024

EASY PROBLEMS

- We restrict to integer lattices: a basis is represented by an integer matrix.
- Given a basis of an integer lattice L and a target v, decide if $v \in L$ and if so, output its basis coefficients.
- Given bases of two integer lattices L and M, decide if L=M, compute a basis of $L \cap M$.
- Given bases of an integer lattice L and a subspace E, compute a basis of $L \cap E$.

THE HERMITE NORMAL FORM

FULL-RANK INTEGER LATTICES

- Let L be a full-rank lattice in $\mathbb{Z}^n$.
- Then the quotient $\mathbb{Z}^n/L$ is a finite Abelian group of order covol(L).
- In particular, $\text{covol}(L)\mathbb{Z}^n \subseteq L$.
  - L has a full-rank family of orthogonal vectors.
  - But this family is usually not a $\mathbb{Z}$-basis of L.

LOWER-TRIANGULAR BASES

- By multiplying the canonical vectors by covol(L), we can derive a $\mathbb{Z}$-basis of L which spans the same subspaces, and is therefore lower-triangular.
- This triangular form can be made unique if we ensure that:
  - The diagonal is > 0.
  - In each column, all coefficients ≥ 0 and $<$ the diagonal coefficient.
- This is the Hermite normal form.

COMPUTING THE HNF

- Given a basis $b_1,\ldots,b_n$ of a full-rank lattice L in $\mathbf{Z}^n$, we can compute in deterministic polynomial time the Hermite normal form of L.
- First, we compute D=covol(L) = $|\det(b_1,\ldots,b_n)|$.
- Then DI_n is the Hermite normal form of $D\mathbb{Z}^n$.

INCREMENTAL HNF

- Input: the HNF of a full-rank integer lattice M and an integer vector b.
- Output: the HNF of the lattice $M + \mathbb{Z}\vec{b}$.
- Let $\vec{a}$ be the last row of M.
- Consider the lattice $L = \mathbb{Z}\vec{a} + \mathbb{Z}\vec{b}$.
  - We can compute $\vec{a}'$and $\vec{b}'$ such that $L = \mathbb{Z}\vec{a}' + \mathbb{Z}\vec{b}'$ and the last coordinate $\vec{a}'$ of is zero: how?
  - What's next?

