Certified lattice reduction

Toward a sound algorithmic Minkowsky theory

Thomas Espitau

March 16, 2018

Sorbonne Université, LIP6, Paris

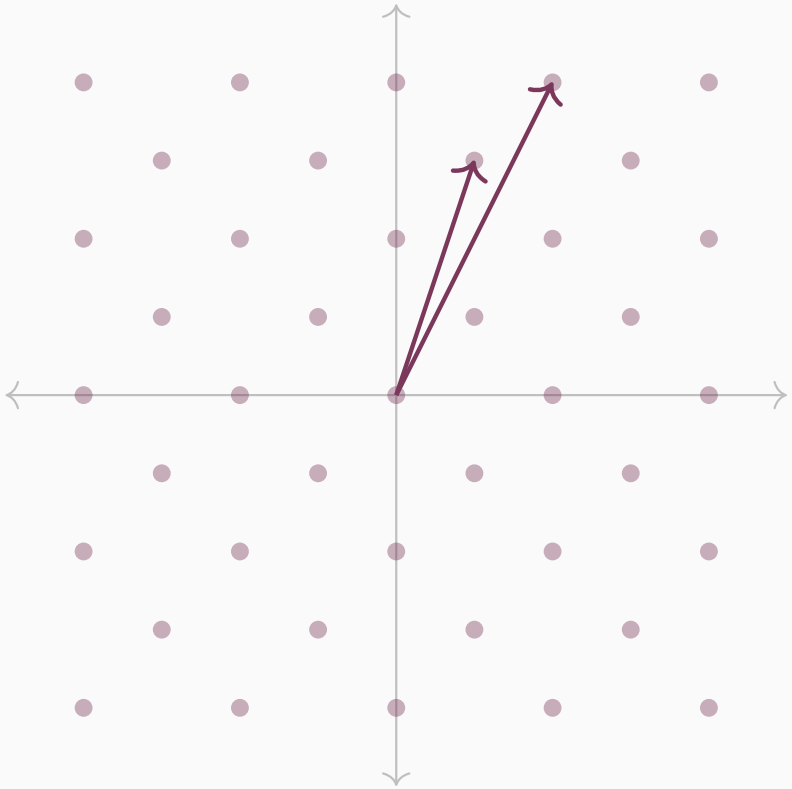
Introduction

What is this all about?

This talk is all about Λ, a free $\mathbf{Z}$-module of finite rank, endowed with a positive-definite quadratic form on its ambient space $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$.

What is this all about?

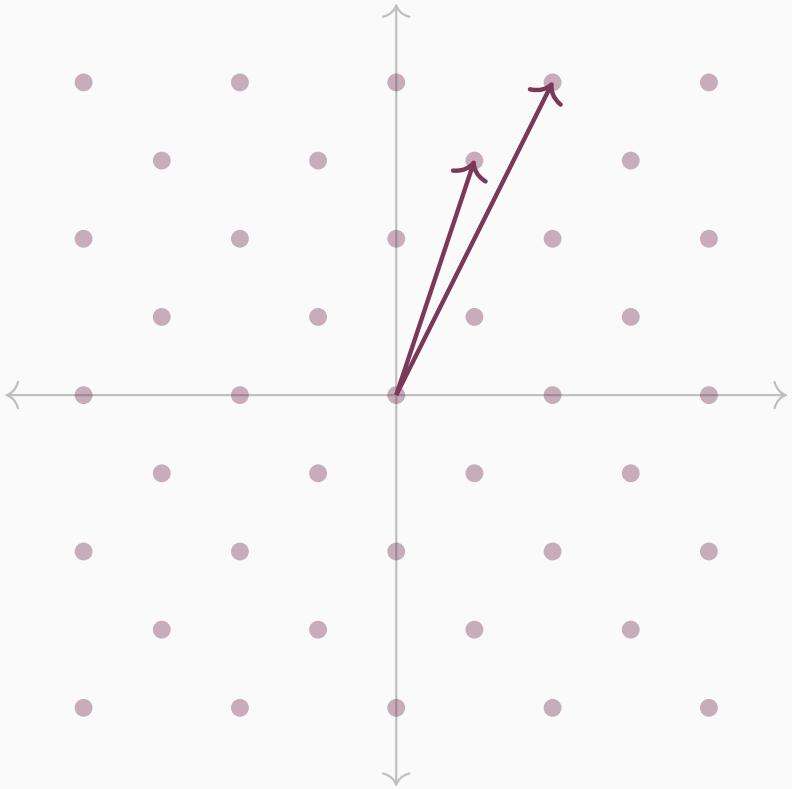
This talk is all about Λ, a free $\mathbf{Z}$-module of finite rank, endowed with a positive-definite quadratic form on its ambient space $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$.



What is this all about?

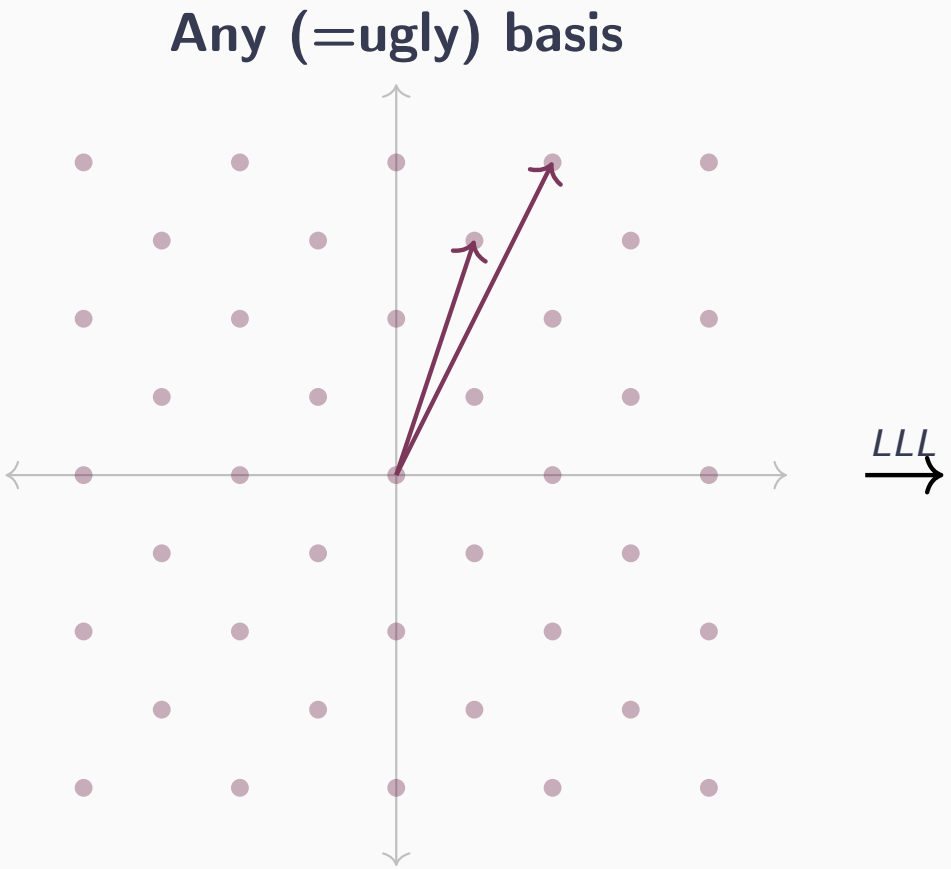
This talk is all about Λ, a free $\mathbf{Z}$-module of finite rank, endowed with a positive-definite quadratic form on its ambient space $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$.

Any (=ugly) basis



What is this all about?

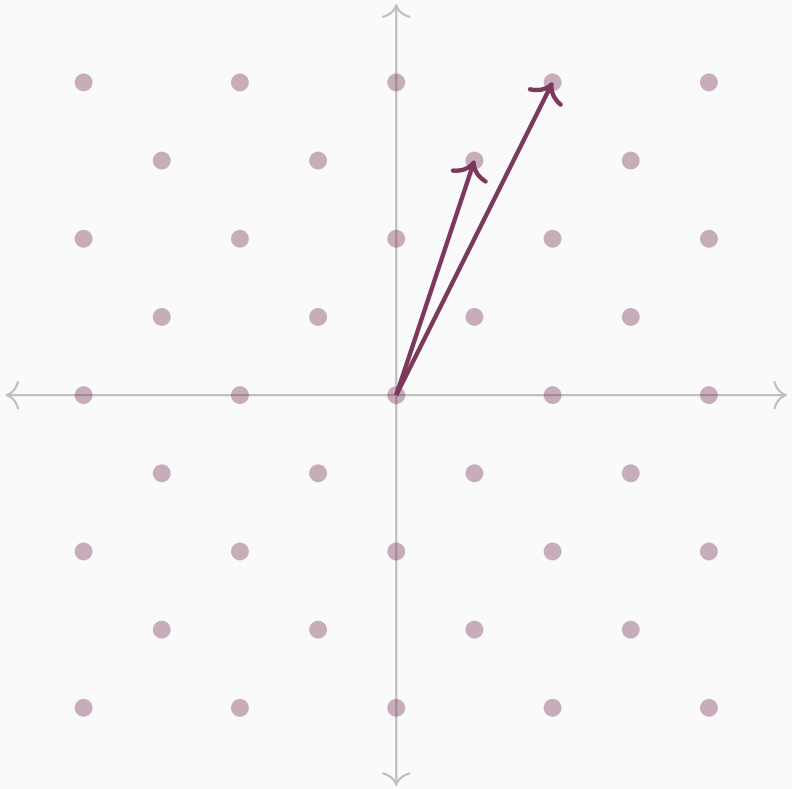
This talk is all about Λ, a free $\mathbf{Z}$-module of finite rank, endowed with a positive-definite quadratic form on its ambient space $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$.



What is this all about?

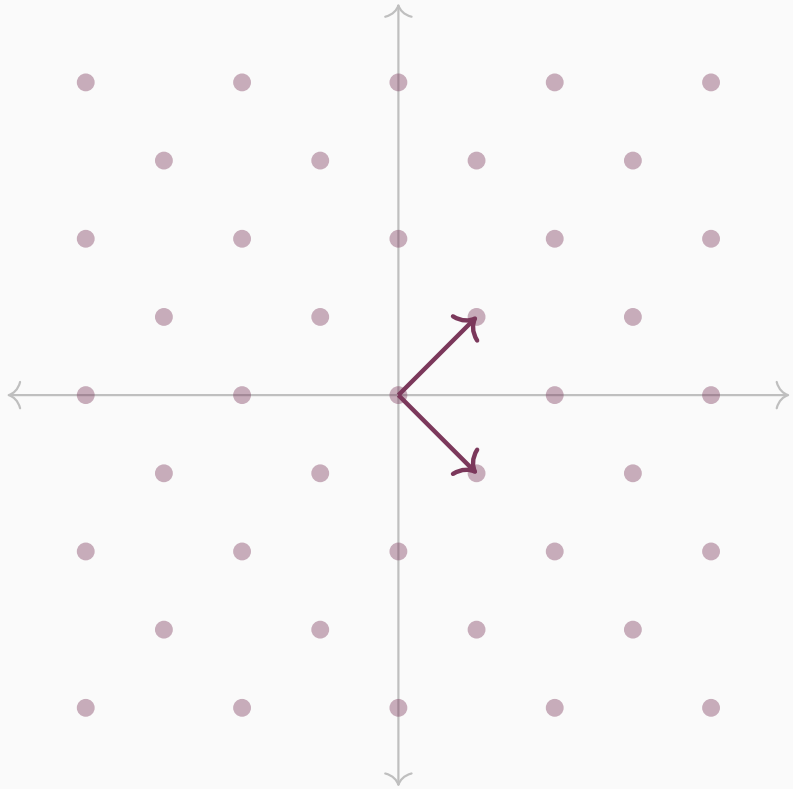
This talk is all about Λ, a free $\mathbf{Z}$-module of finite rank, endowed with a positive-definite quadratic form on its ambient space $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$.

Any (=ugly) basis



$\xrightarrow{LLL}$

Reduced (=nice) basis



Why is it useful?

- Break some crypto (RSA in specific setting, knapsacks...)
- Fast computation of normal forms for integral matrices (Smith, Hermite)
- Factor integer polynomials
- Find small integral relations
- Useful in algorithmic number theory (computation of class group, principal ideal problem, ...)

Motto: Find (somewhat) small elements in a discrete Euclidean structure.

Reduction theory

Some recalls on quadratic algebra

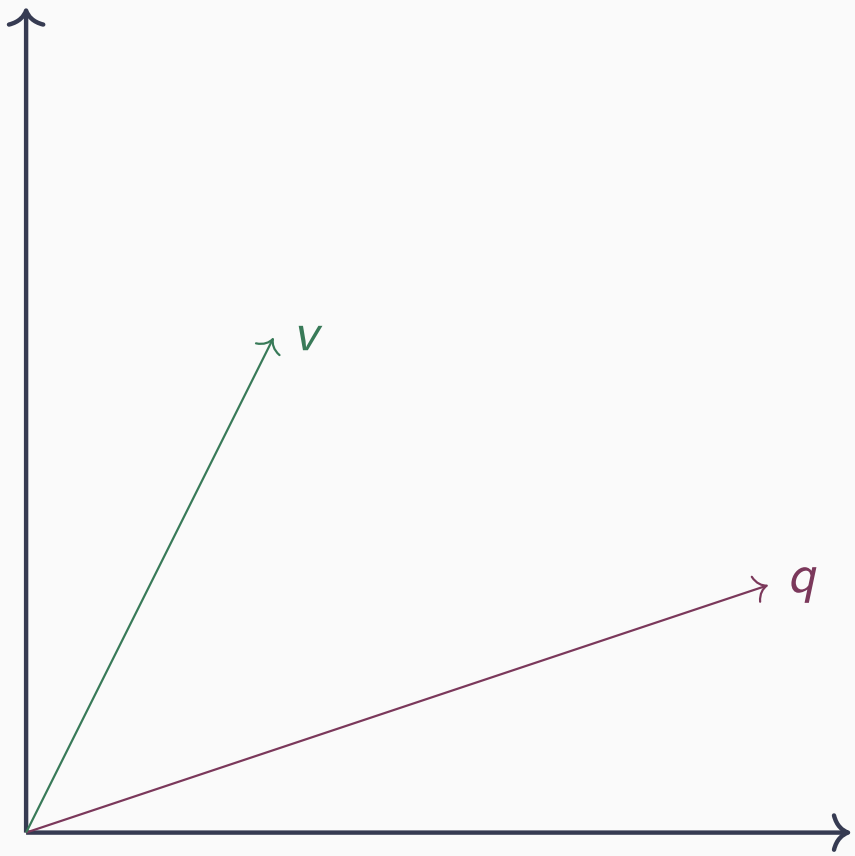
- Euclidean space $(E, \langle \cdot, \cdot \rangle)$. Consider $S = (v_1, \ldots, v_d)$ a basis of E.
- The flag $\mathcal{F}_S$ associated to S is the data of the finite increasing chain of subspaces:

$$\mathbf{R}v_1 \subset \mathbf{R}v_1 \oplus \mathbf{R}v_2 \subset \cdots \subset \bigoplus_{i=1}^{r} v_i\mathbf{R}.$$

