The LLL Algorithm: Lattice Basis Reduction and applications to Approximate Shortest Vector Problem

Lucas Petit

May 26, 2025

Recap: Euclidean Space and Inner Product

Recap: Euclidean Space and Inner Product

We consider a real finite-dimensional vector space $\mathbb{R}^n$ equipped with the standard **Euclidean inner product**:

$$\langle \mathbf{u}, \mathbf{v} \rangle := \sum_{i=1}^{n} \mathbf{u}_i \mathbf{v}_i$$

Recap: Euclidean Space and Inner Product

We consider a real finite-dimensional vector space $\mathbb{R}^n$ equipped with the standard **Euclidean inner product**:

$$\langle \mathbf{u}, \mathbf{v} \rangle := \sum_{i=1}^{n} \mathbf{u}_i \mathbf{v}_i$$

This inner product induces the **Euclidean norm**:

$$\|\mathbf{u}\|_2 = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} = \sqrt{\sum_{i=1}^{n} \mathbf{u}_i^2}$$

Recap: Euclidean Lattice

Recap: Euclidean Lattice

A **Euclidean lattice** $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.

Recap: Euclidean Lattice

A **Euclidean lattice** $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.

- **Additive subgroup:**

$$\mathbf{0} \in \mathscr{L},\ \mathbf{x} + \mathbf{y} \in \mathscr{L},\ -\mathbf{x} \in \mathscr{L} \text{ for all } \mathbf{x}, \mathbf{y} \in \mathscr{L}.$$

Recap: Euclidean Lattice

A **Euclidean lattice** $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.

- **Additive subgroup:**

$$\mathbf{0} \in \mathscr{L},\ \mathbf{x}+\mathbf{y} \in \mathscr{L},\ -\mathbf{x} \in \mathscr{L} \text{ for all } \mathbf{x}, \mathbf{y} \in \mathscr{L}.$$

- **Discrete:** For every $\mathbf{x} \in \mathscr{L}$, there exists $\varepsilon > 0$ such that

$$\mathcal{B}(\mathbf{x}, \varepsilon) \cap \mathscr{L} = \{\mathbf{x}\}$$

  where $\mathcal{B}(\mathbf{x}, \varepsilon)$ denotes the open ball of radius ε centered at $\mathbf{x}$.

Recap: Euclidean Lattice

A **Euclidean lattice** $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.

- **Additive subgroup:**

$$\mathbf{0} \in \mathscr{L},\, \mathbf{x} + \mathbf{y} \in \mathscr{L},\ -\mathbf{x} \in \mathscr{L} \text{ for all } \mathbf{x}, \mathbf{y} \in \mathscr{L}.$$

- **Discrete:** For every $\mathbf{x} \in \mathscr{L}$, there exists $\varepsilon > 0$ such that

$$\mathcal{B}(\mathbf{x}, \varepsilon) \cap \mathscr{L} = \{\mathbf{x}\}$$

  where $\mathcal{B}(\mathbf{x}, \varepsilon)$ denotes the open ball of radius ε centered at $\mathbf{x}$.

Figure: Example of lattice in $\mathbb{R}^2$

Recap: Euclidean Lattice

A **Euclidean lattice** $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.

- **Additive subgroup:**

$$\mathbf{0} \in \mathscr{L}, \; \mathbf{x} + \mathbf{y} \in \mathscr{L}, \; -\mathbf{x} \in \mathscr{L} \text{ for all } \mathbf{x}, \mathbf{y} \in \mathscr{L}.$$

- **Discrete:** For every $\mathbf{x} \in \mathscr{L}$, there exists $\varepsilon > 0$ such that

$$\mathcal{B}(\mathbf{x}, \varepsilon) \cap \mathscr{L} = \{\mathbf{x}\}$$

where $\mathcal{B}(\mathbf{x}, \varepsilon)$ denotes the open ball of radius ε centered at $\mathbf{x}$.

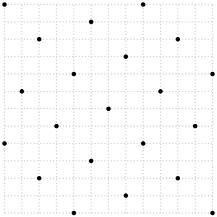


Figure: Example of lattice in $\mathbb{R}^2$

Recap: Euclidean Lattice

A **Euclidean lattice** $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.

- **Additive subgroup:**

$$\mathbf{0} \in \mathscr{L}, \mathbf{x} + \mathbf{y} \in \mathscr{L}, \; -\mathbf{x} \in \mathscr{L} \text{ for all } \mathbf{x}, \mathbf{y} \in \mathscr{L}.$$

- **Discrete:** For every $\mathbf{x} \in \mathscr{L}$, there exists $\varepsilon > 0$ such that

$$\mathcal{B}(\mathbf{x}, \varepsilon) \cap \mathscr{L} = \{\mathbf{x}\}$$

  where $\mathcal{B}(\mathbf{x}, \varepsilon)$ denotes the open ball of radius ε centered at $\mathbf{x}$.

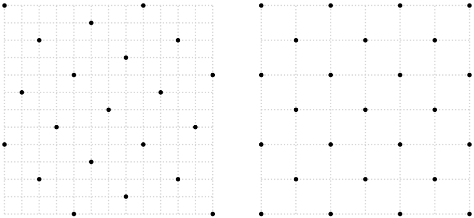


Figure: Example of lattice in $\mathbb{R}^2$

Recap: Euclidean Lattice

A **Euclidean lattice** $\mathscr{L}$ is a discrete additive subgroup of $\mathbb{R}^n$.

- **Additive subgroup:**

$$\mathbf{0} \in \mathscr{L},\ \mathbf{x} + \mathbf{y} \in \mathscr{L},\ -\mathbf{x} \in \mathscr{L} \text{ for all } \mathbf{x}, \mathbf{y} \in \mathscr{L}.$$

- **Discrete:** For every $\mathbf{x} \in \mathscr{L}$, there exists $\varepsilon > 0$ such that

$$\mathcal{B}(\mathbf{x}, \varepsilon) \cap \mathscr{L} = \{\mathbf{x}\}$$

  where $\mathcal{B}(\mathbf{x}, \varepsilon)$ denotes the open ball of radius ε centered at $\mathbf{x}$.

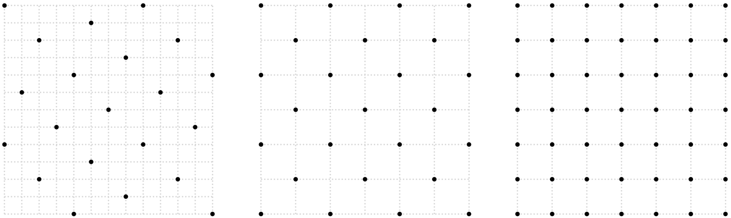


Figure: Example of lattice in $\mathbb{R}^2$

Recap: Lattice Bases

Recap: Lattice Bases

Any lattice $\mathscr{L} \subseteq \mathbb{R}^n$ admits a maximal $\mathbb{Z}$-linearly independent family $(\mathbf{b}_i)_{1 \leq i \leq m}$, with $m \leq n$ such that:

$$\mathscr{L} = \bigoplus_{i=1}^{m} \mathbb{Z}\mathbf{b}_i = \{a_1\mathbf{b}_1 + \cdots + a_m\mathbf{b}_m \mid a_i \in \mathbb{Z}\}$$

Recap: Lattice Bases

Any lattice $\mathscr{L} \subseteq \mathbb{R}^n$ admits a maximal $\mathbb{Z}$-linearly independent family $(\mathbf{b}_i)_{1 \leq i \leq m}$, with $m \leq n$ such that:

$$\mathscr{L} = \bigoplus_{i=1}^{m} \mathbb{Z}\mathbf{b}_i = \{a_1\mathbf{b}_1 + \cdots + a_m\mathbf{b}_m \mid a_i \in \mathbb{Z}\}$$

This family is called a **basis** of the lattice $\mathscr{L}$.