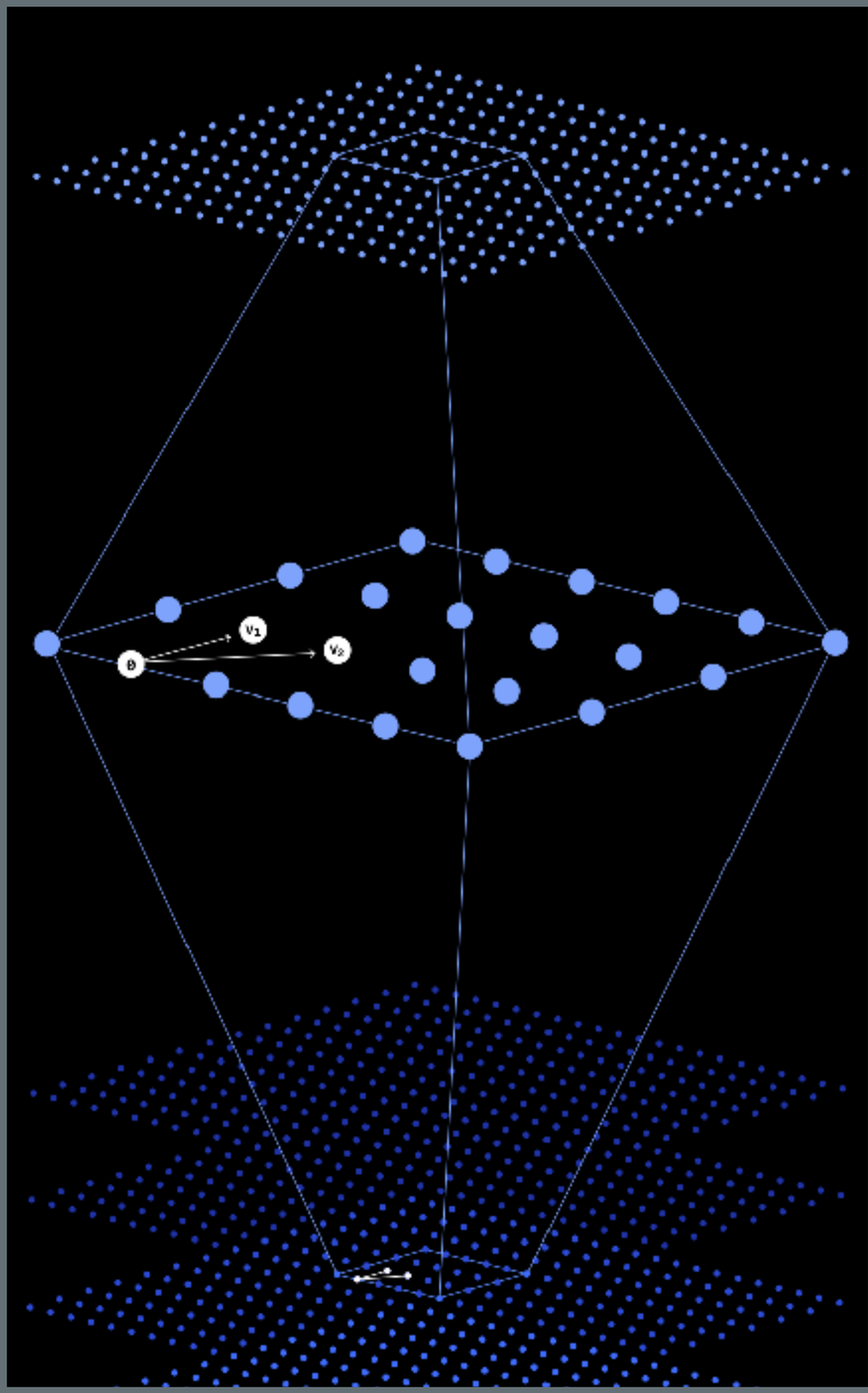
INCREMENTAL HNF

- Consider the lattice $L = \mathbb{Z}\vec{a} + \mathbb{Z}\vec{b}$.
  - Let α and β be the last coordinates of $\vec{a}$ and $\vec{b}$.
  - Using Euclid's XGCD, compute integers u and v s.t. $\alpha u + \beta v = \gcd(\alpha, \beta)$
    - Compute $\vec{a}' = (\beta/\gcd(\alpha,\beta))\vec{a} - (\alpha/\gcd(\alpha,\beta))b$ and $b' = u\vec{a} + vb$. Then $L = \mathbb{Z}\vec{a}' + \mathbb{Z}\vec{b}'$ and the last coordinate of $\vec{a}'$ is zero.
- By subtracting suitable multiples of the first n-1 rows of M, make all the coordinates of $\vec{a}'$ and $\vec{b}'$ between 0 and D.
- Apply recursively Incremental-HNF to the first n-1 rows of M, and $\vec{a}'$, by ignoring their last coordinate (which is zero).
- Use the new HNF to reduce $\vec{b}'$.

TAKE AWAY

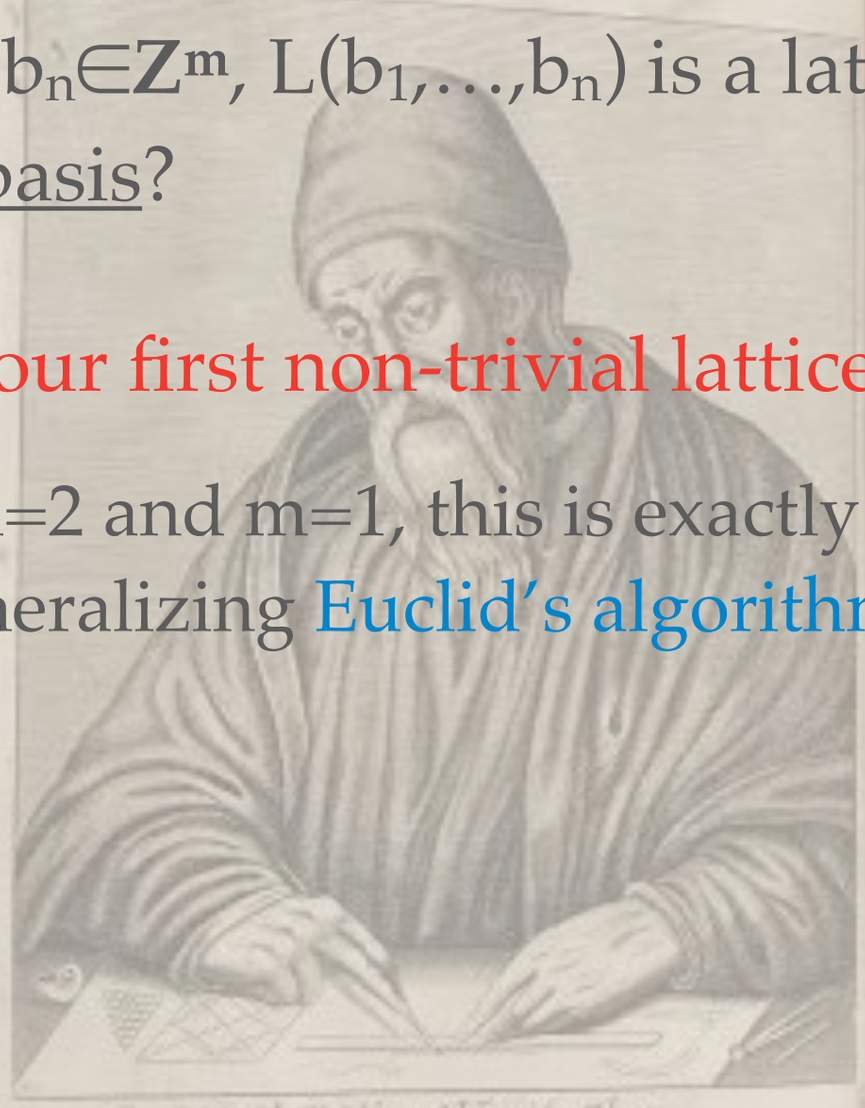
- Key ideas:
  - ➤ Projecting to simplify the problem.
  - ➤ Reducing with the diagonal to make sure that vectors do not get too big.