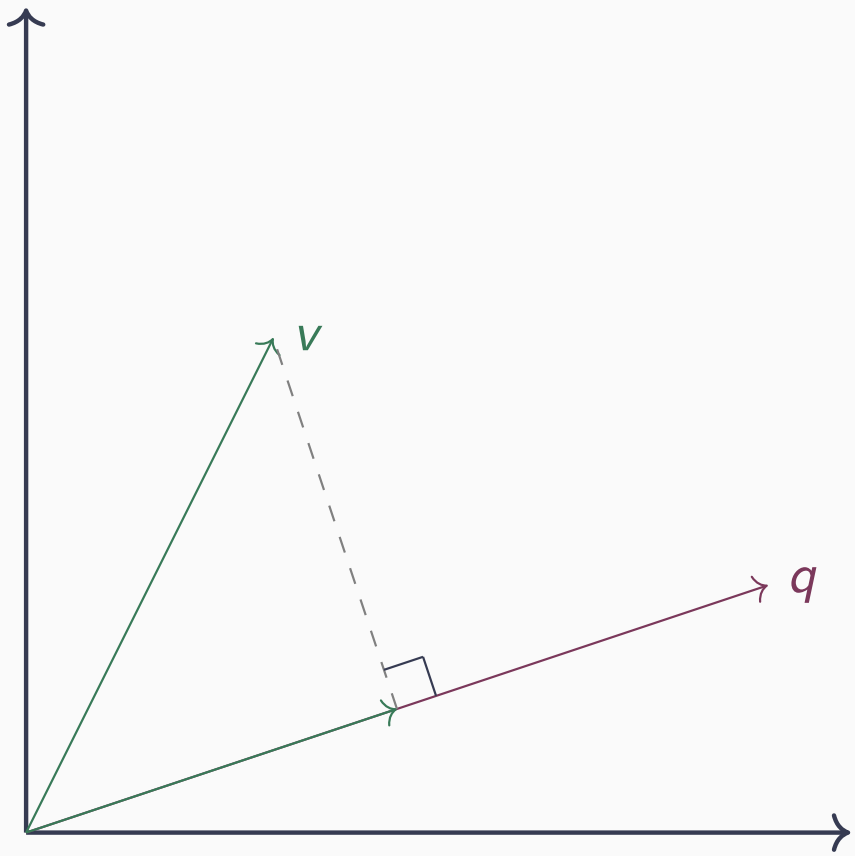
- Orthogonal complement: $A^\perp = \{x \in E \mid \forall a \in A,\ \langle x, a \rangle = 0\}$.
- π_i the orthogonal projection on $(v_1, \ldots, v_{i-1})^\perp$.

Goal: Orthogonalizing S while preserving its flag,

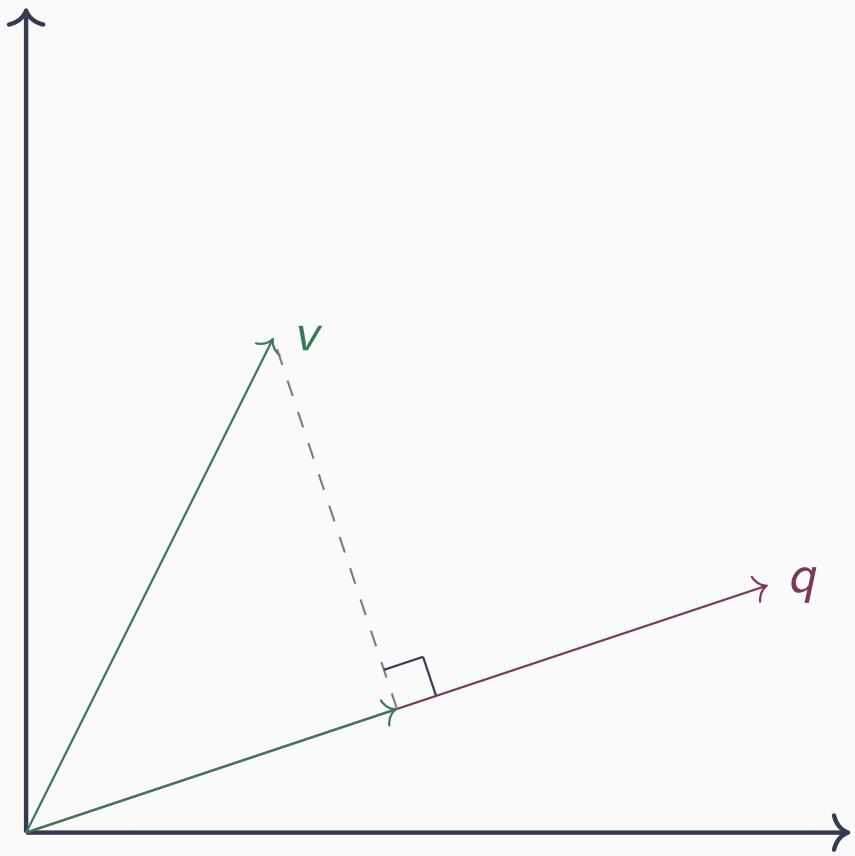
A picture is worth a thousand words



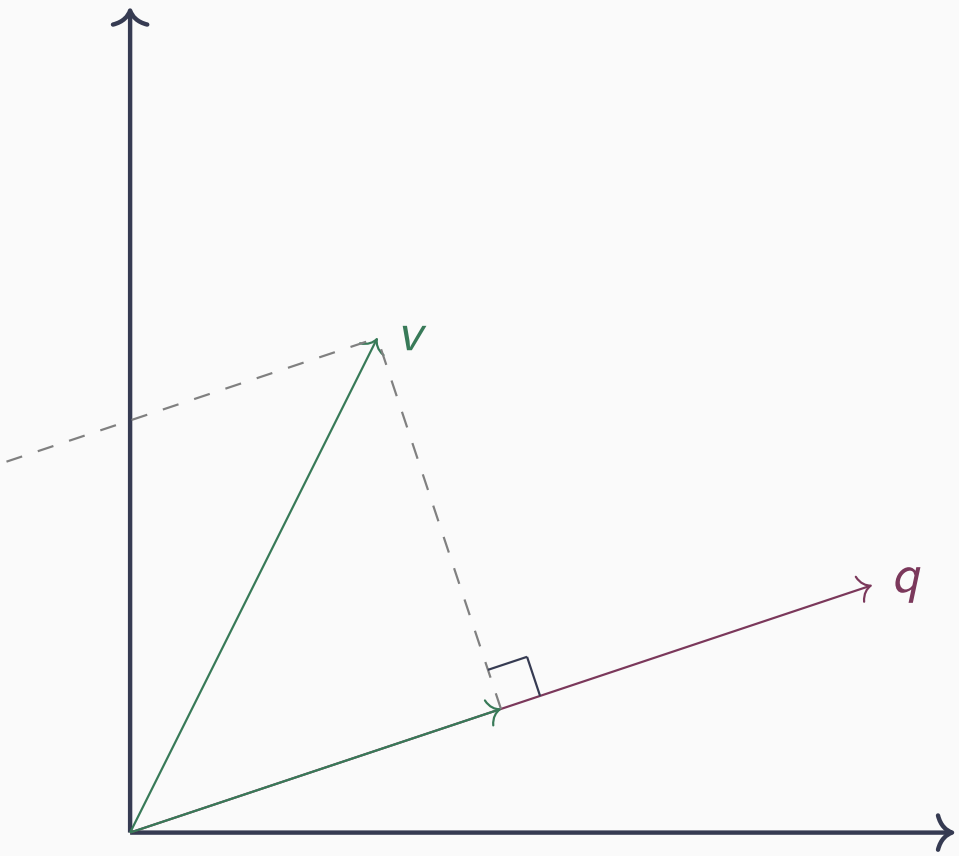
A picture is worth a thousand words



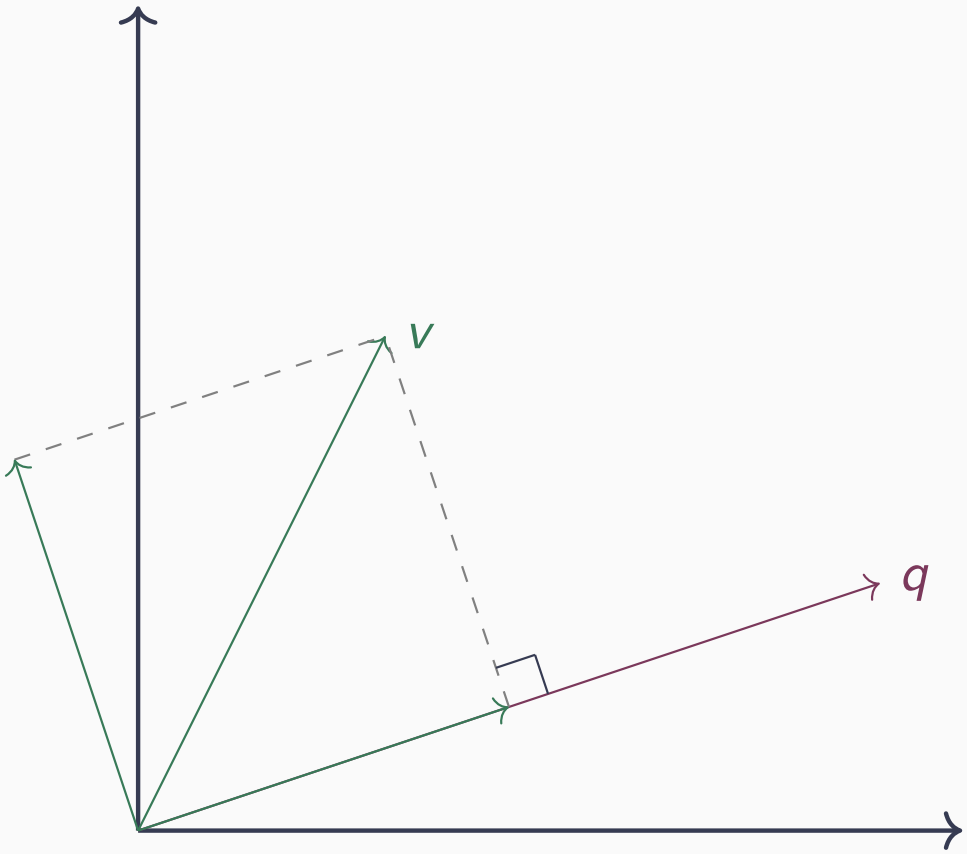
A picture is worth a thousand words



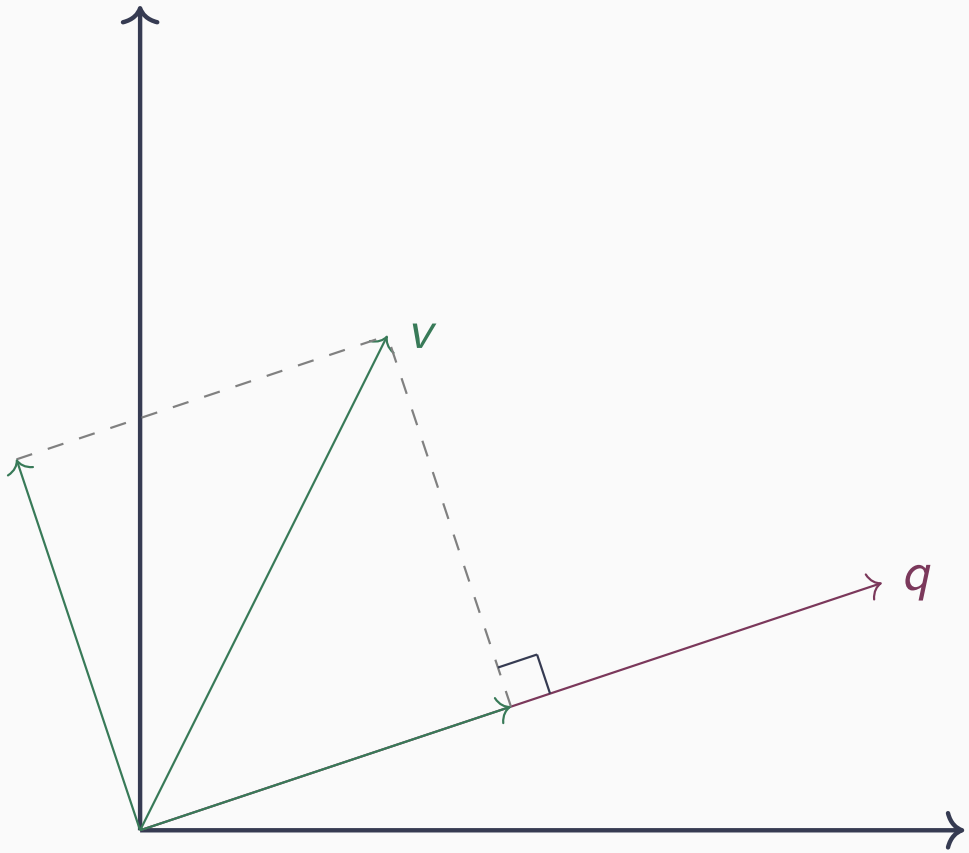
A picture is worth a thousand words



A picture is worth a thousand words



A picture is worth a thousand words



$$\pi_{q^T}(v) = v - \pi_q(v)$$
$$= v - \frac{\langle v, q \rangle}{\|q\|^2} q$$

Gram-Schmidt Orthogonalization

Goal: Orthogonalizing S while preserving its flag,

Iterative solution: GSO

$$\pi_1(v_1) = v_1$$

$$\forall 1 < i \leq r, \quad \pi_i(v_i) = v_i - \sum_{j=1}^{i-1} \frac{\langle v_i, \pi_j(v_j) \rangle}{\langle \pi_j(v_j), \pi_j(v_j) \rangle}.$$

Covolume

Covolume

(co)volume $\mathrm{covol}(S)$ of a family of vector $S = (b_1, \cdots, b_r)$:

$$\mathrm{covol}(S) = \sqrt{\det \langle b_i, b_j \rangle} = \prod_{i=1}^{r} \|\pi_i(v_i)\|$$

Corresponds to the volume of the fundamental domain $\{\sum x_i b_i | 0 \leq x_i \leq 1\}$ (or equivalently the norm of $\|b_1 \wedge \ldots \wedge b_d\|$...)