Recap: Lattice Bases

Any lattice $\mathscr{L} \subseteq \mathbb{R}^n$ admits a maximal $\mathbb{Z}$-linearly independent family $(\mathbf{b}_i)_{1 \leq i \leq m}$, with $m \leq n$ such that:

$$\mathscr{L} = \bigoplus_{i=1}^{m} \mathbb{Z}\mathbf{b}_i = \{a_1\mathbf{b}_1 + \cdots + a_m\mathbf{b}_m \mid a_i \in \mathbb{Z}\}$$

This family is called a **basis** of the lattice $\mathscr{L}$.

Figure: Example of lattice with different basis in $\mathbb{R}^2$

Recap: Lattice Bases

Any lattice $\mathscr{L} \subseteq \mathbb{R}^n$ admits a maximal $\mathbb{Z}$-linearly independent family $(\mathbf{b}_i)_{1\leq i\leq m}$, with $m \leq n$ such that:

$$\mathscr{L} = \bigoplus_{i=1}^{m} \mathbb{Z}\mathbf{b}_i = \{a_1\mathbf{b}_1 + \cdots + a_m\mathbf{b}_m \mid a_i \in \mathbb{Z}\}$$

This family is called a **basis** of the lattice $\mathscr{L}$.

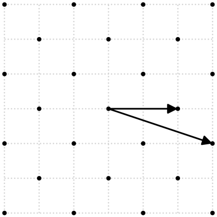


Figure: Example of lattice with different basis in $\mathbb{R}^2$

Recap: Lattice Bases

Any lattice $\mathscr{L} \subseteq \mathbb{R}^n$ admits a maximal $\mathbb{Z}$-linearly independent family $(\mathbf{b}_i)_{1 \le i \le m}$, with $m \le n$ such that:

$$\mathscr{L} = \bigoplus_{i=1}^{m} \mathbb{Z}\mathbf{b}_i = \{a_1\mathbf{b}_1 + \cdots + a_m\mathbf{b}_m \mid a_i \in \mathbb{Z}\}$$

This family is called a **basis** of the lattice $\mathscr{L}$.

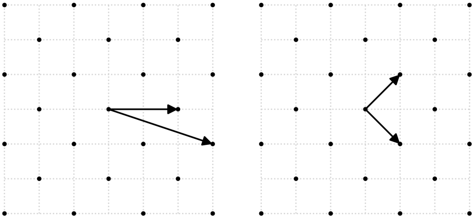


Figure: Example of lattice with different basis in $\mathbb{R}^2$

Recap: Lattice Bases

Any lattice $\mathscr{L} \subseteq \mathbb{R}^n$ admits a maximal $\mathbb{Z}$-linearly independent family $(\mathbf{b}_i)_{1 \le i \le m}$, with $m \le n$ such that:

$$\mathscr{L} = \bigoplus_{i=1}^{m} \mathbb{Z}\mathbf{b}_i = \{a_1\mathbf{b}_1 + \cdots + a_m\mathbf{b}_m \mid a_i \in \mathbb{Z}\}$$

This family is called a **basis** of the lattice $\mathscr{L}$.

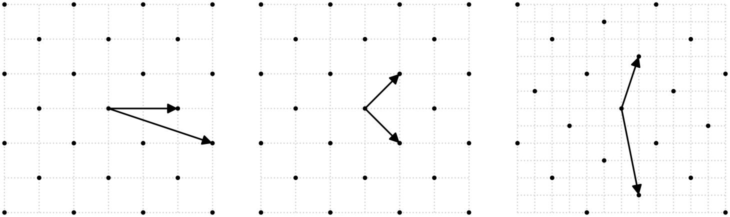
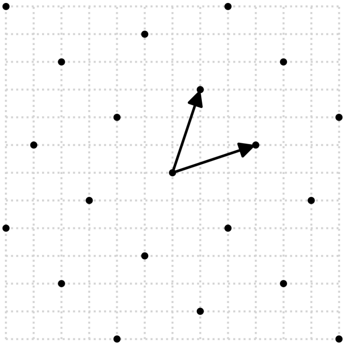


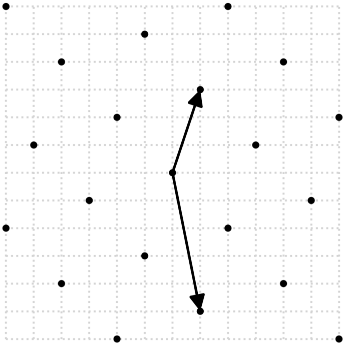
Figure: Example of lattice with different basis in $\mathbb{R}^2$

Two different bases of the same lattice

Two different bases of the same lattice

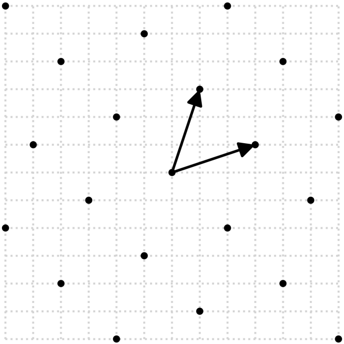


short, nearly orthogonal vectors
looks good

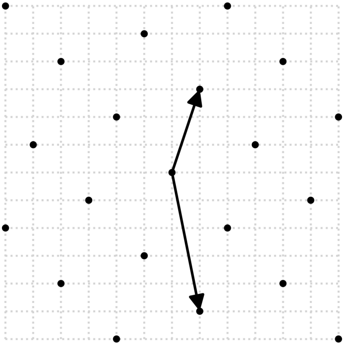


long, skewed basis vectors
looks bad

Two different bases of the same lattice



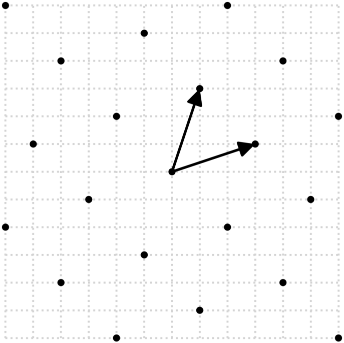
short, nearly orthogonal vectors
looks good



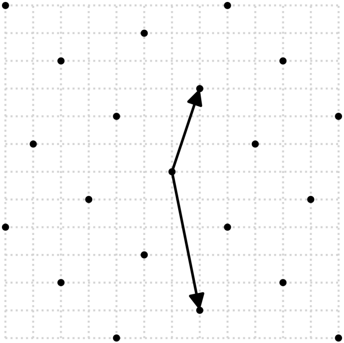
long, skewed basis vectors
looks bad

Can we formalize this?

Two different bases of the same lattice



short, nearly orthogonal vectors
looks good



long, skewed basis vectors
looks bad

Can we formalize this?

→ notion of **quasi-orthogonal** (or **reduced**) bases.