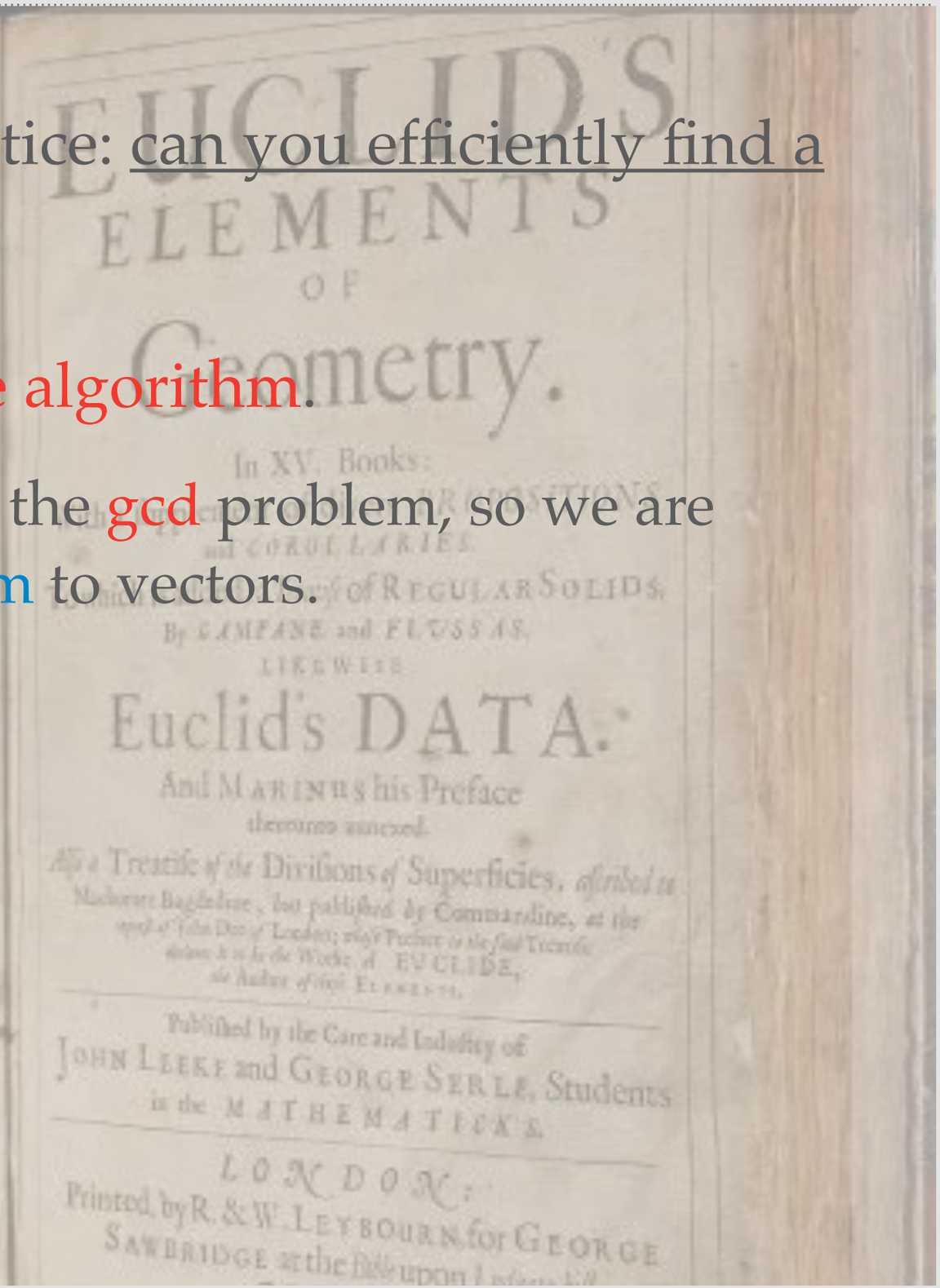
A NON-TRIVIAL LATTICE ALGORITHM



EUCLID WITH VECTORS

- If $b_1,\ldots,b_n \in \mathbf{Z}^m$, $L(b_1,\ldots,b_n)$ is a lattice: can you efficiently find a lattice basis?
- This is our first non-trivial lattice algorithm.
  - If n=2 and m=1, this is exactly the gcd problem, so we are generalizing Euclid's algorithm to vectors.





THE BASIS ALGORITHM

- INPUT: $b_1,\ldots,b_n \in \mathbf{Z}^m$ s.t. $b_1 \neq 0$
- OUTPUT: a basis of the lattice $L(b_1,\ldots,b_n)$ spanned by the b_i's.
- We may assume that $b_1,\ldots,b_{n-1}$ are linearly independent: by iterating this algorithm n times, we solve the original problem.

THE ITERATIVE BASIS ALGORITHM

- INPUT: $b_1,...,b_n \in \mathbf{Z}^m$ s.t. $b_1,\ldots,b_{n-1}$ are linearly independent.
- OUTPUT: a basis of the lattice $L(b_1,...,b_n)$ spanned by the b_i's.
- This means that $n-1 \leq \text{rank}(L(b_1,\ldots,b_n)) \leq n$.

- We need tools:
  - How can we decide linear independence efficiently?
  - How can we compute projections?

DUALITY

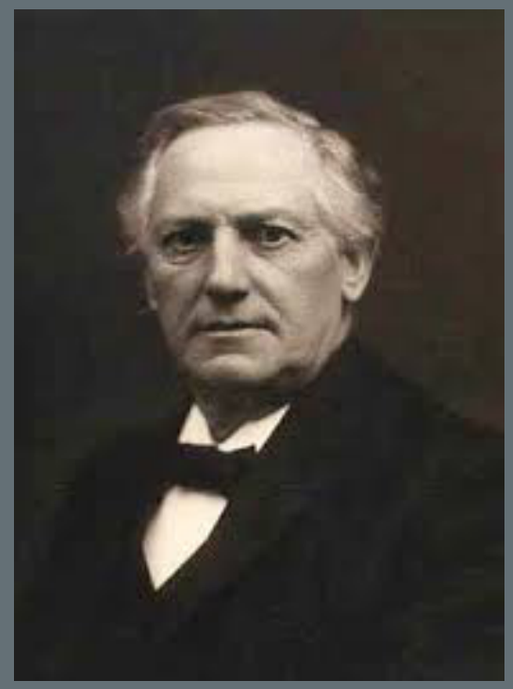
THE DUAL LATTICE

- Let L be a lattice.
- The dual lattice of L is $L^x=\{y\in\text{span}(L) \text{ s.t. } \langle x,y\rangle\in\mathbf{Z} \text{ for all } x\in L\}$
  - What is the dual of $\mathbb{Z}^n$?
  - Show that L^x is a lattice.
  - If B is a basis of L, then $(BB^t)^{-1}B$ is a basis of L^x.
  - $\text{rank}(L)=\text{rank}(L^x)$ and $\text{covol}(L)\text{covol}(L^x)=1$
  - L^x/L is a finite group of order $\text{covol}(L)^2$.