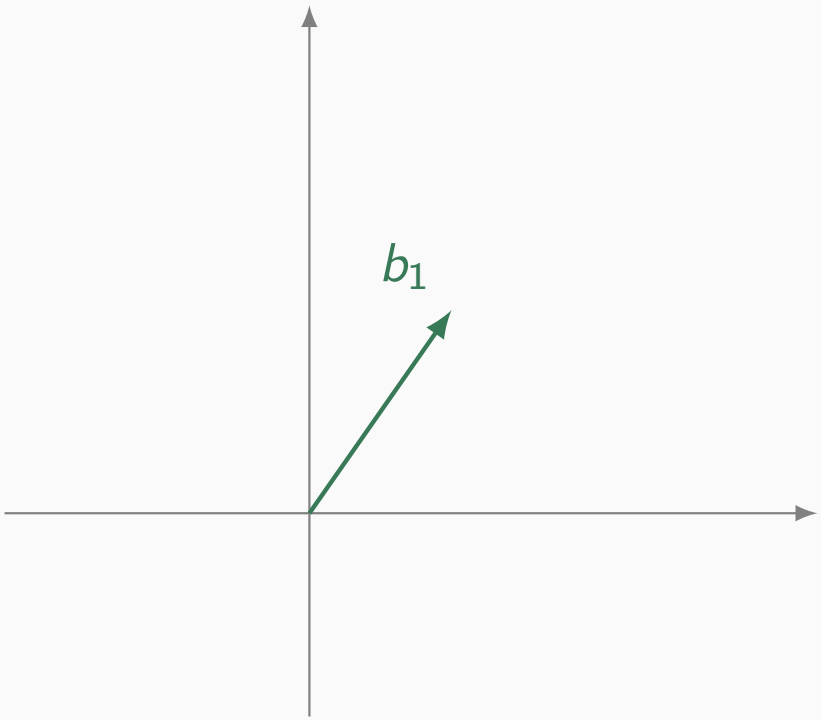
Lattices

Lattice

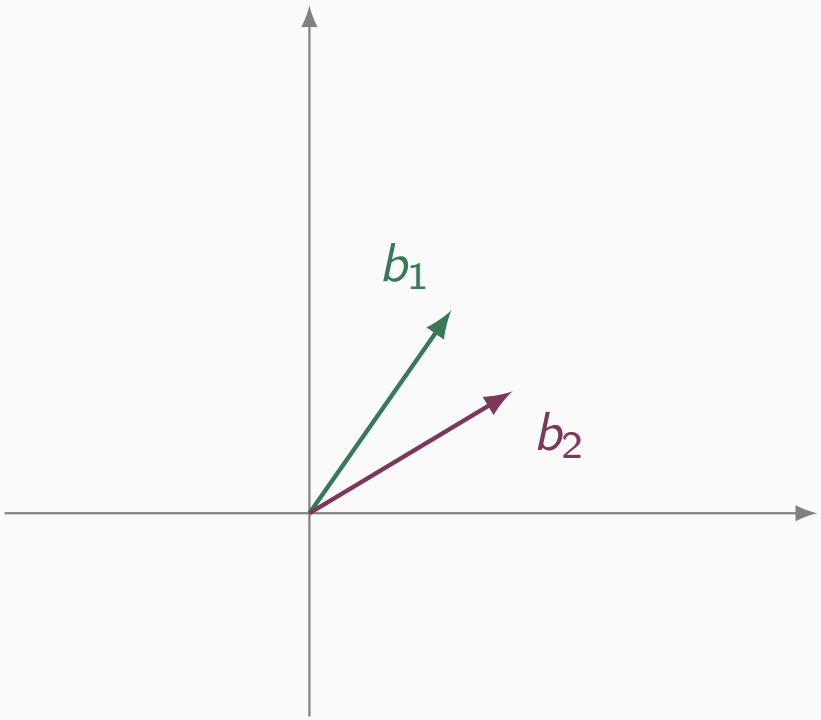
A (real) lattice Λ is a finitely generated free $\mathbf{Z}$-module, endowed with a positive-definite quadratic form $\langle \cdot, \cdot \rangle$ on its ambient space $\Lambda \otimes_{\mathbf{Z}} \mathbf{Q}$, making Λ discrete for the induced norm.

There exists a finite family $b_1, \ldots, b_d \in \Lambda^d$ such that $\Lambda = \bigoplus_{i=1}^{d} b_i \mathbf{Z}$: a basis of Λ.