Recap: Orthogonal Bases and Gram-Schmidt Process

Recap: Orthogonal Bases and Gram-Schmidt Process

A basis $(\mathbf{b}_i)_{1 \le i \le n}$ of $\mathbb{R}^n$ is called **orthogonal** if

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for all } i \neq j.$$

Recap: Orthogonal Bases and Gram-Schmidt Process

A basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of $\mathbb{R}^n$ is called **orthogonal** if

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for all } i \neq j.$$

Figure: Orthogonal or not orthogonal basis

Recap: Orthogonal Bases and Gram-Schmidt Process

A basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of $\mathbb{R}^n$ is called **orthogonal** if

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for all } i \neq j.$$

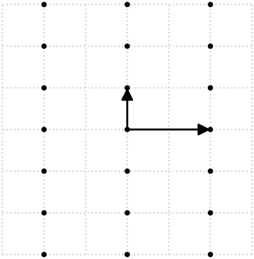


Figure: Orthogonal or not orthogonal basis

Recap: Orthogonal Bases and Gram-Schmidt Process

A basis $(\mathbf{b}_i)_{1 \leq i \leq n}$ of $\mathbb{R}^n$ is called **orthogonal** if

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for all } i \neq j.$$

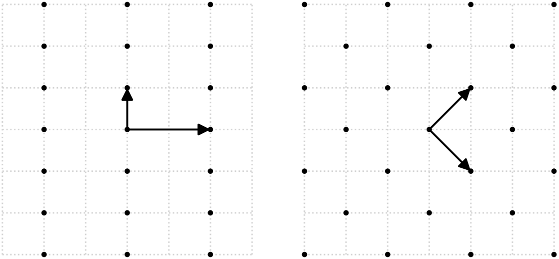


Figure: Orthogonal or not orthogonal basis

Recap: Orthogonal Bases and Gram-Schmidt Process

A basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of $\mathbb{R}^n$ is called **orthogonal** if

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for all } i \neq j.$$

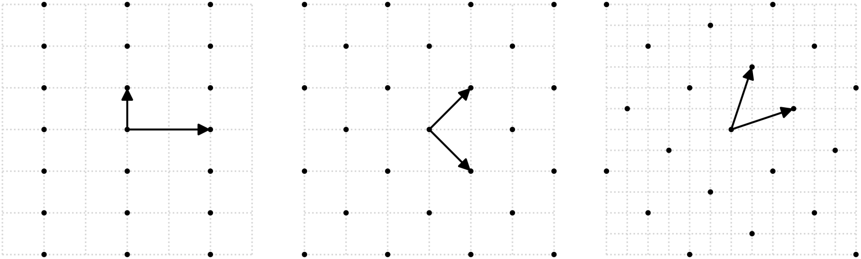


Figure: Orthogonal or not orthogonal basis

Recap: Orthogonal Bases and Gram-Schmidt Process

A basis $(\mathbf{b}_i)_{1 \le i \le n}$ of $\mathbb{R}^n$ is called **orthogonal** if

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for all } i \neq j.$$

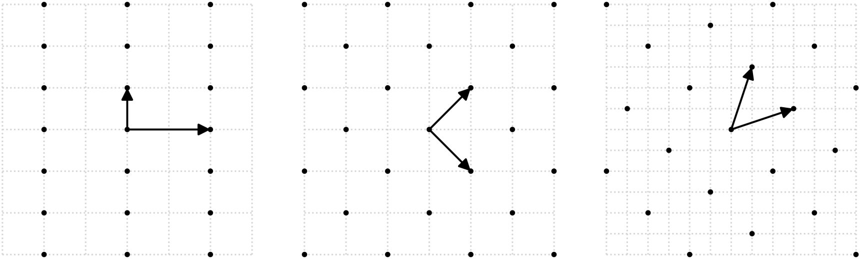


Figure: Orthogonal or not orthogonal basis

How an we compute an orthogonal basis ?

Recap: Orthogonal Bases and Gram-Schmidt Process

A basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of $\mathbb{R}^n$ is called **orthogonal** if

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \quad \text{for all } i \neq j.$$

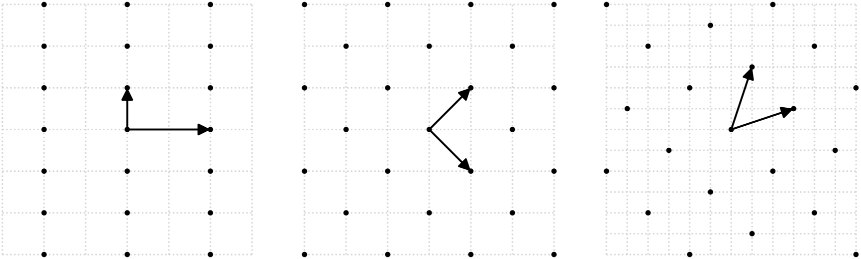


Figure: Orthogonal or not orthogonal basis

How an we compute an orthogonal basis ?

→ **Gram-Schmidt orthogonalization process**

Recap: Gram–Schmidt orthogonalization

Recap: Gram–Schmidt orthogonalization

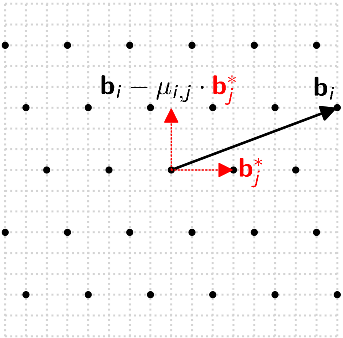
Let $(\mathbf{b}_i)_{1\le i\le n}$ be a basis of $\mathbb{R}^n$. The associated orthogonal basis $(\mathbf{b}_i^*)_{1\le i\le n}$ is constructed via the **Gram–Schmidt orthogonalization process**:

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

Recap: Gram–Schmidt orthogonalization

Let $(\mathbf{b}_i)_{1\leq i\leq n}$ be a basis of $\mathbb{R}^n$. The associated orthogonal basis $(\mathbf{b}_i^*)_{1\leq i\leq n}$ is constructed via the **Gram–Schmidt orthogonalization process**:

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



Recap: Gram–Schmidt orthogonalization

Recap: Gram–Schmidt orthogonalization

The coefficients $\mu_{i,j}$ are called **Gram–Schmidt coefficients**.

$$\begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \mu_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mu_{n,1} & \cdots & \mu_{n,n-1} & 1 \end{pmatrix} \times \begin{pmatrix} \mathbf{b}_1^* \\ \mathbf{b}_2^* \\ \vdots \\ \mathbf{b}_n^* \end{pmatrix}$$

Recap: Gram–Schmidt orthogonalization

The coefficients $\mu_{i,j}$ are called **Gram–Schmidt coefficients**.

$$\begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \mu_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mu_{n,1} & \cdots & \mu_{n,n-1} & 1 \end{pmatrix} \times \begin{pmatrix} \mathbf{b}_1^* \\ \mathbf{b}_2^* \\ \vdots \\ \mathbf{b}_n^* \end{pmatrix}$$

The resulting family $(\mathbf{b}_i^*)_{1 \leq i \leq n}$ is orthogonal.