EX: KERNEL LATTICES

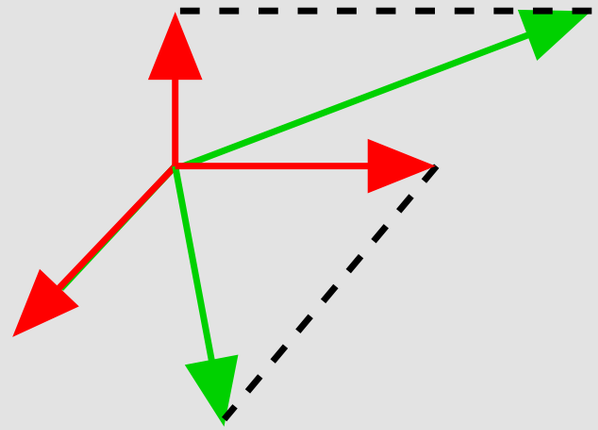
- Let $n,m,q \in \mathbf{N}$.
- Let A be an mxn matrix over $\mathbf{Z}$.
- The kernel $L_A=\{x \in \mathbf{Z}^m \text{ s.t. } xA \equiv 0 \bmod q\}$ is a full-rank lattice in $\mathbf{Z}^m$ s.t. $\mathrm{covol}(L_A) \mid q^n$.
- Its dual lattice is $(1/q)L'_A$ where L'_A is the «image» i.e. $L'_A=\{y \in \mathbf{Z}^m \text{ s.t. } y \equiv zA^t \bmod q \text{ for some } z \in \mathbf{Z}^n\}$

COMPUTATIONAL GRAM–SCHMIDT



GRAM-SCHMIDT

- Let $b_1,\ldots,b_n \in \mathbf{R}^m$.
- Its Gram-Schmidt Orthogonalization is $b_1^*,\ldots,b_n^* \in \mathbf{R}^m$ defined as:
  - $b_1^* = b_1$



  - For $2 \le i \le n$, b_i^* = projection of b_i over $\text{span}(b_1,\ldots,b_{i-1})^\perp$
- Classical tool to show the existence of orthonormal bases in Euclidean spaces.

FACTS

- $b_i^* \perp \text{span}(b_1,\ldots,b_{i-1})$, i.e. $\langle b_i^*, b_j \rangle = 0$ if $i>j$
- $b_i^* - b_i \in \text{span}(b_1,\ldots,b_{i-1}) = \text{span}(b_1^*,\ldots,b_{i-1}^*)$
  - Notation: $b_i - b_i^* = \sum_{j<i} \mu_{i,j} b_j^*$.
- $||b_i^*||$ = distance of b_i to $\text{span}(b_1,\ldots,b_{i-1})$.
- $\langle b_i^*, b_j^* \rangle = 0$ if $i \neq j$

FORMULAS

- Let $b_1,\ldots,b_n \in \mathbf{R}^m$ be **linearly independent**.
- Then all $b_j^* \neq 0$.
  - For $1 \leq j < i \leq n$, let $\mu_{i,j} = \dfrac{\langle \vec{b}_i, \vec{b}_j^* \rangle}{\|\vec{b}_j^*\|^2}$
  - Then: $\vec{b}_1^\star = \vec{b}_1$ $\qquad \vec{b}_i^\star = \vec{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \vec{b}_j^\star$
- Thus:

$$\|\vec{b}_i^*\|^2 = \|\vec{b}_i\|^2 - \sum_{j=1}^{i-1} \mu_{i,j}^2 \|\vec{b}_j^*\|^2$$

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

- This gives an algorithm, but not necessarily efficient: we want cheap operations on reasonably-sized numbers.

EFFICIENT COMPUTATIONS

- We only deal with integers, so assume that $\vec{b}_1, \ldots, \vec{b}_n \in \mathbb{Z}^m$ and let $M = \max_{1 \le i \le n} \|\vec{b}_i\|$.
- Define the following integers:
  - $d_0 = 1$
  - $d_i = \text{Gram}(\vec{b}_1, \ldots, \vec{b}_i) = \prod_{j=1}^{i} \|\vec{b}_j^*\|$ so $1 \le d_i \le M^{2i}$
- Then $\mu_{i,j}$ and $\|\vec{b}_i^*\|^2 \in \mathbb{Q}$, $\vec{b}_i^* \in \mathbb{Q}^m$

INTEGRAL GRAM-SCHMIDT

- Lemma: Let $b_1,\ldots,b_n \in \mathbf{Z}^m$ be linearly independent. Then for all $1 \le j < i \le n$:
  - $d_{i-1}\vec{b}_i^* \in L(\vec{b}_1, \ldots, \vec{b}_i) \subseteq \mathbb{Z}^m$ with $\|d_{i-1}\vec{b}_i^*\| \le M^{2i-1}$
  - $d_j\mu_{i,j} \in \mathbb{Z}$ with $|d_j\mu_{i,j}| \le M^{2j}$
- Proof: Let $L=L(b_1,\ldots,b_i)$ and denote by L^x its dual lattice. Then $[L^x:L]=\mathrm{covol}(L)^2=d_i$.
  - Note that $\vec{b}_i^*/\|\vec{b}_i^*\|^2 \in L^\times$
  - Therefore $[L^\times : L]\vec{b}_i^*/\|\vec{b}_i^*\|^2 \in L$ i.e. $d_{i-1}\vec{b}_i^* \in L(\vec{b}_1, \ldots, \vec{b}_i) \subseteq \mathbb{Z}^m$.