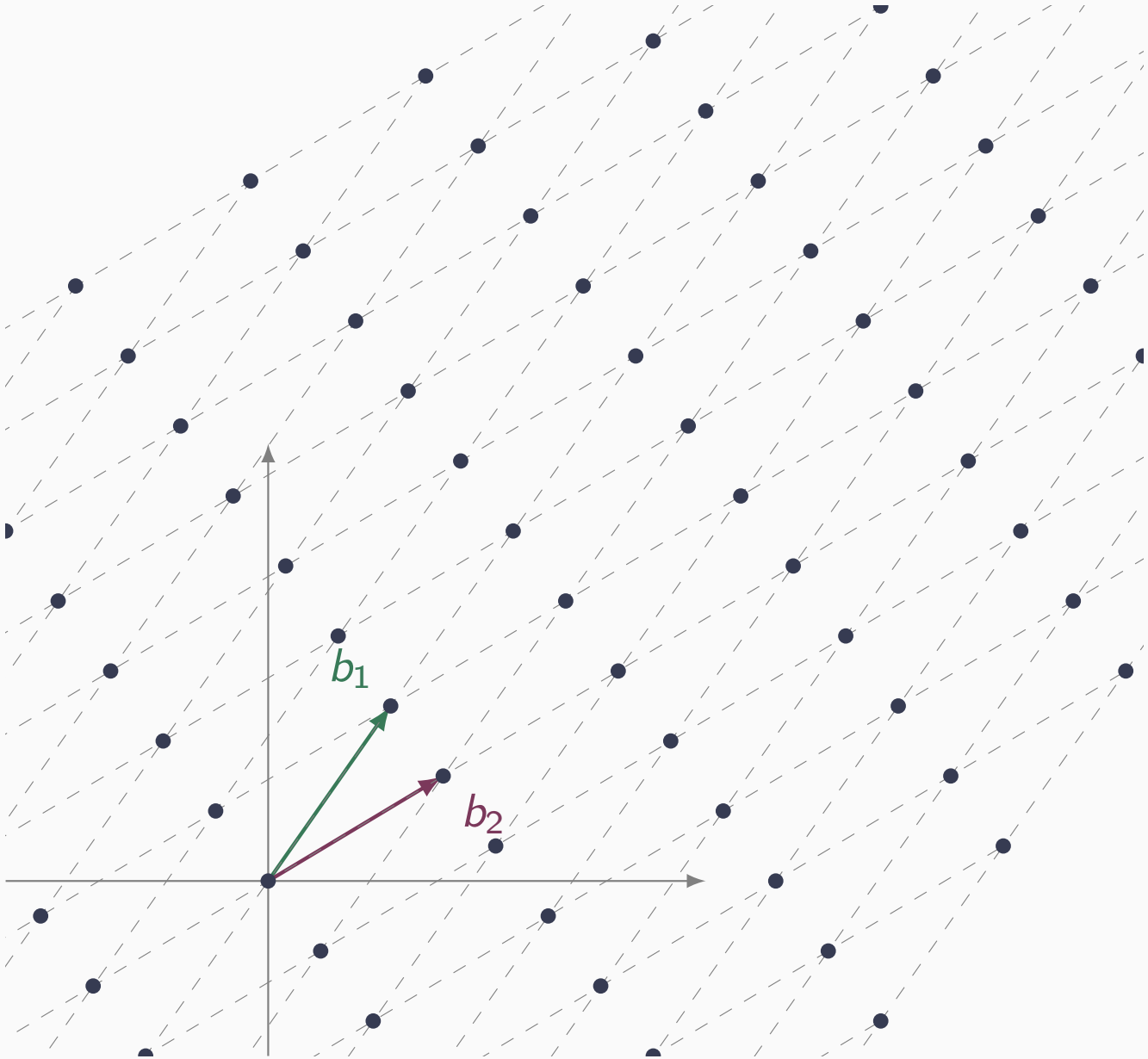
A lattice and two basis



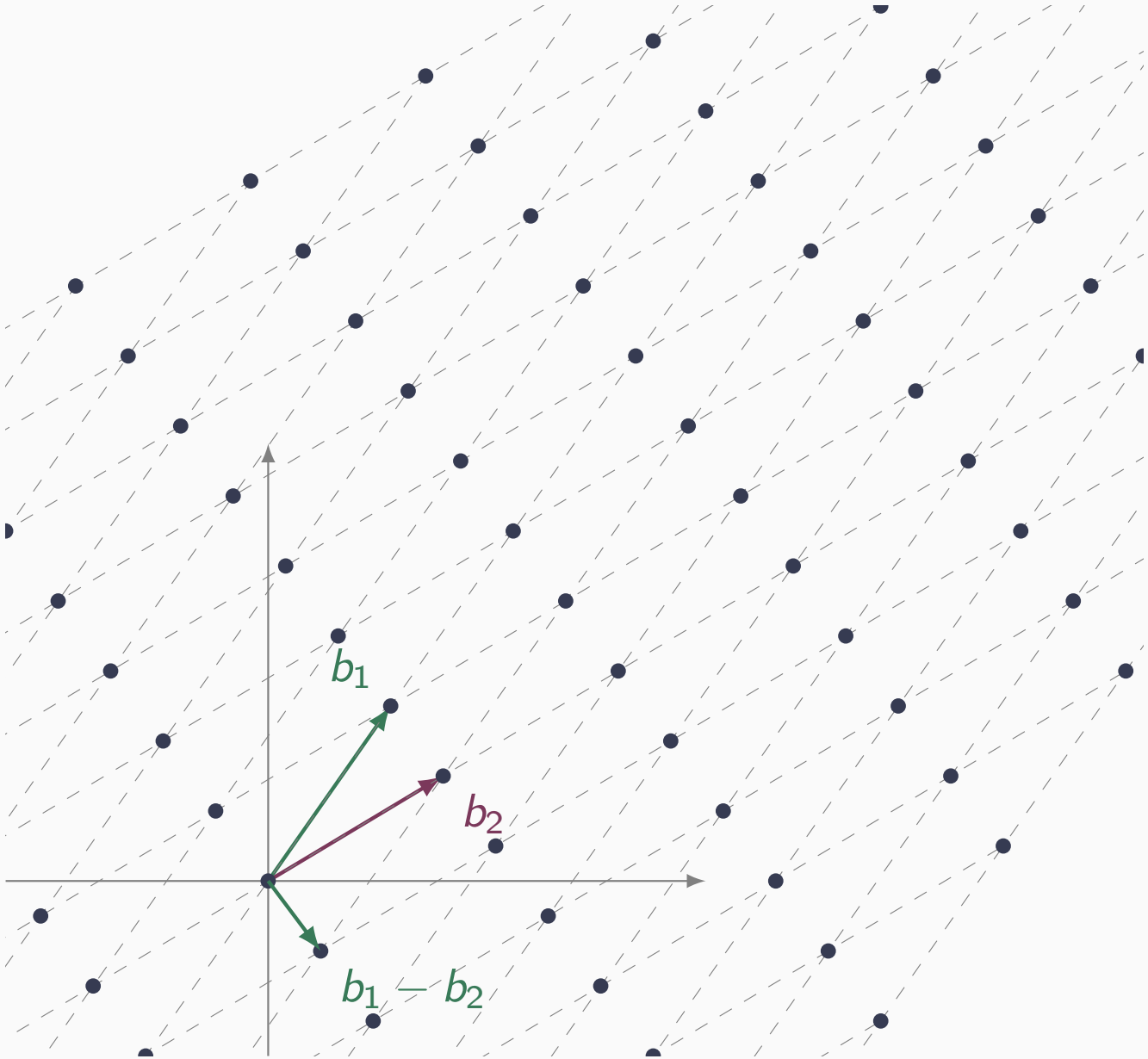
A lattice and two basis



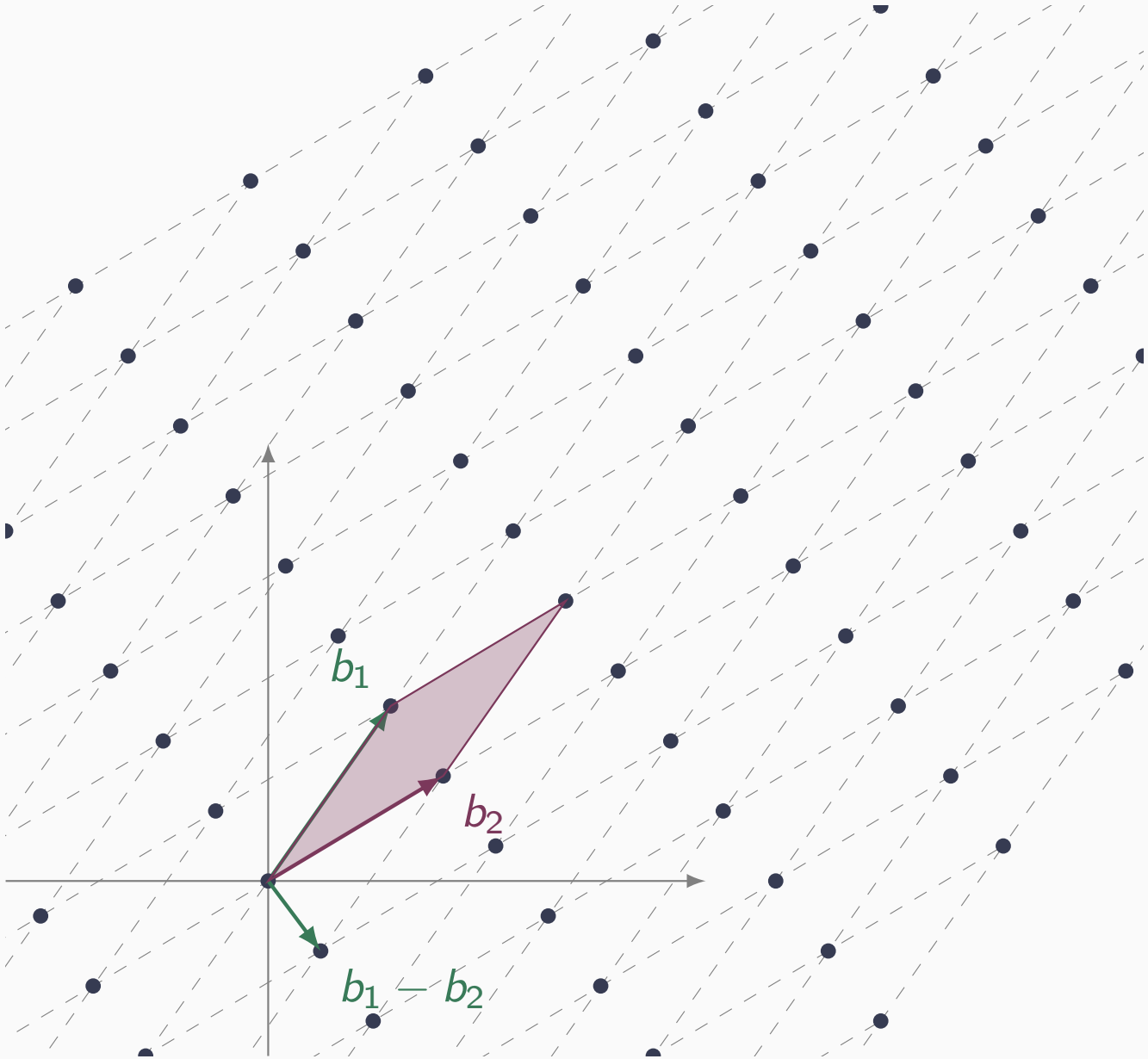
A lattice and two basis



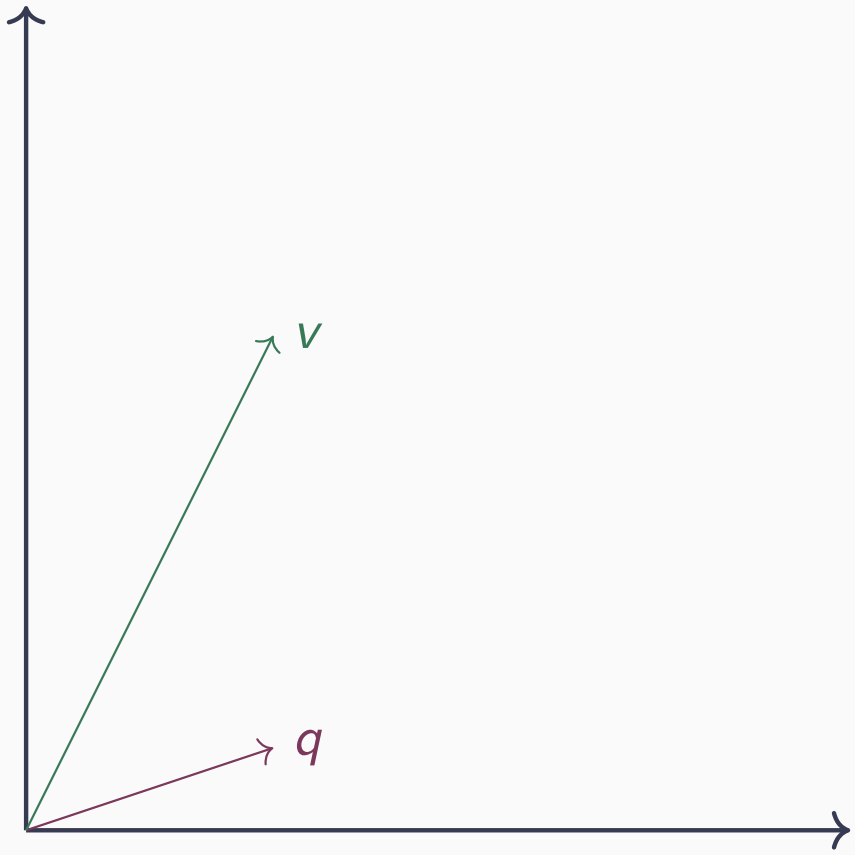
A lattice and two basis



A lattice and two basis

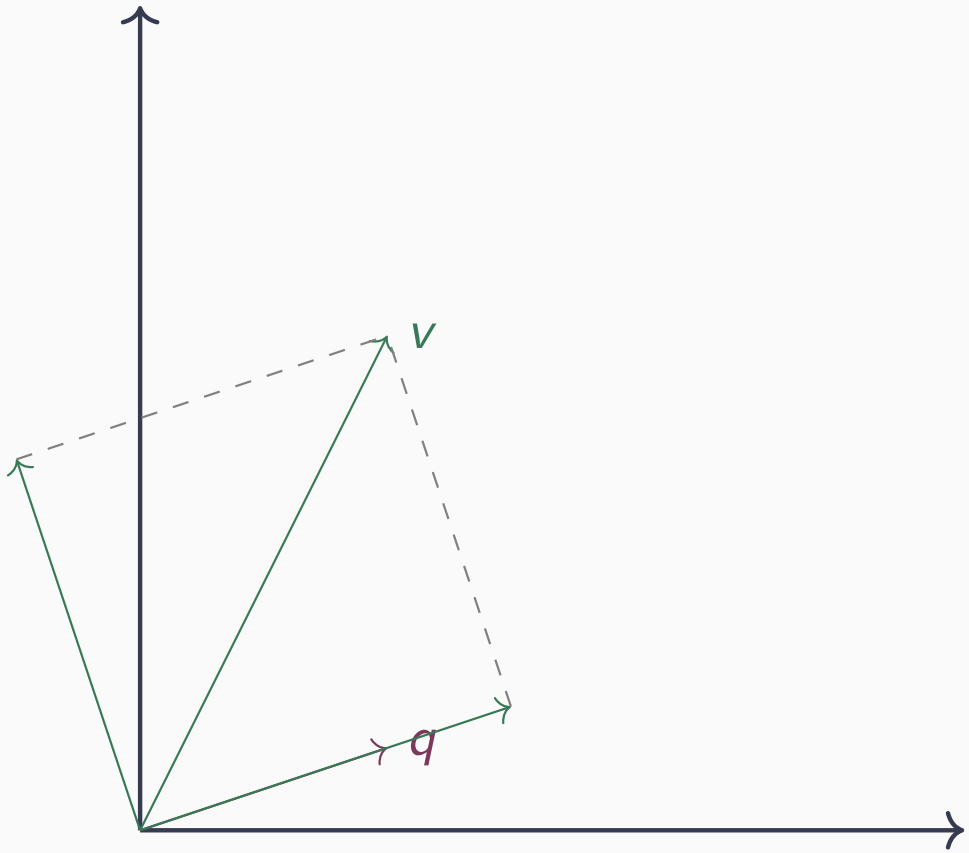


Is there some generic transformation to reduce a vector?



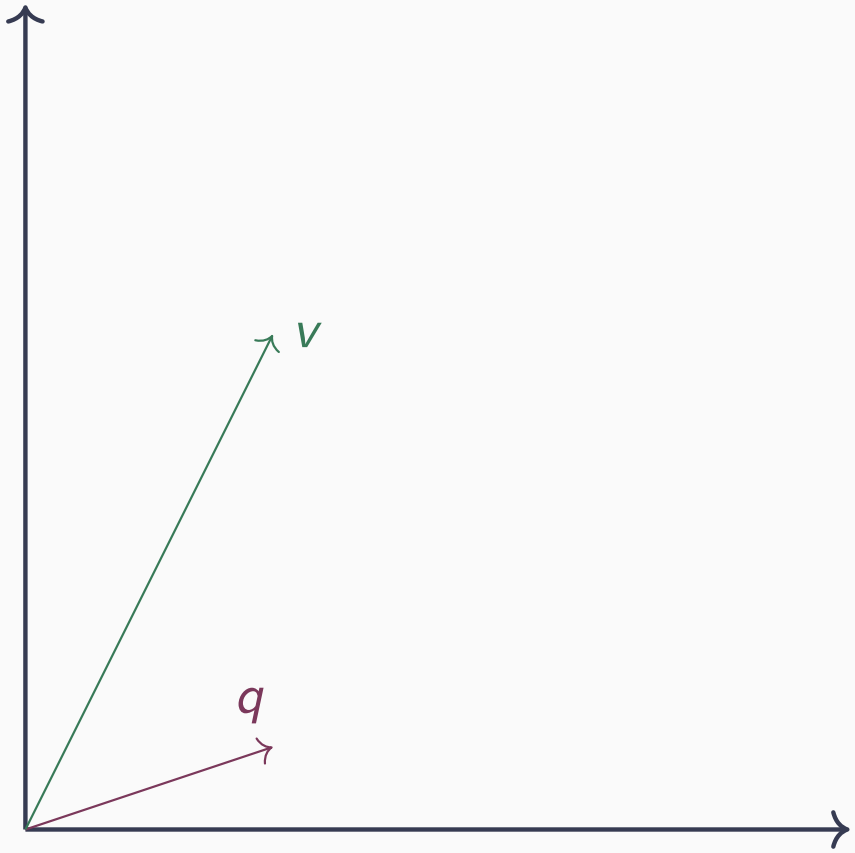
$$\pi_{q^T}(v) = v - \pi_q(v)$$
$$= v - \frac{\langle v, q \rangle}{\|q\|^2} q$$

Is there some generic transformation to reduce a vector?



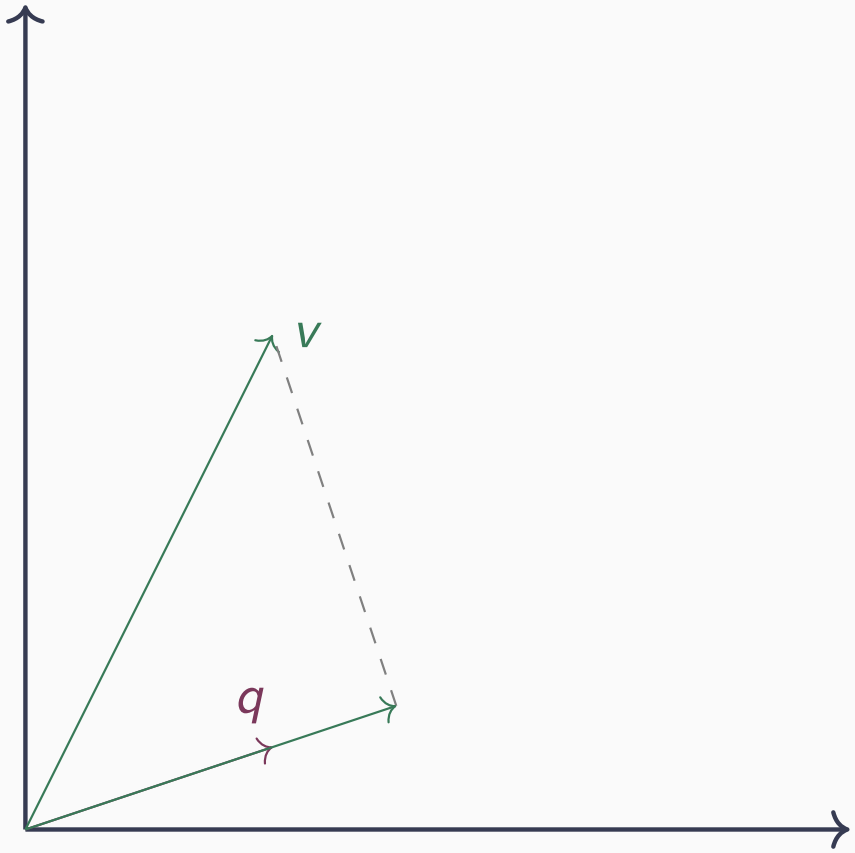
$$\pi_{q^T}(v) = v - \pi_q(v)$$
$$= v - \frac{\langle v, q \rangle}{\|q\|^2} q$$

Is there some generic transformation to reduce a vector?



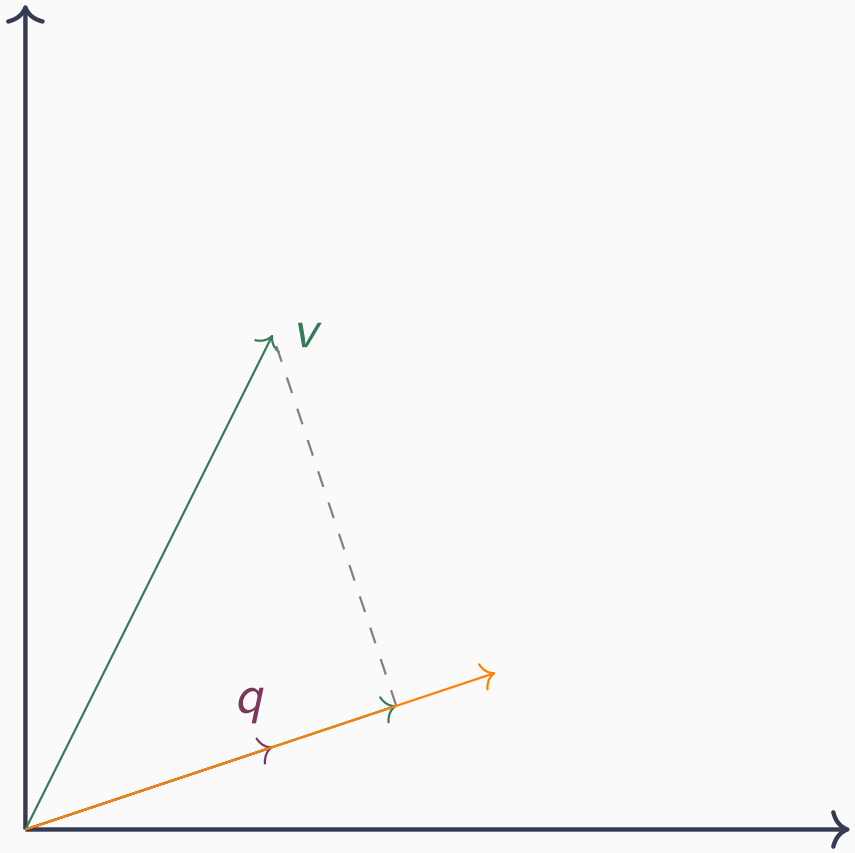
$$\overline{\pi_{q^T}}(v) = v - \overline{\pi_q}(v)$$
$$= v - \left\lfloor \frac{\langle v, q \rangle}{\|q\|^2} \right\rceil q$$

Is there some generic transformation to reduce a vector?



$$\overline{\pi_{q^T}}(v) = v - \overline{\pi_q}(v)$$
$$= v - \left\lfloor \frac{\langle v, q \rangle}{\|q\|^2} \right\rceil q$$

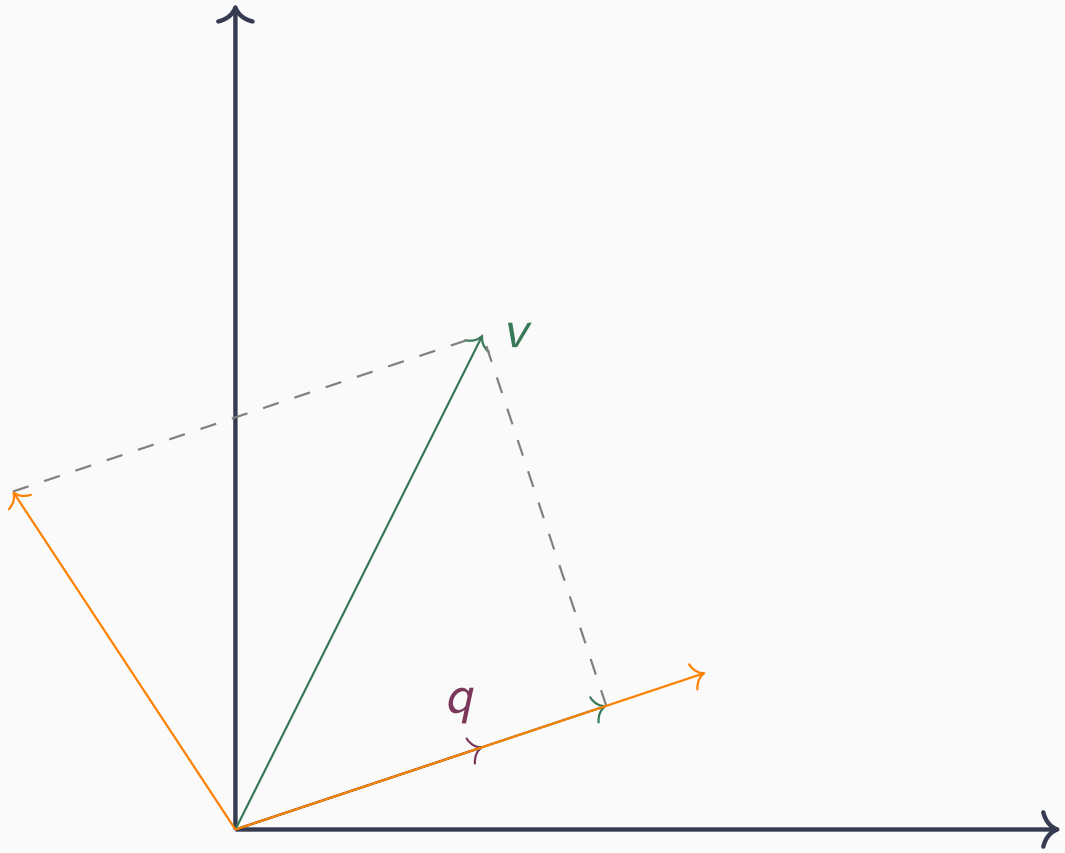
Is there some generic transformation to reduce a vector?