Example: Gram–Schmidt Orthogonalization

Example: Gram–Schmidt Orthogonalization

Let

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

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^*$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* :=$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 :=$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 =$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 =$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1)$, $\|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1}$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\left\langle \mathbf{b}_2, \mathbf{b}_1^* \right\rangle}{\|\mathbf{b}_1^*\|^2}$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^*$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^* :=$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1)$, $\|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^* := \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^*$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1)$, $\|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^* := \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^* = (3, 0, 2) + \frac{4}{9}(-2, 2, 1)$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^* := \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^* = (3, 0, 2) + \frac{4}{9}(-2, 2, 1) = \left(\frac{19}{9}, \frac{8}{9}, \frac{22}{9}\right)$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^* := \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^* = (3, 0, 2) + \frac{4}{9}(-2, 2, 1) = \left(\frac{19}{9}, \frac{8}{9}, \frac{22}{9}\right)$

Step 3 : $\mu_{3,1} = 0$, $\mu_{3,2} = \frac{54}{101}$,

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^* := \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^* = (3, 0, 2) + \frac{4}{9}(-2, 2, 1) = \left(\frac{19}{9}, \frac{8}{9}, \frac{22}{9}\right)$

Step 3 : $\mu_{3,1} = 0$, $\mu_{3,2} = \frac{54}{101}, \mathbf{b}_3^* = \left(\frac{88}{101}, \frac{154}{101}, -\frac{132}{101}\right)$

Example: Gram–Schmidt Orthogonalization

Let

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

$$\textbf{Step 1 :} \quad \mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1), \ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$$

$$\textbf{Step 2 :} \quad \mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$$

$$\mathbf{b}_2^* := \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^* = (3, 0, 2) + \frac{4}{9}(-2, 2, 1) = \left(\frac{19}{9}, \frac{8}{9}, \frac{22}{9}\right)$$

$$\textbf{Step 3 :} \ \mu_{3,1} = 0 \ , \ \mu_{3,2} = \frac{54}{101}, \mathbf{b}_3^* = \left(\frac{88}{101}, \frac{154}{101}, -\frac{132}{101}\right)$$

$$\overbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}}^{B} =$$

Example: Gram–Schmidt Orthogonalization

Let

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

Step 1 : $\mathbf{b}_1^* := \mathbf{b}_1 := (-2, 2, 1),\ \|\mathbf{b}_1^*\|^2 = 2^2 + 2^2 + 1 = 9$

Step 2 : $\mu_{2,1} = \frac{\langle \mathbf{b}_2, \mathbf{b}_1^* \rangle}{\|\mathbf{b}_1^*\|^2} = -\frac{4}{9}$

$\mathbf{b}_2^* := \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1^* = (3, 0, 2) + \frac{4}{9}(-2, 2, 1) = \left(\frac{19}{9}, \frac{8}{9}, \frac{22}{9}\right)$

Step 3 : $\mu_{3,1} = 0$, $\mu_{3,2} = \frac{54}{101}, \mathbf{b}_3^* = \left(\frac{88}{101}, \frac{154}{101}, -\frac{132}{101}\right)$

$$\overbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}}^{B} = \overbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & \frac{54}{101} & 1 \end{pmatrix}}^{U} \times \overbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}^{B^*}$$

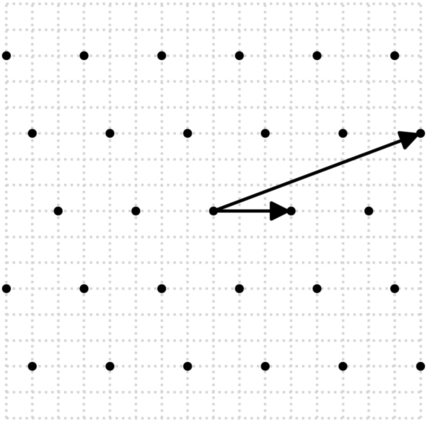
Size Reduction of a Basis

Size Reduction of a Basis

Problem: The Gram–Schmidt orthogonal basis of B is generally not a basis of the lattice $\mathscr{L}(B)$.

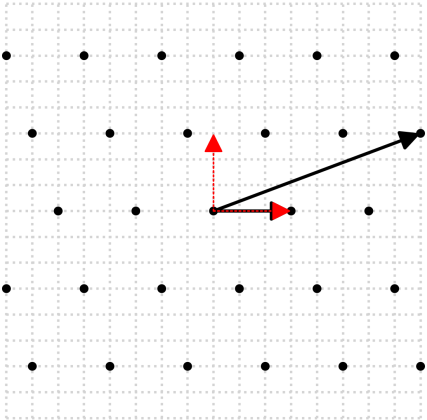
Size Reduction of a Basis

Problem: The Gram–Schmidt orthogonal basis of B is generally not a basis of the lattice $\mathscr{L}(B)$.



Size Reduction of a Basis

Problem: The Gram–Schmidt orthogonal basis of B is generally not a basis of the lattice $\mathscr{L}(B)$.



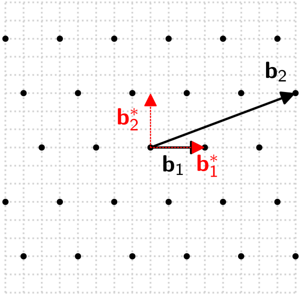
Size Reduction of a Basis

Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:

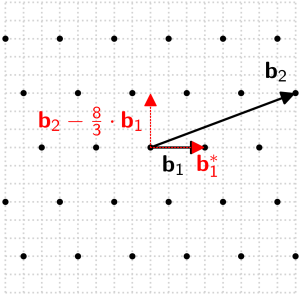
Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:



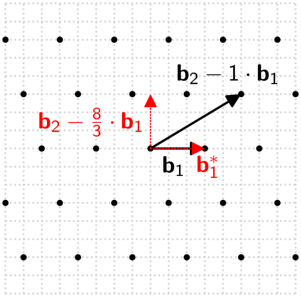
Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:



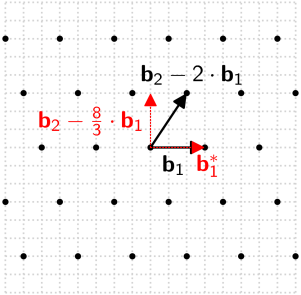
Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:


$\mathbf{b}_2 - 1 \cdot \mathbf{b}_1$
$\mathbf{b}_2 - \frac{8}{3} \cdot \mathbf{b}_1$
$\mathbf{b}_1$
$\mathbf{b}_1^*$

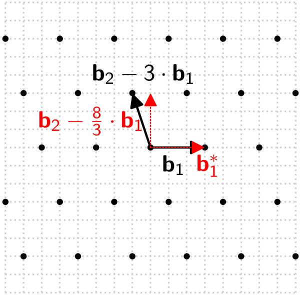
Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:


$\mathbf{b}_2 - 2 \cdot \mathbf{b}_1$
$\mathbf{b}_2 - \frac{8}{3} \cdot \mathbf{b}_1$
$\mathbf{b}_1$
$\mathbf{b}_1^*$

Size Reduction of a Basis

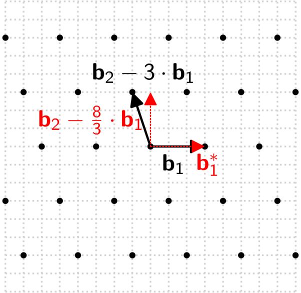
We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:



$\mathbf{b}_2 - 3 \cdot \mathbf{b}_1$
$\mathbf{b}_2 - \frac{8}{3} \cdot \mathbf{b}_1$
$\mathbf{b}_1$
$\mathbf{b}_1^*$

Size Reduction of a Basis

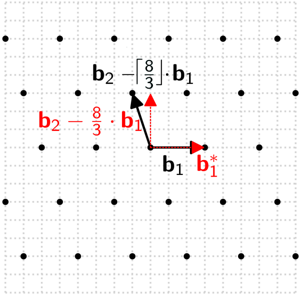
We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:


$\mathbf{b}_2 - 3 \cdot \mathbf{b}_1$
$\mathbf{b}_2 - \frac{8}{3} \cdot \mathbf{b}_1$
$\mathbf{b}_1$
$\mathbf{b}_1^*$

We define the **nearest integer**, as $\lceil x \rfloor := \left\lfloor x + \frac{1}{2} \right\rfloor$.

Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:

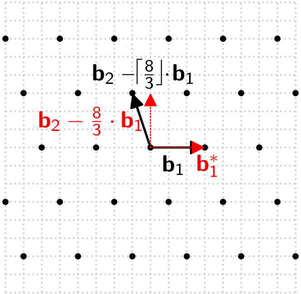


$\mathbf{b}_2 - \lceil \frac{8}{3} \rfloor \cdot \mathbf{b}_1$
$\mathbf{b}_2 - \frac{8}{3} \cdot \mathbf{b}_1$
$\mathbf{b}_1$
$\mathbf{b}_1^*$

We define the **nearest integer**, as $\lceil x \rfloor := \left\lfloor x + \frac{1}{2} \right\rfloor$.

Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:



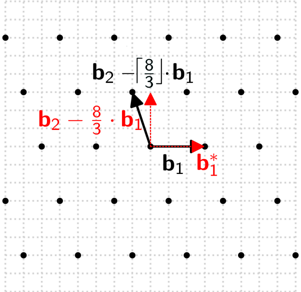
$\mathbf{b}_2 - \lceil \frac{8}{3} \rfloor \cdot \mathbf{b}_1$
$\mathbf{b}_2 - \frac{8}{3} \cdot \mathbf{b}_1$
$\mathbf{b}_1$
$\mathbf{b}_1^*$

$|\lceil x \rfloor - x| \leq \frac{1}{2}$ for all $x \in \mathbb{R}$

We define the **nearest integer**, as $\lceil x \rfloor := \left\lfloor x + \frac{1}{2} \right\rfloor$.

Size Reduction of a Basis

We want a basis of $\mathscr{L}$ that *approximates* the Gram–Schmidt basis as closely as possible:


$\mathbf{b}_2 - \lceil \frac{8}{3} \rfloor \cdot \mathbf{b}_1$
$\mathbf{b}_2 - \frac{8}{3} \cdot \mathbf{b}_1$
$\mathbf{b}_1$
$\mathbf{b}_1^*$

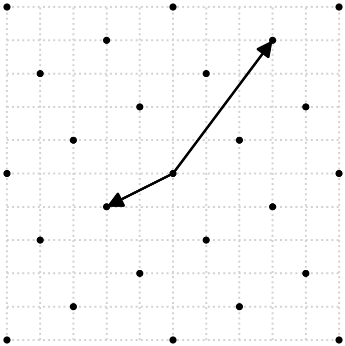
$$|\lceil x \rfloor - x| \leq \frac{1}{2} \text{ for all } x \in \mathbb{R}$$

We define the **nearest integer**, as $\lceil x \rfloor := \left\lfloor x + \frac{1}{2} \right\rfloor$.

Definition: A basis is said to be **size-reduced** if:

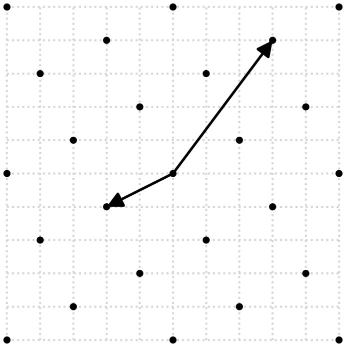
$$\max_{1 \leq j < i \leq n} |\mu_{i,j}| \leq \frac{1}{2}$$

Why Size Reduction is Not Enough



A size-reduced basis.

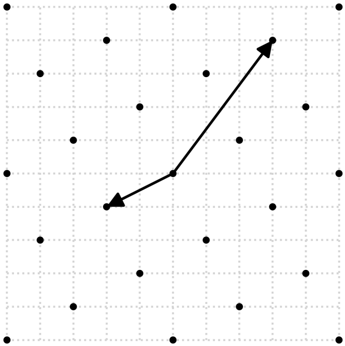
Why Size Reduction is Not Enough



A size-reduced basis.

$$\overbrace{\begin{pmatrix} 3 & 4 \\ -2 & -1 \end{pmatrix}}^{B} = \overbrace{\begin{pmatrix} 1 & 0 \\ -\frac{2}{5} & 1 \end{pmatrix}}^{U} \cdot \overbrace{\begin{pmatrix} 3 & 4 \\ -\frac{4}{5} & \frac{3}{5} \end{pmatrix}}^{B^*}$$

Why Size Reduction is Not Enough



A size-reduced basis.

$$\overbrace{\begin{pmatrix} 3 & 4 \\ -2 & -1 \end{pmatrix}}^{B} = \overbrace{\begin{pmatrix} 1 & 0 \\ -\frac{2}{5} & 1 \end{pmatrix}}^{U} \cdot \overbrace{\begin{pmatrix} 3 & 4 \\ -\frac{4}{5} & \frac{3}{5} \end{pmatrix}}^{B^*}$$

Length reduction alone ***does not imply*** *almost-orthogonality!*

Definition: Lovász condition

Definition: Lovász condition

Ideally, we would like to find a basis $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \ldots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

Definition: Lovász condition

Ideally, we would like to find a basis $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \dots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

Definition: Lovász condition

Ideally, we would like to find a basis $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \ldots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1 \leq i \leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$

Definition: Lovász condition

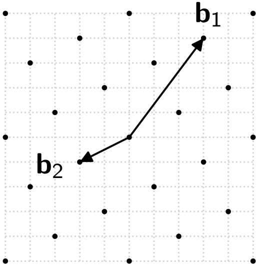
Ideally, we would like to find a basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \ldots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1\leq i\leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$



Definition: Lovász condition

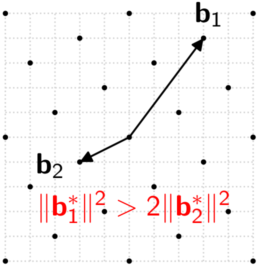
Ideally, we would like to find a basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \dots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1\leq i\leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$



Definition: Lovász condition

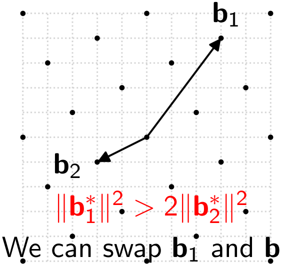
Ideally, we would like to find a basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \ldots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1\leq i\leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$



$$\|\mathbf{b}_1^*\|^2 > 2\|\mathbf{b}_2^*\|^2$$

We can swap $\mathbf{b}_1$ and $\mathbf{b}_2$.

Definition: Lovász condition

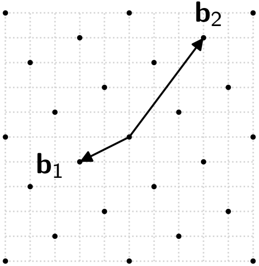
Ideally, we would like to find a basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \dots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1\leq i\leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$



Definition: Lovász condition

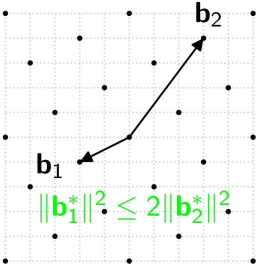
Ideally, we would like to find a basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \dots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1\leq i\leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$



Definition: Lovász condition

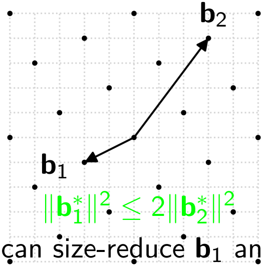
Ideally, we would like to find a basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \ldots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1\leq i\leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$



We can size-reduce $\mathbf{b}_1$ and $\mathbf{b}_2$!