GRAM-SCHMIDT ALGORITHM

- Induction formulas can be rewritten with integers, giving an efficient algorithm.
- Let $\lambda_{i,j}=d_j\mu_{i,j}\in\mathbf{Z}$.

$$d_i = d_{i-1}\|\vec{b}_i\|^2 - \sum_{j=1}^{i-1} \frac{\lambda_{i,j}^2}{d_j d_{j-1}}$$

$$\lambda_{i,j} = d_{j-1}\langle \vec{b}_i, \vec{b}_j \rangle - \sum_{k=1}^{j-1} \frac{d_{j-1}\lambda_{j,k}\lambda_{i,k}}{d_k d_{k-1}}$$

- Could also derive explicitly b_i^*, but usually not needed

RECAP

- If $b_1,\ldots,b_n \in \mathbf{Z}^m$ are linearly independent, we can compute efficiently all the integers $d_i = \mathrm{Gram}(\vec{b}_1, \ldots, \vec{b}_i) = \prod_{j=1}^{i} \|\vec{b}_j^*\|^2$ and $\lambda_{i,j} = d_j \mu_{i,j} \langle \vec{b}_i, \vec{b}_j^* \rangle / \|\vec{b}_j^*\|^2$

$$\vec{b}_i^* = \vec{b}_i - \sum_{j=1}^{i-1} \frac{\lambda_{i,j}}{d_j} \vec{b}_j^*$$

$$\|\vec{b}_i^*\|^2 = \frac{d_i}{d_{i-1}}$$

APPLICATION: LATTICE MEMBERSHIP

- Let $b_1,\ldots,b_n \in \mathbf{Z}^m$ be linearly independent: let $L=L(b_1,\ldots,b_n)$.
- Given $t \in \mathbf{Z}^m$, decide if $t \in L$, and if so, find its integer coefficients in the decomposition $t=x_1b_1+\ldots+x_nb_n$.
- Then $\langle t,b_n^* \rangle = x_n \langle b_n,b_n^* \rangle = x_n ||b_n^*||^2$
- Letting $b_{n+1}=t$, then $x_n=\mu_{n+1,n}$
  - Derive x_n from Gram-Schmidt over $(b_1,\ldots,b_n,t)$,
  - Repeat with $t-x_nb_n$ and $L(b_1,\ldots,b_{n-1})$, etc.
- We can find efficiently $x_n, x_{n-1},\ldots x_1 \in \mathbf{Z}$
- By checking if $t=x_1b_1+\ldots+x_nb_n$, we can decide if $t \in L$.

APPLICATION: SIZE-REDUCTION

- Let $b_1,\ldots,b_d \in \mathbf{Z}^m$ be linearly independent.
- $B=(b_1,\ldots,b_d)$ is size-reduced if all $|\mu_{i,j}| \leq \frac{1}{2}$
- Th: There is an efficient algorithm to size-reduce B, without changing the Gram-Schmidt vectors.

- For i = 2 to d
  - For j = i-1 downto 1
    - size-reduce b_i with respect to b_j: make $|\mu_{i,j}| \leq 1/2$ by $b_i := b_i - \text{round}(\mu_{i,j}) b_j$
    - Update all $\mu_{i,j'}$ for $j' \leq j$.
- The translation does not affect the previous $\mu_{i',j'}$ where $i' < i$, or $i'=i$ and $j'>j$.

VISUALIZING SIZE-REDUCTION

- If we take an appropriate orthonormal basis, the matrix of the lattice basis becomes triangular.

$$\begin{pmatrix}
\|\vec{b}_1^*\| & 0 & 0 & \dots & 0 \\
\mu_{2,1}\|\vec{b}_1^*\| & \|\vec{b}_2^*\| & 0 & \dots & 0 \\
\mu_{3,1}\|\vec{b}_1^*\| & \mu_{3,2}\|\vec{b}_2^*\| & \|\vec{b}_3^*\| & \dots & 0 \\
\vdots & \dots & \dots & \dots & \vdots \\
\mu_{d,1}\|\vec{b}_1^*\| & \mu_{d,2}\|\vec{b}_2^*\| & \dots & \mu_{d,d-1}\|\vec{b}_{d-1}^*\| & \|\vec{b}_d^*\|
\end{pmatrix}$$

THE ITERATIVE BASIS ALGORITHM

- INPUT: $b_1,...,b_n \in \mathbf{Z}^m$ s.t. $b_1,\ldots,b_{n-1}$ are linearly independent.
- OUTPUT: a basis of the lattice $L(b_1,...,b_n)$ spanned by the b_i's.
- Let d be the dimension of $L(b_1,\ldots,b_n)$: $n-1 \leq d \leq n$.

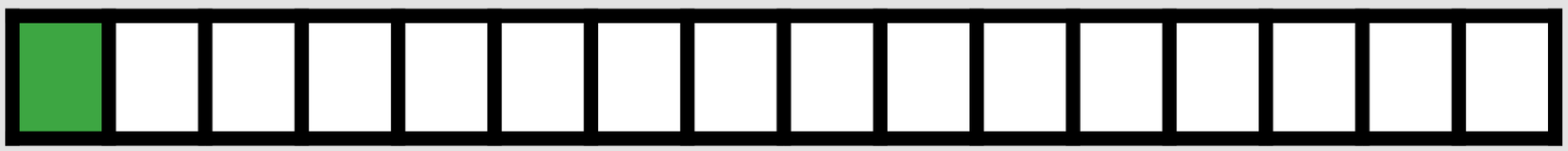
- Compute Integral Gram-Schmidt
- If $b_n^* \neq 0$
  - Return $(b_1,\ldots,b_n)$
- Else…

INTUITION

- Color in **green** the index $i \in \{1,\dots,n\}$ st. $b^*_i=0$
- Initially

- During the algorithm…we move the green cell to the left without ever changing $L(b_1,\dots,b_n)$:

- At the end



THE ITERATIVE BASIS ALGORITHM

- INPUT: $b_1,...,b_n \in \mathbf{Z}^m$ s.t. $b_1,\ldots,b_{n-1}$ are linearly independent.
- OUTPUT: a basis of the lattice $L(b_1,...,b_n)$ spanned by the b_i's.
- Let d be the dimension of $L(b_1,\ldots,b_n)$: $n-1 \leq d \leq n$.