$$\overline{\pi_{q^T}}(v) = v - \overline{\pi_q}(v)$$

$$= v - \left\lfloor \frac{\langle v, q \rangle}{\|q\|^2} \right\rceil q$$

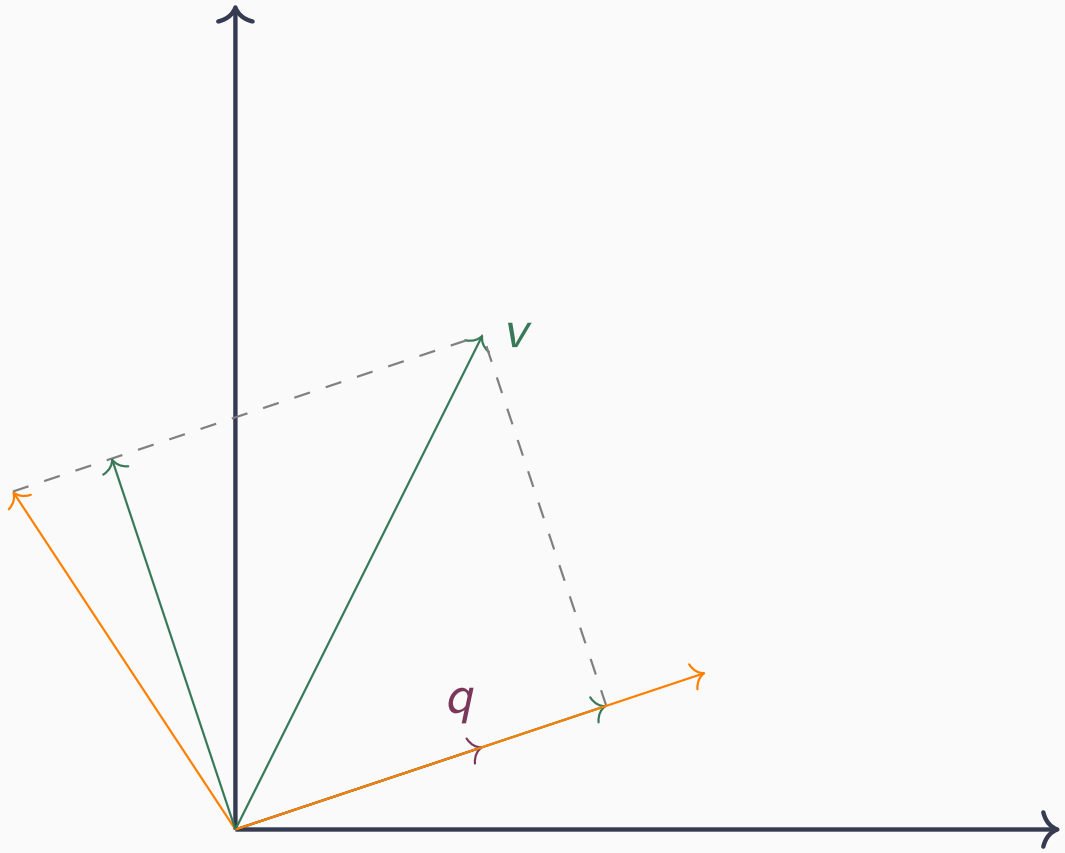
Is there some generic transformation to reduce a vector?



$$\overline{\pi_{q^T}}(v) = v - \overline{\pi_q}(v)$$

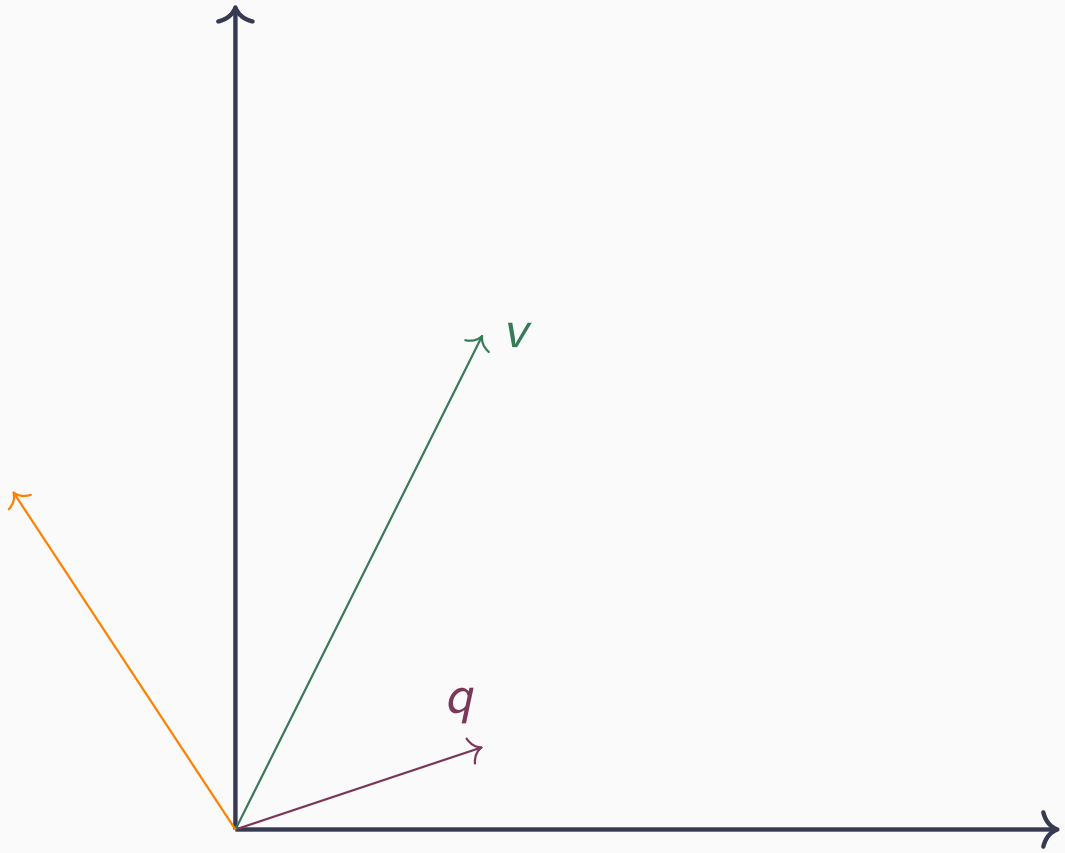
$$= v - \left\lfloor \frac{\langle v, q \rangle}{\|q\|^2} \right\rceil q$$

Is there some generic transformation to reduce a vector?



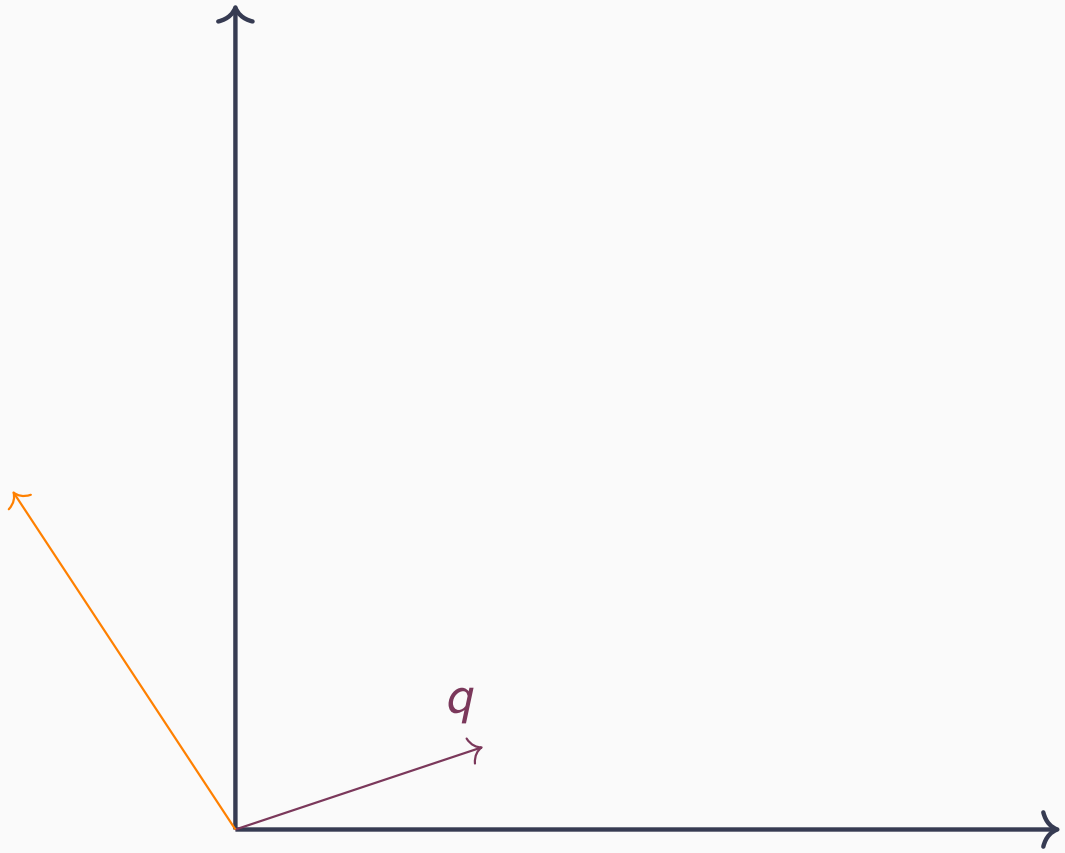
$$\overline{\pi_{q^T}}(v) = v - \overline{\pi_q}(v)$$
$$= v - \left\lfloor \frac{\langle v, q \rangle}{\|q\|^2} \right\rceil q$$

Is there some generic transformation to reduce a vector?



$$\overline{\pi_{q^T}}(v) = v - \overline{\pi_q}(v)$$
$$= v - \left\lfloor \frac{\langle v, q \rangle}{\|q\|^2} \right\rceil q$$

Is there some generic transformation to reduce a vector?



$$\overline{\pi_{q^T}}(v) = v - \overline{\pi_q}(v)$$
$$= v - \left\lfloor \frac{\langle v, q \rangle}{\|q\|^2} \right\rceil q$$

In dim 2... The Gauss reduction algorithm

Algorithm 1: The original Gauss algorithm.

Input: Initial basis (b_1, b_2)

Result: A reduced basis

// Exchange step

1 **if** $\|b_1\| > \|b_1\|$ **then**

  // Swap b_1 and b_2

2 Call the reduction on (b_2, b_1);

3 **end**

// Size reduction step

4 $b_2 \leftarrow b_2 - \left\lceil \frac{\langle b_1, b_2 \rangle}{\|b_1\|^2} \right\rfloor \cdot b_1$;

5 **if** *Nothing happened* **then**

6 **return** (b_1, b_2)

7 **end**

8 Call the reduction on (b_2, b_1);

Guarantees?

Properties of a Gauss-reduced basis

- $\|b_1\| \leq \|b_2\| \leq \|b_2 + qb_1\|$ for all $q \in \mathbf{Z}$.
- $\|b_1\| \leq \|b_2\| \leq \|b_2 \pm b_1\|$.
- $\|b_1\| \leq \|b_2\|$ and $|\langle b_1, b_2 \rangle| \leq \frac{\|b_1\|^2}{2}$.