Definition: Lovász condition

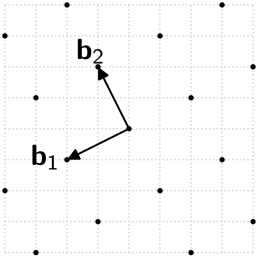
Ideally, we would like to find a basis $(\mathbf{b}_i)_{1\leq i\leq n}$ of the lattice $\mathscr{L}$ such that:

$$\|\mathbf{b}_1\| = \lambda_1(\mathscr{L}), \quad \|\mathbf{b}_2\| = \lambda_2(\mathscr{L}), \quad \dots, \quad \|\mathbf{b}_n\| = \lambda_n(\mathscr{L})$$

This would imply $\|\mathbf{b}_1\| \leq \cdots \leq \|\mathbf{b}_n\|$, but is it hard to find a such basis.

A basis $(\mathbf{b}_i)_{1\leq i\leq m}$ satisfies the **original Lovász condition** if:

$$\|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \quad \text{for all } 1 \leq i < n$$



Definition: LLL– reduced Basis

A basis is called LLL–reduced if:

- It is size-reduced;
- It satisfies the Lovász condition.

Recap:The $\gamma - \mathrm{SVP}$ Problem

Definitions of λ_1, λ_2, ... are detailed in (Boudgoust 2023).

Recap:The $\gamma - \mathrm{SVP}$ Problem

Definitions of λ_1, λ_2, ... are detailed in (Boudgoust 2023).

Approximate Shortest Vector Problem ($\gamma - \mathrm{SVP}$)

Given a basis B of a lattice $\mathscr{L} \subset \mathbb{R}^n$ and an approximation factor $\gamma > 0$, **find a non-zero vector $\mathbf{v}$** $\in \mathscr{L} \setminus \{\mathbf{0}\}$ such that:

$$\|\mathbf{v}\|_2 \leq \gamma \cdot \lambda_1(\mathscr{L})$$

Recap:The $\gamma - \mathrm{SVP}$ Problem

Definitions of λ_1, λ_2, ... are detailed in (Boudgoust 2023).

Approximate Shortest Vector Problem ($\gamma - \mathrm{SVP}$)

Given a basis B of a lattice $\mathscr{L} \subset \mathbb{R}^n$ and an approximation factor $\gamma > 0$, **find a non-zero vector $\mathbf{v}$** $\in \mathscr{L} \setminus \{\mathbf{0}\}$ such that:

$$\|\mathbf{v}\|_2 \leq \gamma \cdot \lambda_1(\mathscr{L})$$

$\gamma = 1$	exact SVP — NP-hard
$\gamma = \mathrm{poly}(n)$	relevant for **lattice-based cryptography**
$\gamma = 2^{\mathcal{O}(n)}$	solvable in **polynomial time** via LLL

Lemma

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \sum_{i=1}^{k} \lambda_i \mathbf{b}_i$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \sum_{i=1}^{k} \lambda_i \left(\mathbf{b}_i^* + \sum_{j=1}^{i} \mu_{ij} \mathbf{b}_j^* \right)$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \sum_{i=1}^{k} \lambda_i \mathbf{b}_i^* + \lambda_i \sum_{j=1}^{i} \mu_{ij} \mathbf{b}_j^*$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \lambda_k \mathbf{b}_k^* + \sum_{i<k} \lambda_i \sum_{j=1}^{i} \mu_{ij} \mathbf{b}_j^*$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \lambda_k \mathbf{b}_k^* + \sum_{i<k} \nu_i \mathbf{b}_i^*, \quad \nu_i \in \mathbb{R}$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \lambda_k \mathbf{b}_k^* + \sum_{i<k} \nu_i \mathbf{b}_i^*, \quad \nu_i \in \mathbb{R}$$

Hence

$$\|\mathbf{b}\|^2 = \lambda_k^2 \|\mathbf{b}_k^*\|^2 + \sum_{i<k} \nu_i^2 \|\mathbf{b}_i^*\|^2$$

$$\geq \lambda_k^2 \|\mathbf{b}_k^*\|^2 \geq \|\mathbf{b}_k^*\|^2$$

Lemma

Lemma. For any $\mathbf{b} \in \mathscr{L} \setminus \{0\}$ we have:

$$\|\mathbf{b}\| \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Proof. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ of the lattice $\mathscr{L}$, and write:

$$\mathbf{b} = \sum_{i=1}^{n} \lambda_i \mathbf{b}_i \in \mathscr{L} \setminus \{0\}, \quad \lambda_i \in \mathbb{Z}.$$

Let k be the largest index such that $\lambda_k \neq 0$. We can write

$$\mathbf{b} = \lambda_k \mathbf{b}_k^* + \sum_{i<k} \nu_i \mathbf{b}_i^*, \quad \nu_i \in \mathbb{R}$$

Hence

$$\|\mathbf{b}\|^2 = \lambda_k^2 \|\mathbf{b}_k^*\|^2 + \sum_{i<k} \nu_i^2 \|\mathbf{b}_i^*\|^2$$

$$\geq \lambda_k^2 \|\mathbf{b}_k^*\|^2 \geq \|\mathbf{b}_k^*\|^2 \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Theorem: Bound on First Vector in a Reduced Basis

Theorem: Bound on First Vector in a Reduced Basis

Theorem. Let $(\mathbf{b}_i)_{1\leq i\leq n}$ be a reduced basis of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, and let $\mathbf{b} \in \mathscr{L} \setminus \{0\}$. Then:

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \|\mathbf{b}\|.$$

Theorem: Bound on First Vector in a Reduced Basis

Theorem. Let $(\mathbf{b}_i)_{1\leq i\leq n}$ be a reduced basis of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, and let $\mathbf{b} \in \mathscr{L} \setminus \{0\}$. Then:

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \|\mathbf{b}\|.$$

Proof.

$$\|\mathbf{b}_1\|^2 = \|\mathbf{b}_1^*\|^2 \leq 2\|\mathbf{b}_2^*\|^2 \leq 2^2\|\mathbf{b}_3^*\|^2 \leq \cdots \leq 2^{n-1}\|\mathbf{b}_n^*\|^2.$$

Thus,

$$\|\mathbf{b}\| \geq \min\{\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_n^*\|\} \geq 2^{-(n-1)/2}\|\mathbf{b}_1\|$$

□

Theorem: Bound on First Vector in a Reduced Basis

Theorem. Let $(\mathbf{b}_i)_{1 \leq i \leq n}$ be a reduced basis of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, and let $\mathbf{b} \in \mathscr{L} \setminus \{0\}$. Then:

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \|\mathbf{b}\|.$$

Theorem: Bound on First Vector in a Reduced Basis

Theorem. Let $(\mathbf{b}_i)_{1\leq i\leq n}$ be a reduced basis of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, and let $\mathbf{b} \in \mathscr{L} \setminus \{0\}$. Then:

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \|\mathbf{b}\|.$$

Corollary.

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \lambda_1(\mathscr{L}).$$

Theorem: Bound on First Vector in a Reduced Basis

Theorem. Let $(\mathbf{b}_i)_{1\leq i\leq n}$ be a reduced basis of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, and let $\mathbf{b} \in \mathscr{L} \setminus \{0\}$. Then:

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \|\mathbf{b}\|.$$

Corollary.