- Compute Integral Gram-Schmidt
- If $b_n^* \neq 0$
  - Return $(b_1,\ldots,b_n)$
- Else For j=n downto 2
  - ColorSwap(b_{j-1},b_j)
  - Return $(b_2,\ldots,b_n)$ because $b_1=0$

COLOR-SWAP $B_J^*=0$ AND $B_{J-1}^*\neq 0$

- Let π be the projection over $\text{span}(b_1,\ldots b_{j-2})^{\perp}$.
- Then $\pi(b_{j-1})$ and $\pi(b_j)$ are linearly dependent.
- $b_{j-1}^* = \pi(b_{j-1})$ and $\pi(b_j)\in\mathbf{Q}b_{j-1}^*$ so $\pi(b_j)=(p/q)b_{j-1}^*$ with coprime p and q.
- Then $\pi(pb_{j-1}-qb_j)=0$: if we replace b_{j-1} by $pb_{j-1}-qb_j$ then $\text{NEW}(b_{j-1}^*)=0$, but what about b_j?
  - Using XGCD(p,q), find $(u,v)\in\mathbf{Z}^2$ s.t. $up+vq=1$
  - Replace (b_{j-1},b_j) by $(pb_{j-1}-qb_j,vb_{j-1}+ub_j)$: this transformation has det=+1 so the lattice is preserved.
  - Then $\text{NEW}(b_{j-1}^*)=0$ and $\text{NEW}(b_j^*)\neq 0$.

LIMITING SIZES

- We used XGCD to find a generator of the 1-dim lattice $L(\pi(b_{j-1}), \pi(b_j))$, then $NEW(b_{j-1}^*)=0$ and $NEW(b_j^*)\neq 0$.
- In particular, $||NEW(b_j^*)|| \leq ||OLD(b_j^*)||$.
- So $\max_i ||b_i^*||$ did not increase, but how about $\max_i ||b_i||$? We can keep it under control, using size-reduction.
- Now, $b_{j-1}^*=0$ and $b_j^*\neq 0$
- $b_{j-1} \in \text{span}(b_1,\ldots,b_{j-2})=\text{span}(b_1^*,\ldots,b_{j-2}^*)$
- Subtract to b_{j-1} an element of $L(b_1,\ldots,b_{j-2})$ so that b_{j-1} is in a box: $b_{j-1}=y_1 b_1^*+\ldots+y_{j-2}b_{j-2}^*$ where $|y_i| \leq 1/2$. Then $||b_{j-1}|| \leq (\sqrt{d}/2) \max_i ||b_i^*||$.
- Similarly, we can make sure that $||b_j|| \leq (\sqrt{d}/2) \max_i ||b_i^*||$.
- This is similar to lifting.