Some bases

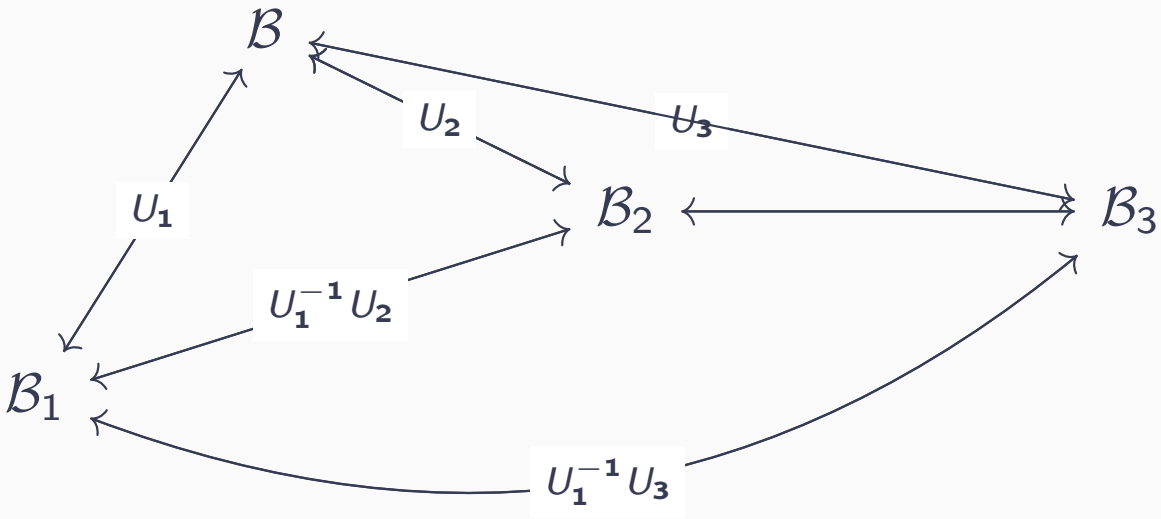
$\mathcal{B}$

$\mathcal{B}_2$

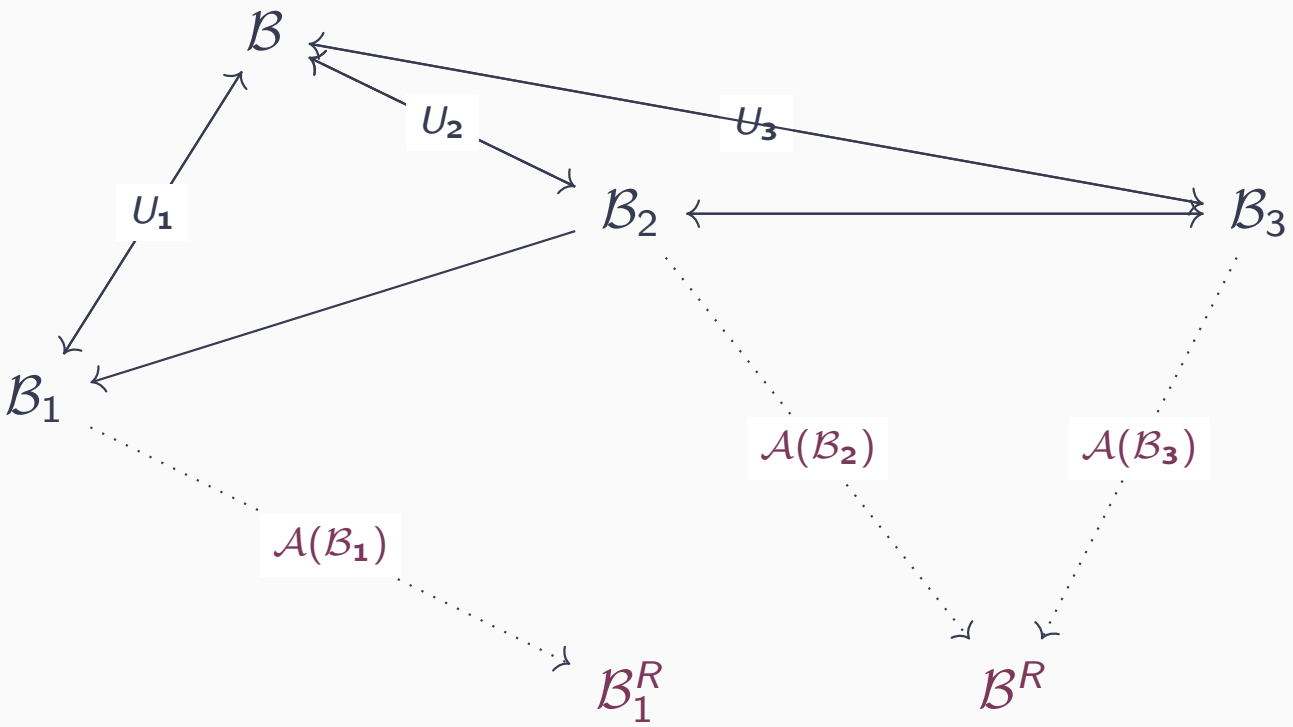
$\mathcal{B}_3$

$\mathcal{B}_1$

Related by unimodular transformations



Algorithm = walk in $\mathrm{Sl}(\mathbb{Z})$



LLL reduced basis

LLL reduction

- Size-Reduction condition

$$\forall i < j, \quad |\langle b_j, \pi_i(b_i)\rangle| \leq \frac{1}{2}\|\pi_i(b_i)\|^2$$

- Lovász condition

$$\forall i, \quad \delta\|\pi_i(b_i)\|^2 \leq \left(\|\pi_{i+1}(b_{i+1})\|^2 + \frac{\langle b_{i+1}, \pi_i(b_i)\rangle}{\|\pi_i(b_i)\|^2}\right)$$

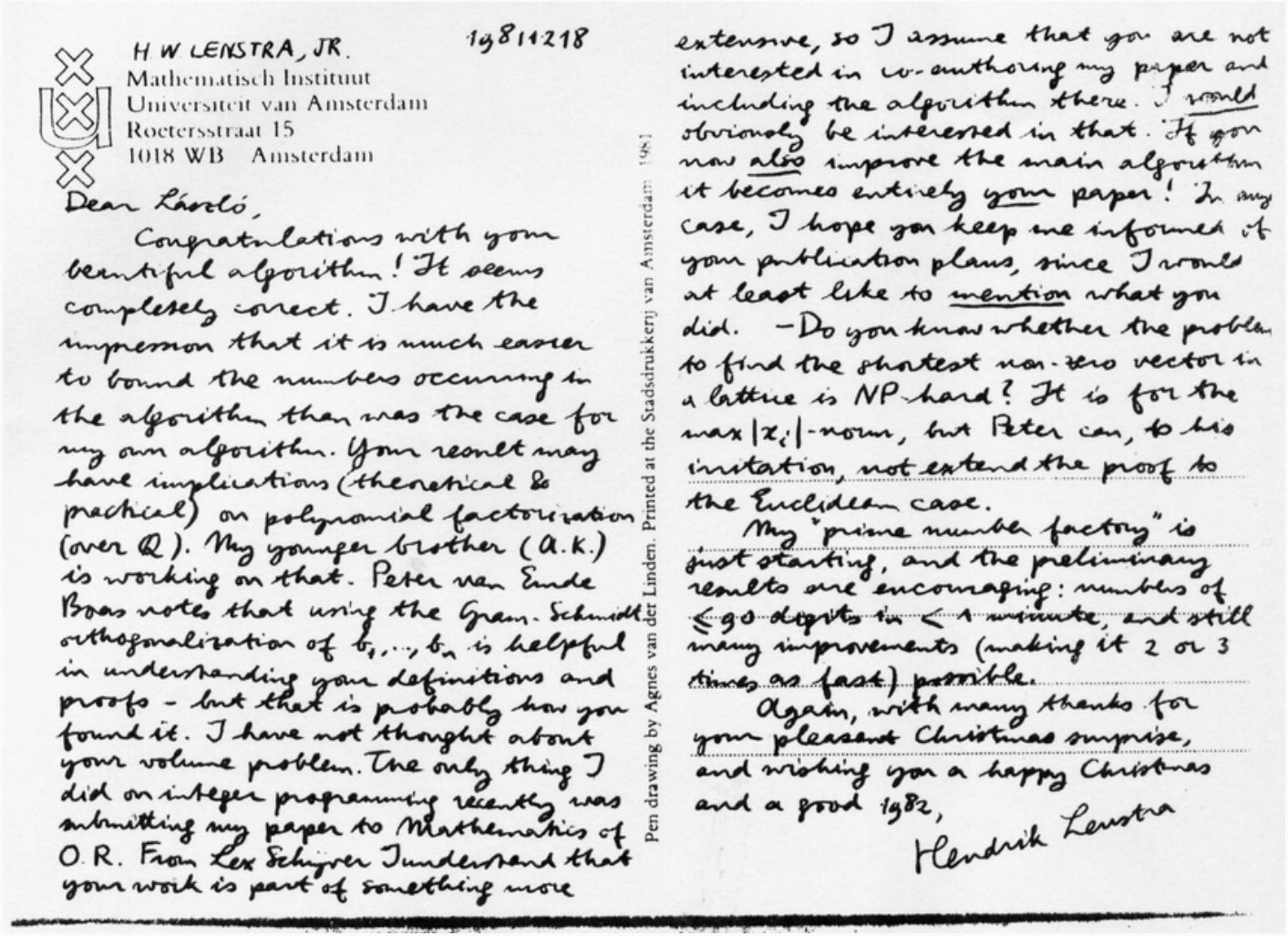
Family picture

Peter van Emde Boas, László Lovász, Hendrik Lenstra and Arjen Lenstra.
(Bonn on February 27th 1982)

Family picture!

Peter van Emde Boas, László Lovász, Hendrik Lenstra and Arjen Lenstra.
(Le moulin de Bully on June 29th 2007)

Greetings from Hendrik



H W LENSTRA, JR.
Mathematisch Instituut
Universiteit van Amsterdam
Roetersstraat 15
1018 WB Amsterdam

19811218

Dear László,

Congratulations with your beautiful algorithm! It seems completely correct. I have the impression that it is much easier to bound the numbers occurring in the algorithm than was the case for my own algorithm. Your result may have implications (theoretical & practical) on polynomial factorization (over $\mathbb{Q}$). My younger brother (A.K.) is working on that. Peter van Emde Boas notes that using the Gram-Schmidt orthogonalization of $b_1, \ldots, b_n$ is helpful in understanding your definitions and proofs - but that is probably how you found it. I have not thought about your volume problem. The only thing I did on integer programming recently was submitting my paper to Mathematics of O.R. From Lex Schrijver I understand that your work is part of something more extensive, so I assume that you are not interested in co-authoring my paper and including the algorithm there. I would obviously be interested in that. If you now also improve the main algorithm it becomes entirely your paper! In any case, I hope you keep me informed of your publication plans, since I would at least like to mention what you did. - Do you know whether the problem to find the shortest non-zero vector in a lattice is NP-hard? It is for the $\max |x_i|$-norm, but Peter can, to his irritation, not extend the proof to the Euclidean case.

My "prime number factory" is just starting, and the preliminary results are encouraging: numbers of ≤ 90 digits in < 1 minute, and still many improvements (making it 2 or 3 times as fast) possible.

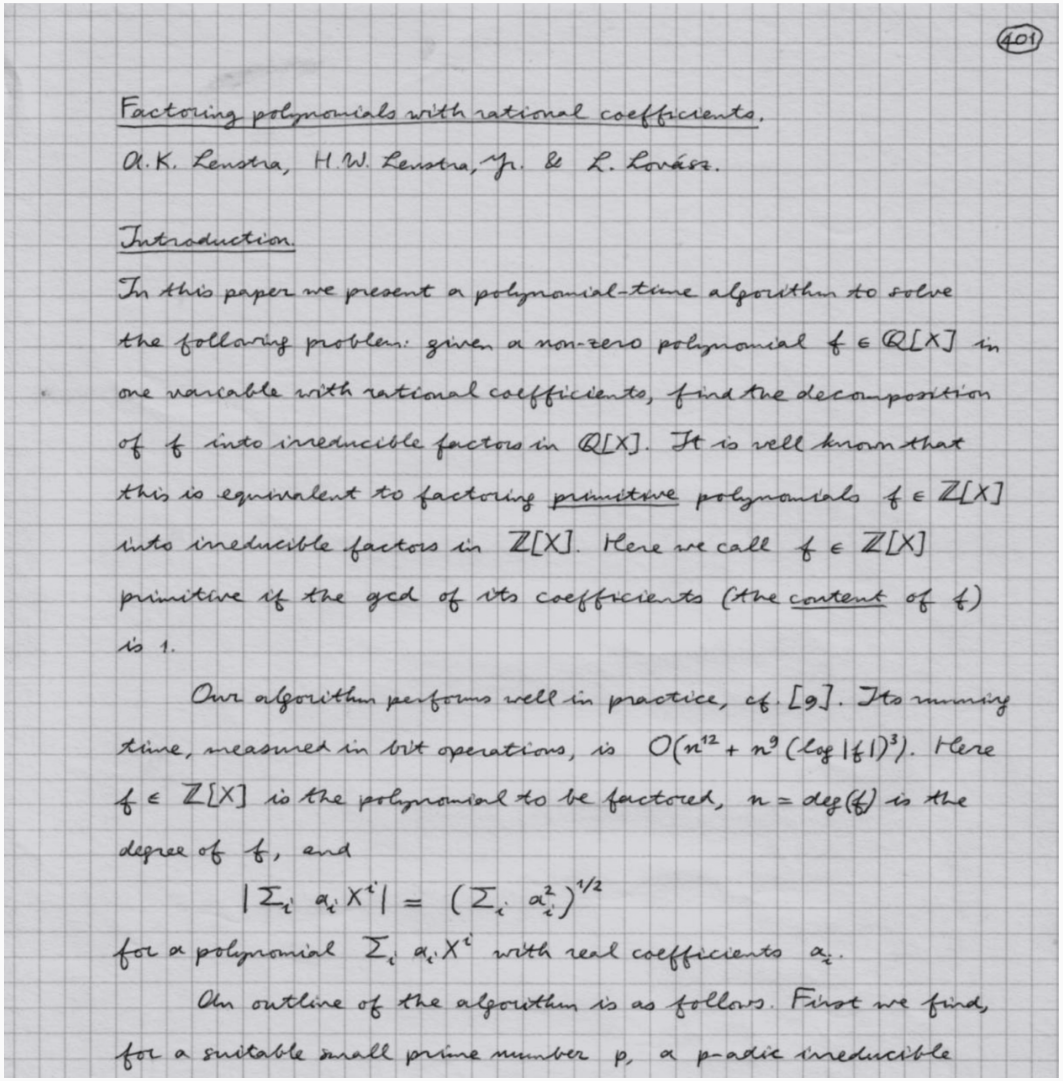
Again, with many thanks for your pleasant Christmas surprise, and wishing you a happy Christmas and a good 1982,

Hendrik Lenstra

Pen drawing by Agnes van der Linden. Printed at the Stadsdrukkerij van Amsterdam 1981

Greetings... 31 years ago!

First draft of the paper



(401)

Factoring polynomials with rational coefficients.

A.K. Lenstra, H.W. Lenstra, Jr. & L. Lovász.

Introduction.

In this paper we present a polynomial-time algorithm to solve the following problem: given a non-zero polynomial $f \in \mathbb{Q}[X]$ in one variable with rational coefficients, find the decomposition of f into irreducible factors in $\mathbb{Q}[X]$. It is well known that this is equivalent to factoring primitive polynomials $f \in \mathbb{Z}[X]$ into irreducible factors in $\mathbb{Z}[X]$. Here we call $f \in \mathbb{Z}[X]$ primitive if the gcd of its coefficients (the content of f) is 1.

Our algorithm performs well in practice, cf. [9]. Its running time, measured in bit operations, is $O(n^{12} + n^9 (\log |f|)^3)$. Here $f \in \mathbb{Z}[X]$ is the polynomial to be factored, $n = \deg(f)$ is the degree of f, and

$$|\textstyle\sum_i a_i X^i| = (\sum_i a_i^2)^{1/2}$$

for a polynomial $\sum_i a_i X^i$ with real coefficients a_i.

Our outline of the algorithm is as follows. First we find, for a suitable small prime number p, a p-adic irreducible

LLL reduction (1982)

Algorithm 2: The original LLL algorithm.

1 Compute the $\pi_i(b_i)$'s with the gso process ;
// Size-reduction steps
2 **for** $i = 2$ *to* d **do**
3 **for** $j = i - 1$ *to* d **do**
4 $b_i \leftarrow b_i - \left\lceil \frac{\langle b_i, \pi_i(b_j) \rangle}{\|\pi_i(b_i)\|^2} \right\rfloor \cdot b_j$;
5 **end**
6 **end**
// Test if the Lovász condition is fulfilled.
7 **for** $i = d - 1$ *to* 1 **do**
8 **if** $\delta \|\pi_i(b_i)\|^2 > \left(\|\pi_{i+1}(b_{i+1})\|^2 + \frac{\langle b_{i+1}, \pi_i(b_i) \rangle}{\|\pi_i(b_i)\|^2} \right)$ **then**
9 Swap b_i and b_{i+1};
10 **goto** 1;
11 **end**
12 **end**
13 **return** $(b_1, \ldots, b_d)$

Guarantees offered by LLL

Bounds on flag's covolumes

Let $1/4 < \delta < 1$ be an admissible LLL parameter. Let $(b_1, \ldots, b_d)$ a δ-LLL reduced basis of rank-d lattice $(\Lambda, \langle \cdot, \cdot \rangle)$. Then for any $1 \leq k \leq d$:

$$\mathrm{covol}(b_1, \ldots, b_k) \leq \left(\delta - \frac{1}{4}\right)^{-\frac{(n-k)k}{4}} \mathrm{covol}(\Lambda)^{\frac{k}{d}}$$

But... How fast is this reduction?