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \lambda_1(\mathscr{L}).$$

Interpretation. The vector $\mathbf{b}_1$ of a reduced basis **solves** $2^{(n-1)/2} - \mathrm{SVP}$.

Theorem: Bound on First Vector in a Reduced Basis

Theorem. Let $(\mathbf{b}_i)_{1\leq i\leq n}$ be a reduced basis of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, and let $\mathbf{b} \in \mathscr{L} \setminus \{0\}$. Then:

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \|\mathbf{b}\|.$$

Corollary.

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \lambda_1(\mathscr{L}).$$

Interpretation. The vector $\mathbf{b}_1$ of a reduced basis **solves** $2^{(n-1)/2} - \mathrm{SVP}$.

How can we compute a reduced basis in practice?

Theorem: Bound on First Vector in a Reduced Basis

Theorem. Let $(\mathbf{b}_i)_{1\leq i\leq n}$ be a reduced basis of a lattice $\mathscr{L} \subseteq \mathbb{R}^n$, and let $\mathbf{b} \in \mathscr{L} \setminus \{0\}$. Then:

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \|\mathbf{b}\|.$$

Corollary.

$$\|\mathbf{b}_1\| \leq 2^{(n-1)/2} \cdot \lambda_1(\mathscr{L}).$$

Interpretation. The vector $\mathbf{b}_1$ of a reduced basis **solves** $2^{(n-1)/2} - \mathrm{SVP}$.

How can we compute a reduced basis in practice?

→ Use the LLL (Lenstra 1982)(Lenstra, Lenstra, Lovasz) algorithm!

LLL Algorithm

LLL Algorithm

```
Algorithm 0: LLL

Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \dots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$(G^*, U) \leftarrow$ GRAM-SCHMIDT $G$
while $i \leq n$ do
    for $j = i-1, i-2, \dots, 1$ do
        $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$
    if  $i > 1$ and $\|bg_{i-1}^*\|^2 > 2\|bg_i^*\|^2$ then
        Swap
        $\mathrm{bg}_{i-1}$ and
        $\mathrm{bg}_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

LLL Algorithm

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \ldots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$   Gram-Schmidt
while $i \leq n$ do
    for $j = i-1, i-2, \ldots, 1$ do
        $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$
    if $i > 1$ and $\|bg^*_{i-1}\|^2 > 2\|bg^*_i\|^2$ then
        Swap
        $\mathrm{bg}_{i-1}$ and
        $\mathrm{bg}_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

LLL Algorithm

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = (\mathrm{bg}_1, \dots, \mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$    Gram-Schmidt
while $i \leq n$ do
    for $j = i-1, i-2, \dots, 1$ do
        $\boxed{\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor\, \mathrm{bg}_j,\ \text{update}\ (G^*, U)}$    Size Reduction
    if $i > 1$ and $\|bg^*_{i-1}\|^2 > 2\|bg^*_i\|^2$ then
        Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

LLL Algorithm

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
bg$_1, \dots,$
bg$_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$    Gram-Schmidt
while $i \leq n$ do
    for $j = i-1, i-2, \dots, 1$ do
        $\boxed{\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor\, \mathrm{bg}_j, \text{ update } (G^*, U)}$    Size Reduction
    if $i > 1$ and $\boxed{\|bg^*_{i-1}\|^2 > 2\|bg^*_i\|^2}$ then    Lovász Condition
        Swap
        bg$_{i-1}$ and
        bg$_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

Example

Example

Let's compute a LLL reduced basis of $\mathscr{L}(B)$ with

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

Example

Let's compute a LLL reduced basis of $\mathscr{L}(B)$ with

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

We start by compute its Gram-Schmidt decomposition :

Example

Let's compute a LLL reduced basis of $\mathscr{L}(B)$ with

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

We start by compute its Gram-Schmidt decomposition :

We did it previously!

Example

Let's compute a LLL reduced basis of $\mathscr{L}(B)$ with

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

We start by compute its Gram-Schmidt decomposition :

We did it previously!

$$\overbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}}^{B} = \overbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & \frac{54}{101} & 1 \end{pmatrix}}^{U} \times \overbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}^{B^*}$$

Example

Example

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & \frac{54}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & \frac{54}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & \frac{54}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ 2 & 2 & 0 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & \frac{54}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ -1 & 2 & -2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & -\frac{47}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ -1 & 2 & -2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & -\frac{47}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ -1 & 2 & -2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & -\frac{47}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -1 & 2 & -2 \\ 3 & 0 & 2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ \frac{4}{9} & 1 & 0 \\ -\frac{4}{9} & -\frac{47}{65} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -\frac{1}{9} & \frac{10}{9} & -\frac{22}{9} \\ \frac{132}{165} & \frac{22}{13} & -\frac{44}{65} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ 3 & 0 & 2 \\ -1 & 2 & -2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{4}{9} & 1 & 0 \\ 0 & -\frac{47}{101} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ \frac{19}{9} & \frac{8}{9} & \frac{22}{9} \\ \frac{88}{101} & \frac{154}{101} & -\frac{132}{101} \end{pmatrix}}_{G^*}$$

Example

Example

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -1 & 2 & -2 \\ 3 & 0 & 2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ \frac{4}{9} & 1 & 0 \\ -\frac{4}{9} & -\frac{47}{65} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -\frac{1}{9} & \frac{10}{9} & -\frac{22}{9} \\ \frac{132}{165} & \frac{22}{13} & -\frac{44}{65} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -1 & 2 & -2 \\ 3 & 0 & 2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ \frac{4}{9} & 1 & 0 \\ -\frac{4}{9} & -\frac{47}{65} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -\frac{1}{9} & \frac{10}{9} & -\frac{22}{9} \\ \frac{132}{165} & \frac{22}{13} & -\frac{44}{65} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -1 & 2 & -2 \\ 3 & 0 & 2 \end{pmatrix}}_{G} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ \frac{4}{9} & 1 & 0 \\ -\frac{4}{9} & -\frac{47}{65} & 1 \end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix} -2 & 2 & 1 \\ -\frac{1}{9} & \frac{10}{9} & -\frac{22}{9} \\ \frac{132}{165} & \frac{22}{13} & -\frac{44}{65} \end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix}-2 & 2 & 1\\ -1 & 2 & -2\\ 3 & 0 & 2\end{pmatrix}}_{G} = \underbrace{\begin{pmatrix}1 & 0 & 0\\ \frac{4}{9} & 1 & 0\\ -\frac{4}{9} & -\frac{47}{65} & 1\end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix}-2 & 2 & 1\\ -\frac{1}{9} & \frac{10}{9} & -\frac{22}{9}\\ \frac{132}{165} & \frac{22}{13} & -\frac{44}{65}\end{pmatrix}}_{G^*}$$

$$\underbrace{\begin{pmatrix}-2 & 2 & 1\\ -1 & 2 & -2\\ 2 & 2 & 0\end{pmatrix}}_{G} = \underbrace{\begin{pmatrix}1 & 0 & 0\\ \frac{4}{9} & 1 & 0\\ 0 & \frac{18}{65} & 1\end{pmatrix}}_{U} \cdot \underbrace{\begin{pmatrix}-2 & 2 & 1\\ -\frac{1}{9} & \frac{10}{9} & -\frac{22}{9}\\ \frac{132}{165} & \frac{22}{13} & -\frac{44}{65}\end{pmatrix}}_{G^*}$$

LLL: Example of a Reduced Basis

LLL: Example of a Reduced Basis

We obtain the following LLL reduced basis:

$$_{\text{reduced}} = \begin{pmatrix} -2 & 2 & 1 \\ -1 & 2 & -2 \\ 2 & 2 & 0 \end{pmatrix}$$

LLL: Example of a Reduced Basis

We obtain the following LLL reduced basis:

$${}_{\text{reduced}} = \begin{pmatrix} -2 & 2 & 1 \\ -1 & 2 & -2 \\ 2 & 2 & 0 \end{pmatrix}$$

The vector $(2, 2, 0)$ is a shortest nonzero vector in the lattice, hence:

$$\lambda_1(\mathscr{L}) = 2\sqrt{2}.$$

LLL Complexity

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \ldots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$(G^*, U) \leftarrow$ GRAM-SCHMIDT $G$
  while $i \leq n$ do
    for $j = i-1, i-2, \ldots, 1$ do
      $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$
      if $i > 1$ and $\|$
    $bg^*_{i-1}\|^2 > 2\|$
    $bg^*_i\|^2$ then
      Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
      $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($ $\mathrm{bg}_1, \ldots,$ $\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$ $\mathcal{O}(n^3)$
  while $i \leq n$ do
      for $j = i-1, i-2, \ldots, 1$ do
          $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$
      if $i > 1$ and $\| bg^*_{i-1}\|^2 > 2\| bg^*_i\|^2$ then
          Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
          $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = (\mathrm{bg}_1, \ldots, \mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$ $\mathcal{O}(n^3)$
while $i \leq n$ do
    for $j = i-1, i-2, \ldots, 1$ do
        $\boxed{\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j, \text{ update } (G^*, U)}$ $\mathcal{O}(n)$
    if $i > 1$ and $\|bg^*_{i-1}\|^2 > 2\|bg^*_i\|^2$ then
        Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

LLL Complexity