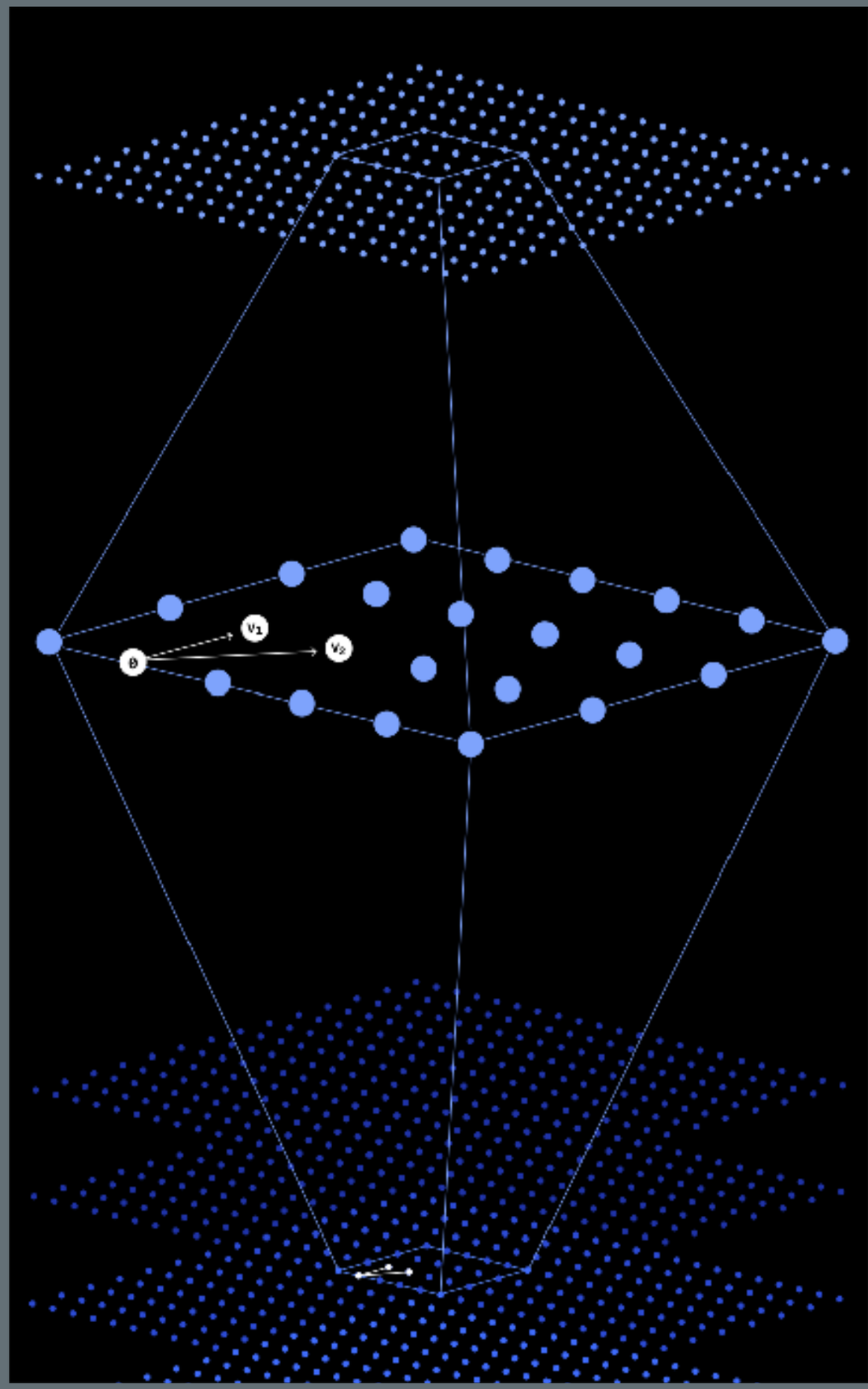
RECAP

- Let π be the projection over $\text{span}(b_1,\ldots b_{j-2})^{\perp}$.
- Compute coprime p and q s.t. $\pi(b_j)=(p/q)b_{j-1}^*$.
- Find $(u,v)\in\mathbf{Z}^2$ s.t. $up+vq=1$ using XGCD(p,q).
- Replace (b_{j-1},b_j) by $(pb_{j-1}-qb_j, vb_{j-1}+ub_j)$:
- Size-reduce b_{j-1} and b_j: make $\|\vec{b}_{j-1}\|$ and $\|\vec{b}_j\| \leq \sqrt{d}\max_i\|\vec{b}_i^*\|$ using $L(b_1,\ldots,b_{j-2})$.
- The BASIS algorithm runs in polynomial time and outputs a basis of the lattice L which is ``short'':
  - If B_0 is the input generator, then the output B satisfies: $||B|| \leq \sqrt{\dim(L)}\,||B_0||$ where $||B|| = \max_i ||b_i||$

TAKE AWAY

- Key ideas:
  - ➤ Projecting to simplify the problem.
  - ➤ Lifting to make sure that vectors do not get too big.