- Naive analysis:

$$\mathrm{O}\left(d^6 \log^3 \|B\|_\infty\right)$$

But... How fast is this reduction?

- Naive analysis:

$$\mathrm{O}\left(d^6 \log^3 \|B\|_\infty\right)$$

- Refined analysis:

$$\mathrm{O}\left(\frac{d^5 \log^2 \|B\|_\infty}{d + \log \|B\|_\infty} M(d + \log \|B\|_\infty)\right)$$

But... How fast is this reduction?

- Naive analysis:

$$\mathrm{O}\left(d^6 \log^3 \|B\|_\infty\right)$$

- Refined analysis:

$$\mathrm{O}\left(\frac{d^5 \log^2 \|B\|_\infty}{d + \log \|B\|_\infty} M(d + \log \|B\|_\infty)\right)$$

- Still slow... Bottleneck: Size of numerators/denominators in GSO computations.

And what if we use an approximation?

- If very precautious one can use floating-point representation of the GSO.

LLL reduced basis

(relaxed) LLL reduction

- Size-Reduction condition

$$\forall i < j, \quad |\langle b_j, \pi_i(b_i)\rangle| \leq \frac{1}{2}\|\pi_i(b_i)\|^2$$

- Lovász condition

$$\forall i, \quad \delta\|\pi_i(b_i)\|^2 \leq \left(\|\pi_{i+1}(b_{i+1})\|^2 + \frac{\langle b_{i+1}, \pi_i(b_i)\rangle}{\|\pi_i(b_i)\|^2}\right)$$

LLL reduced basis

(relaxed) LLL reduction

- Size-Reduction condition

$$\forall i < j, \quad |\langle b_j, \pi_i(b_i)\rangle| \leq \eta \|\pi_i(b_i)\|^2$$

- Lovász condition

$$\forall i, \quad \delta\|\pi_i(b_i)\|^2 \leq \left(\|\pi_{i+1}(b_{i+1})\|^2 + \frac{\langle b_{i+1}, \pi_i(b_i)\rangle}{\|\pi_i(b_i)\|^2}\right)$$

And what if we use an approximation?

- If very precautious one can use floating-point representation of the GSO.

And what if we use an approximation?

- If very precautious one can use floating-point representation of the GSO.
- From refined analysis of the exact case:

$$\mathrm{O}\left(\frac{d^5 \log^2 \|B\|_\infty}{d + \log \|B\|_\infty} M(d + \log \|B\|_\infty)\right)$$

And what if we use an approximation?

- If very precautious one can use floating-point representation of the GSO.
- From refined analysis of the exact case:

$$\mathrm{O}\left(\frac{d^5 \log^2 \|B\|_\infty}{d + \log \|B\|_\infty} M(d + \log \|B\|_\infty)\right)$$

- We go down to:

$$\mathrm{O}\left(d^4 \log(\|B\|_\infty) M(d + \log \|B\|_\infty)\right)$$

Faster (!)

And what if we use an approximation?

- If very precautious one can use floating-point representation of the GSO.
- From refined analysis of the exact case:

$$\mathrm{O}\left(\frac{d^5 \log^2 \|B\|_\infty}{d + \log \|B\|_\infty} M(d + \log \|B\|_\infty)\right)$$

- We go down to:

$$\mathrm{O}\left(d^4 \log(\|B\|_\infty) M(d + \log \|B\|_\infty)\right)$$

  Faster (!)
- Still cubic in

$$\log \|B\|_\infty$$

And what if we use an approximation?

- Fastest floating-point variant of Nguyen-Stehlé:

$$\mathrm{O}\big(d^5(d + \log(\|B\|_\infty))\log(\|B\|_\infty)\big)$$

And what if we use an approximation?

- Fastest floating-point variant of Nguyen-Stehlé:

$$\mathrm{O}\big(d^5(d + \log(\|B\|_\infty))\log(\|B\|_\infty)\big)$$

- Precision required

$$d\log\left(\frac{(1+\eta)^2}{(\delta-\eta)^2} + \epsilon\right) + o(d)$$

And what if we use an approximation?

- Fastest floating-point variant of Nguyen-Stehlé:

$$\mathrm{O}\big(d^5(d + \log(\|B\|_\infty))\log(\|B\|_\infty)\big)$$

- Precision required

$$d\log\left(\frac{(1+\eta)^2}{(\delta-\eta)^2}+\epsilon\right)+o(d)$$

- But...

But...

In practice the precision needed is

$$0.25d + o(d)$$

bits

Summary of the limitations

- Non-optimal number of bits used...
- No proved way of dealing with external inner product

A primer on Interval arithmetic

Intervals and their bounds

$\underline{x}$ is an interval

An algebra of intervals

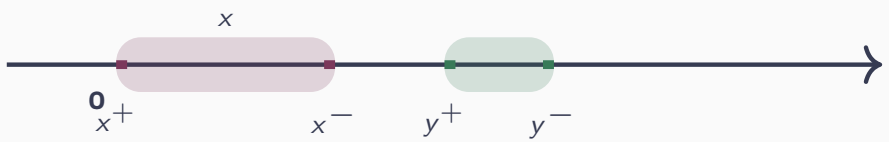
- $\bowtie$ is a binary operation—resp. f be a function—over the reals.
- The result $\underline{x} \bowtie \underline{y}$ between the intervals $\underline{x}$ and $\underline{y}$—resp $f(\underline{x})$, is the smallest interval, in the sense of inclusion, containing

$$\{x \bowtie y | (x, y) \in \underline{x} \times \underline{y}\} \qquad \text{— resp. } \{f(x) | x \in \underline{x}\} \text{ —}$$

An algebra of intervals

- $\bowtie$ is a binary operation—resp. f be a function—over the reals.
- The result $\underline{x} \bowtie \underline{y}$ between the intervals $\underline{x}$ and $\underline{y}$—resp $f(\underline{x})$, is the smallest interval, in the sense of inclusion, containing

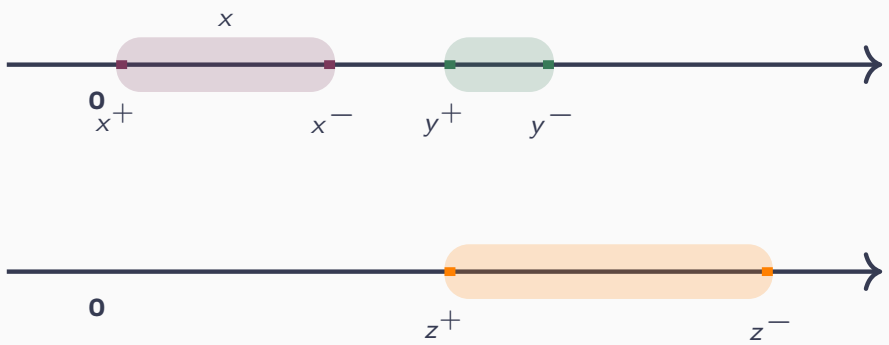
$$\{x \bowtie y | (x, y) \in \underline{x} \times \underline{y}\} \qquad \text{— resp. } \{f(x) | x \in \underline{x}\} \text{ —}$$



An algebra of intervals

- $\bowtie$ is a binary operation—resp. f be a function—over the reals.
- The result $\underline{x} \bowtie \underline{y}$ between the intervals $\underline{x}$ and $\underline{y}$—resp $f(\underline{x})$, is the smallest interval, in the sense of inclusion, containing

$$\{x \bowtie y | (x, y) \in \underline{x} \times \underline{y}\} \qquad \text{— resp. } \{f(x) | x \in \underline{x}\} \text{ —}$$



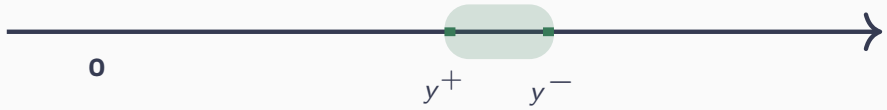
Why is it useful?

- When real numbers are represented by intervals, the interval resulting from the evaluation of an algebraic expression contains the exact value of the evaluated expression.
- For $(x_i)_{1\leq i\leq n}$ and $(\underline{x}_i)_{1\leq i\leq n}$ intervals such as for each i, $x_i \in \underline{x}_i$

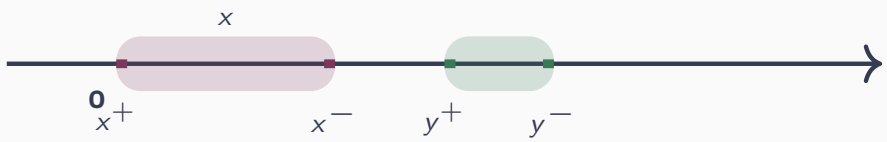
$$f(x_1, \ldots, x_n) \in f(\underline{x}_1, \ldots, \underline{x}_N),$$

  for any algebraic expression f.
- Results given in Interval Arithmetic are certified.

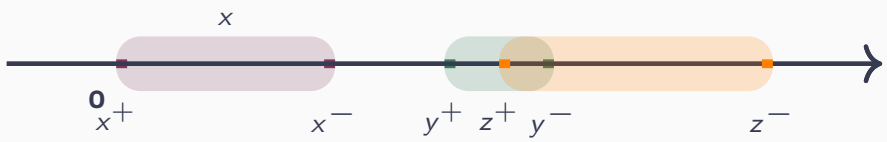
Certification property of an inequality



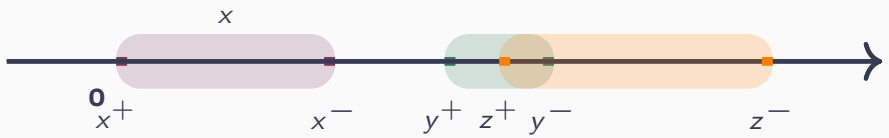
Certification property of an inequality



Certification property of an inequality



Certification property of an inequality



Interval Arithmetic used in such a way allows detecting a lack of precision or accuracy at runtime of a numerical algorithm.

The idea

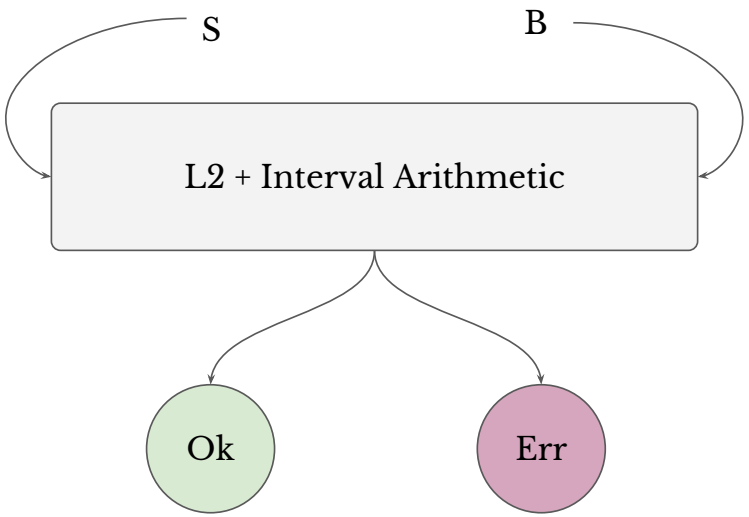
Certified lattice reduction

Interval arithmetic + LLL

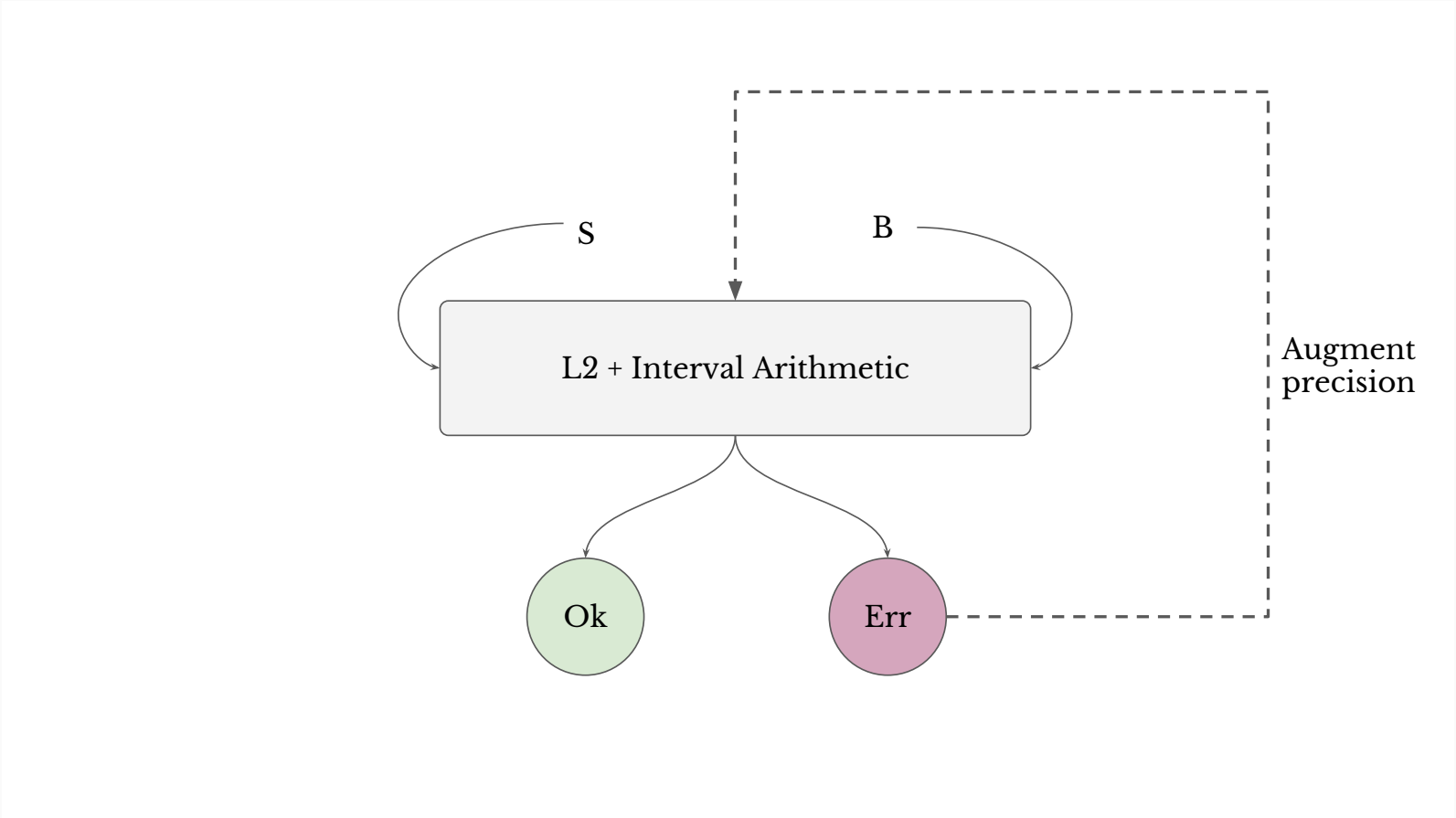
=

Certified Lattice Reduction

Adaptive precision lattice reduction



Adaptive precision lattice reduction



Application: Towards a sound algorithmic Minkowsky Theory

Number field and embeddings

- $\mathbf{K} = \mathbf{Q}(\alpha)$ number field of dimension d,

$$\mathbf{K} \cong \mathbf{Q}[X]/(P) \qquad \text{with } P(\alpha) = 0$$

Number field and embeddings

- $\mathbf{K} = \mathbf{Q}(\alpha)$ number field of dimension d,

$$\mathbf{K} \cong \mathbf{Q}[X]/(P) \qquad \text{with } P(\alpha) = 0$$

- $(\alpha_1, \ldots, \alpha_d) \in \mathbf{C}^n$ complex roots of P.