LATTICE PROBLEMS

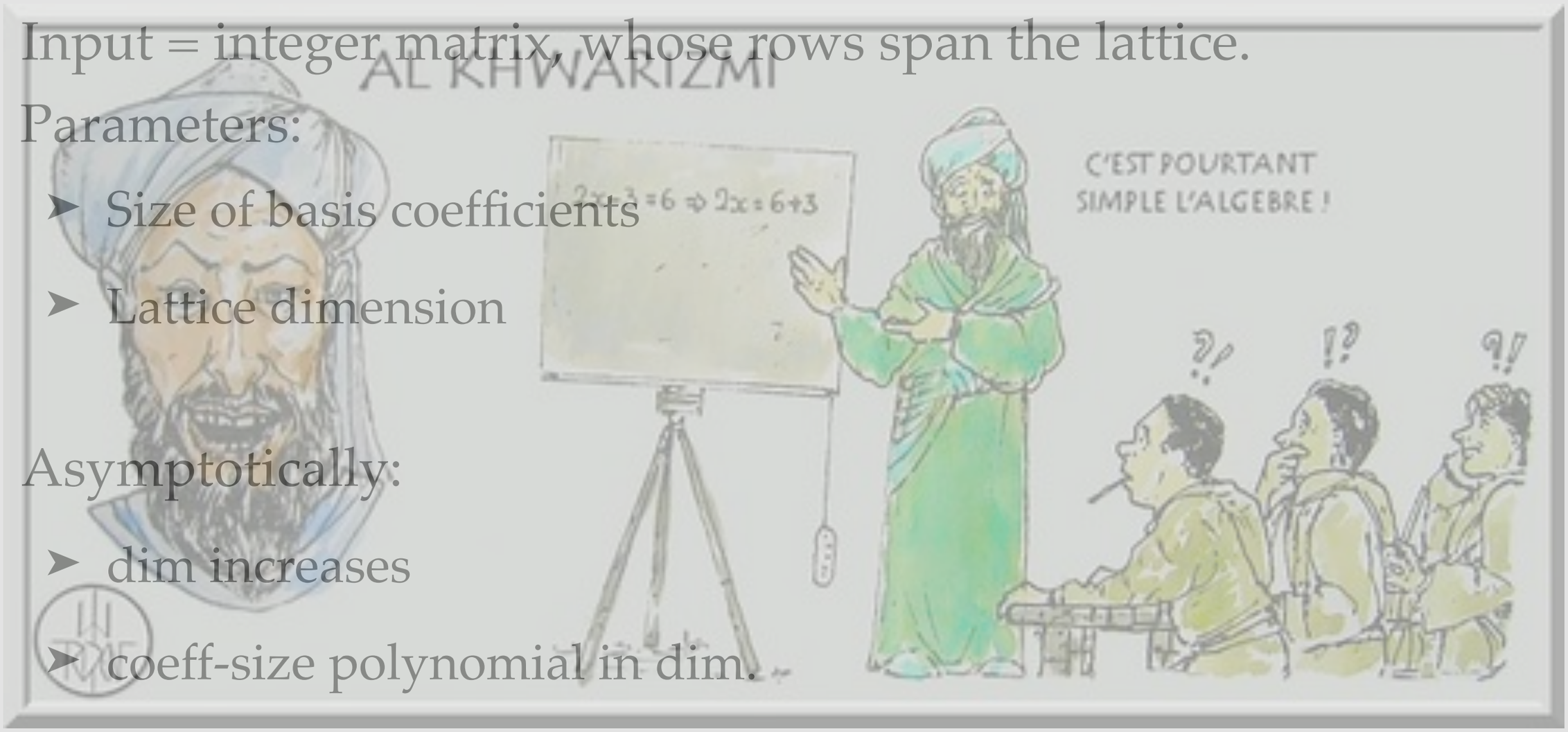


LATTICE ALGORITHMS

- Input = integer matrix, whose rows span the lattice.

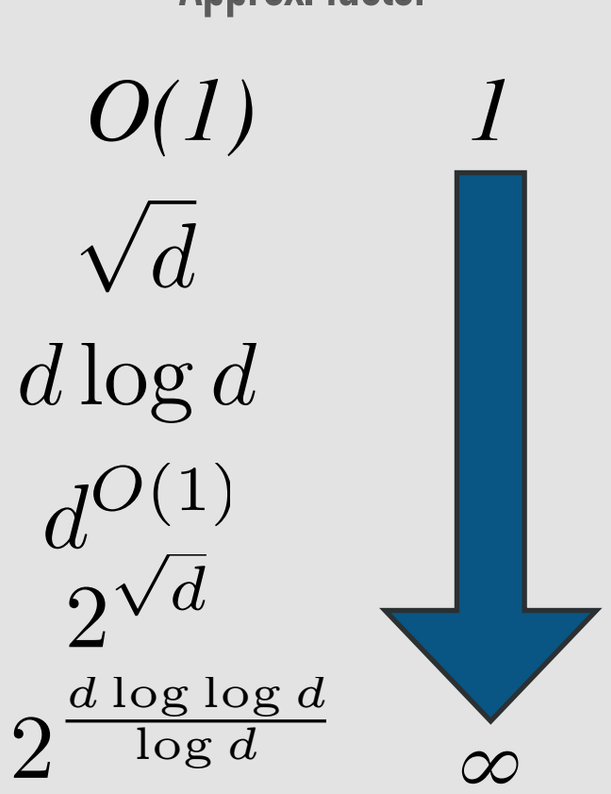
  Parameters:
  - Size of basis coefficients
  - Lattice dimension

- Asymptotically:
  - dim increases
  - coeff-size polynomial in dim.



HARD LATTICE PROBLEMS

- Since 1996, lattices are very trendy in classical and quantum complexity theory.
- Depending on the dimension d:
  - NP-hardness
  - non NP-hardness (NP∩co-NP)
  - worst-case / average-case reduction
  - cryptography
  - subexp-time algorithms
  - poly-time algorithms



HARD LATTICE PROBLEMS

- Input: a lattice L and an n-dim ball C.
- Output: decide if L∩C is non-trivial, and find a point when applicable. Easy if $L=\mathbf{Z}^n$.
- Two settings
  - ➤ Approx: L∩C has many points.

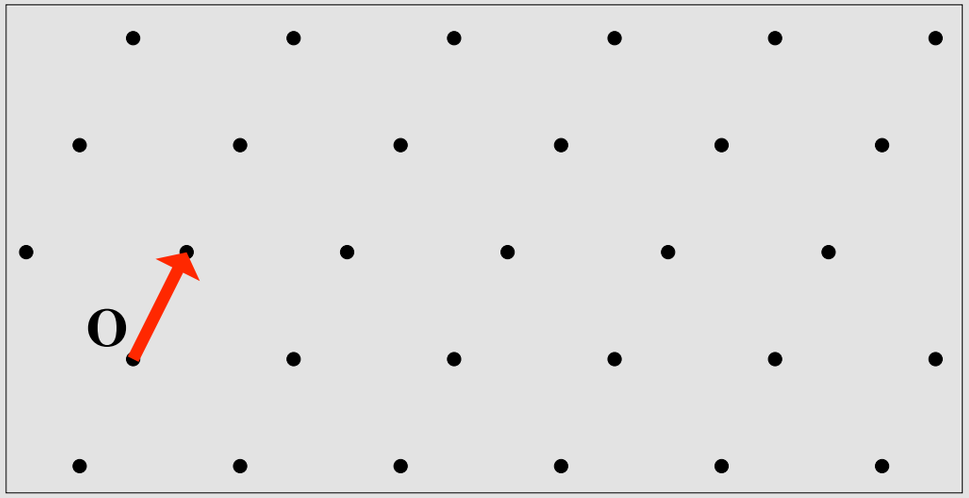
    Ex: SIS and ISIS.
  - ➤ Unique: only one non-trivial point.

    Ex: BDD.

THE SHORTEST VECTOR PROBLEM

THE SHORTEST VECTOR PROBLEM (SVP)

- Input: a basis of a d-dim lattice L.
- Output: nonzero $v \in L$ minimizing $||v||$ i.e. $||v|| = \lambda_1(L)$



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

THE SHORTEST VECTOR PROBLEM (SVP): DECISIONAL VARIANT

- Input: a basis of a d-dim lattice L and a rational number r
- Output: is there $v \in L$ such that $||v|| \leq r$?

EXERCISE

- Show that given an oracle for Decisional-SVP, one can solve SVP in polynomial in time.
- Lagrange's algorithm shows how to solve rank-2 SVP in polynomial time.