Number field and embeddings

- $\mathbf{K} = \mathbf{Q}(\alpha)$ number field of dimension d,

$$\mathbf{K} \cong \mathbf{Q}[X]/(P) \qquad \text{with } P(\alpha) = 0$$

- $(\alpha_1, \ldots, \alpha_d) \in \mathbf{C}^n$ complex roots of P.
- Embedding $\sigma_i : \mathbf{K} \to \mathbf{C}$ is the evaluation of $a \in \mathbf{K}$, viewed as a polynomial mod P at the root α_i:

$$\sigma_i : a \mapsto a(\alpha_i)$$

Number field and embeddings

- $\mathbf{K} = \mathbf{Q}(\alpha)$ number field of dimension d,

$$\mathbf{K} \cong \mathbf{Q}[X]/(P) \qquad \text{with } P(\alpha) = 0$$

- $(\alpha_1, \ldots, \alpha_d) \in \mathbf{C}^n$ complex roots of P.
- Embedding $\sigma_i : \mathbf{K} \to \mathbf{C}$ is the evaluation of $a \in \mathbf{K}$, viewed as a polynomial mod P at the root α_i:

$$\sigma_i : a \mapsto a(\alpha_i)$$

- $d = \overbrace{r_1}^{\mathbf{R}} + 2\overbrace{r_2}^{\mathbf{C}}$ then $\mathbf{K} \otimes_{\mathbf{Q}} \mathbf{R} \cong \mathbf{R}^{r_1} \times \mathbf{C}^{r_2} \cong \mathbf{R}^d$ by:

Number field and embeddings

- $\mathbf{K} = \mathbf{Q}(\alpha)$ number field of dimension d,

$$\mathbf{K} \cong \mathbf{Q}[X]/(P) \qquad \text{with } P(\alpha) = 0$$

- $(\alpha_1, \ldots, \alpha_d) \in \mathbf{C}^n$ complex roots of P.
- Embedding $\sigma_i : \mathbf{K} \to \mathbf{C}$ is the evaluation of $a \in \mathbf{K}$, viewed as a polynomial mod P at the root α_i:

$$\sigma_i : a \mapsto a(\alpha_i)$$

- $d = \overbrace{r_1}^{\mathbf{R}} + 2\overbrace{r_2}^{\mathbf{C}}$ then $\mathbf{K} \otimes_{\mathbf{Q}} \mathbf{R} \cong \mathbf{R}^{r_1} \times \mathbf{C}^{r_2} \cong \mathbf{R}^d$ by:

$$\sigma : \left| \begin{array}{ccccc} \mathbf{K} \otimes_{\mathbf{Q}} \mathbf{R} & \longrightarrow & \mathbf{R}^{r_1} \times \mathbf{C}^{r_2} & \longrightarrow & \mathbf{R}^d \\ x & \longmapsto & \underbrace{(\sigma_1(x) \ldots \sigma_{r_1}(x)}_{r}, \underbrace{\sigma_{r_1+1}(x) \ldots \sigma_{r_1+r_2}(x))^T}_{c} & \longmapsto & (r, \sqrt{2}\Re(c), \sqrt{2}\Im(c))^T \end{array} \right.$$

Ring of integers

- An element γ of $\mathbf{K}$ is said to be integral if its minimal polynomial has integral coefficients and is monic.
- The ring of integers $\mathfrak{o}_{\mathbf{K}}$, or maximal order, of $\mathbf{K}$ is the ring of all integral elements contained in $\mathbf{K}$.
- A free $\mathbf{Z}$-module $\mathfrak{o}$ embedded in $\mathbf{K}$, such as $\mathbf{Q}\mathfrak{o} = \mathbf{K}$ is called an order of $\mathbf{K}$, and lies in $\mathfrak{o}_{\mathbf{K}}$.

$$\mathfrak{o} \cong \bigoplus_{i \in I} \omega_i \mathbf{Z}.$$

Canonical embedding and ideals

- An additive subgroup $\mathfrak{a}$ of $\mathfrak{o}$ for which $\forall a \in \mathfrak{a}$

$$a \cdot \mathfrak{o} = \{a \cdot x | x \in \mathfrak{o}\} \subset \mathfrak{a}$$

  is called an ideal of the number field.
- The Archimedean embedding endows any integral ideal $\mathfrak{a}$ of an order $\mathfrak{o}$ with a lattice structure:

$$(\mathfrak{a}, \langle \cdot, \cdot \rangle_\sigma)$$

Lattices from orders

- For any order, $(\mathfrak{o}, \langle \cdot, \cdot \rangle_\sigma))$ is also a lattice. The square of its (co)volume, denoted by $\Delta_\mathfrak{o}$, is called its discriminant.
- For $\Omega = (\omega_1, \ldots, \omega_n)$ an integral basis of $\mathfrak{o}$, we have

$$\Delta_\mathfrak{o} = \det(S_{\sigma,\Omega}) = \left| \det \begin{pmatrix} \sigma_1(\omega_1) & \sigma_1(\omega_2) & \cdots & \sigma_1(\omega_n) \\ \sigma_2(\omega_1) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_n(\omega_1) & \cdots & \cdots & \sigma_n(\omega_n) \end{pmatrix} \right|^2,$$

where $S_\Omega = \left(\sum_\sigma \sigma(\omega_i)\overline{\sigma(\omega_j)} \right)_{i,j}$ is the Gram-matrix associated to $\mathcal{B}$.

Practical reduction theory for ideals

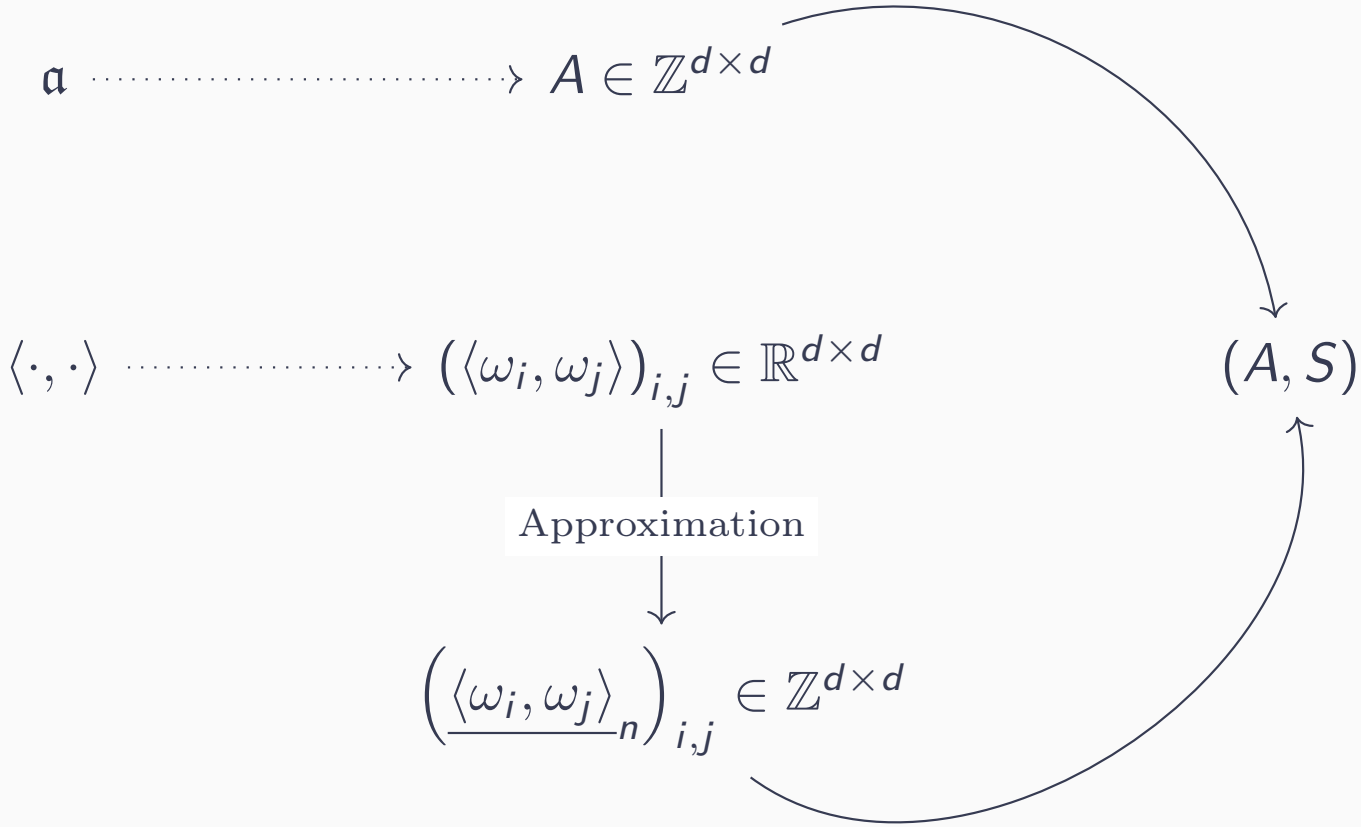
- $\mathcal{B} = (\omega_1, \ldots, \omega_n)$ integral basis of an order $\mathfrak{o}$,
- Every ideal $\mathfrak{a} \subseteq \mathfrak{o}$ can be described by its decomposition in $\mathcal{B}$ (full-rank $\mathbf{Z}$-submodule of $\mathfrak{o}$), $\mathfrak{a}$ as an integer valued matrix A.
- The scalar product of the corresponding lattice is $S_{\sigma,\mathcal{B}}$

Practical reduction theory for ideals

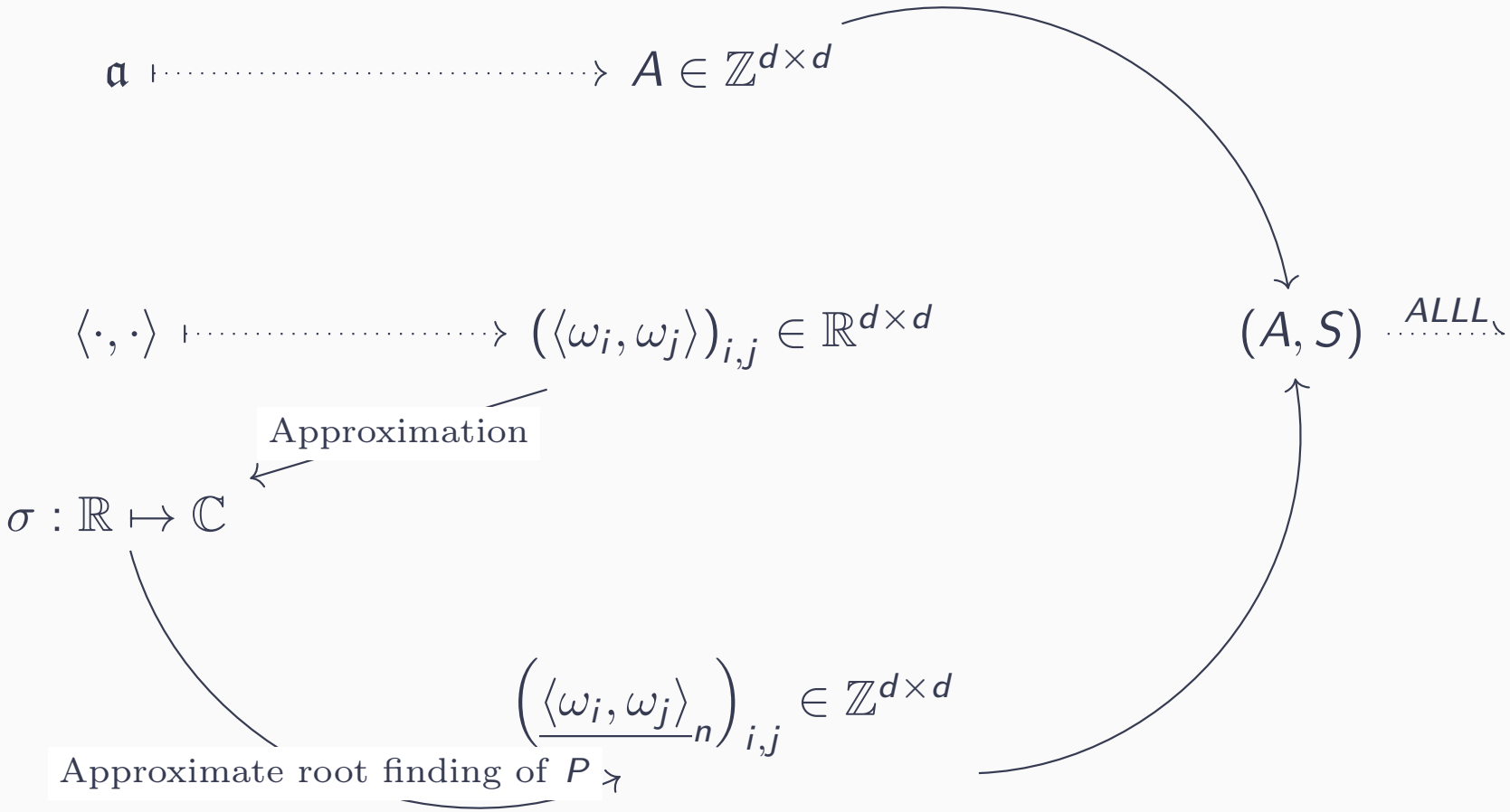
$$\mathfrak{a} \dashrightarrow A \in \mathbb{Z}^{d\times d}$$

$$\langle \cdot, \cdot \rangle \dashrightarrow \left(\langle \omega_i, \omega_j \rangle\right)_{i,j} \in \mathbb{R}^{d\times d}$$

Practical reduction theory for ideals



Practical reduction theory for ideals



Complexity perspective

LLL is polynomial-time in the size of its inputs

Complexity perspective

LLL is polynomial-time in the size of its inputs

but

Complexity perspective

LLL is polynomial-time in the size of its inputs

but

also in the least eigenvalue of the representation of the order.

Take-away message

- In practice: Faster than proved version of fpLLL (state-of-the-art).

Take-away message

- In practice: Faster than proved version of fpLLL (state-of-the-art).
- Has the same execution flow as the exact LLL: can be used to perform sound experiments on the average behavior of LLL.

Take-away message

- In practice: Faster than proved version of fpLLL (state-of-the-art).
- Has the same execution flow as the exact LLL: can be used to perform sound experiments on the average behavior of LLL.
- Methodology is directly applicable to some generalizations of lattices (hermitian vector bundles over arithmetic curves for instance).

Thank you !