```
Algorithm 0: LLL

Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = (\mathrm{bg}_1, \dots, \mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$   $\mathcal{O}(n^3)$
  while $i \leq n$ do
    for $j = i-1, i-2, \dots, 1$ do
        $\boxed{\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j, \text{ update } (G^*, U)}$   $\mathcal{O}(n)$   $\mathcal{O}(n^2)$
    if $i > 1$ and $\| bg^*_{i-1} \|^2 > 2 \| bg^*_i \|^2$ then
        Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

LLL Complexity

```
Algorithm 0: LLL
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \ldots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$ $\mathcal{O}(n^3)$
   while $i \le n$ do
      $\boxed{\text{for } j = i-1, i-2, \ldots, 1 \text{ do}$
         $\boxed{\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j, \text{ update } (G^*, U)}$ $\mathcal{O}(n)}$ $\mathcal{O}(n^2)$
            if $i > 1$ and $\|$
      $bg^*_{i-1}\|^2 > 2\|$
      $bg^*_i\|^2$ then
         $\boxed{\text{Swap } \mathrm{bg}_{i-1} \text{ and } \mathrm{bg}_i, \text{ update } (G^*, U)}$ $\mathcal{O}(n)$
         $i \leftarrow i - 1$
```

LLL Complexity

```
Algorithm 0: LLL

Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \dots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$(G^*, U) \leftarrow$ Gram-Schmidt $G$   $\mathcal{O}(n^3)$
while $i \le n$ do
    for $j = i-1, i-2, \dots, 1$ do
        $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$   $\mathcal{O}(n)$   $\mathcal{O}(n^2)$
    if $i > 1$ and $\|$
    $bg^*_{i-1}\|^2 > 2\|$
    $bg^*_i\|^2$ then
        Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$   $\mathcal{O}(n)$
        $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \ldots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$(G^*, U) \leftarrow$ Gram-Schmidt $G$   $\mathcal{O}(n^3)$
while $i \leq n$ do   How much?
    for $j = i-1, i-2, \ldots, 1$ do   $\mathcal{O}(n^2)$
        $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$   $\mathcal{O}(n)$
    if $i > 1$ and $\|$
    $bg^*_{i-1}\|^2 > 2\|$
    $bg^*_i\|^2$ then
        Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$   $\mathcal{O}(n)$
        $i \leftarrow i - 1$
```

Correctness

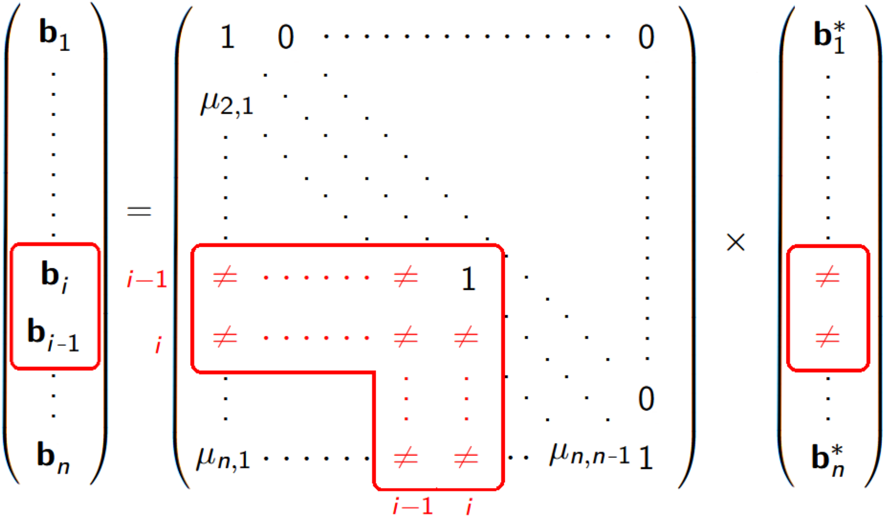
Key idea: Clearly, if the algorithm LLL **terminates**, the returned basis is by construction LLL-reduced.

Therefore, it remains ***to prove*** *that* LLL ***always terminates****.*

How can we prove the termination of the algorithm?

$$\begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{i-1} \\ \mathbf{b}_i \\ \vdots \\ \mathbf{b}_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \mu_{2,1} & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \mu_{i\text{-}1,1} & \cdots & \mu_{i\text{-}1,i\text{-}2} & \ddots & \ddots & \vdots \\ \mu_{i,1} & \cdots & \mu_{i,i\text{-}2} & \mu_{i,i\text{-}1} & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ \mu_{n,1} & \cdots & \cdots & \cdots & \mu_{n,n\text{-}1} & 1 \end{pmatrix} \times \begin{pmatrix} \mathbf{b}_1^* \\ \vdots \\ \mathbf{b}_{i-1}^* \\ \mathbf{b}_i^* \\ \vdots \\ \mathbf{b}_n^* \end{pmatrix}$$

How can we prove the termination of the algorithm?



How can we prove the termination of the algorithm?

$$\begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \vdots \\ \mathbf{b}_i \\ \mathbf{b}_{i-1} \\ \vdots \\ \mathbf{b}_n \end{pmatrix} = \begin{array}{c|ccccccc|} & 1 & 0 & \cdots & & & & 0 \\ & \mu_{2,1} & \ddots & \ddots & & & & \vdots \\ & \vdots & & \ddots & \ddots & & & \vdots \\ i-1 & \neq & \cdots & \neq & 1 & \ddots & & \vdots \\ i & \neq & \cdots & \neq & \neq & \ddots & \ddots & \vdots \\ & \vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\ & \mu_{n,1} & \cdots & \neq & \neq & \cdots & \mu_{n,n-1} & 1 \\ & & & i-1 & i & & & \end{array} \times \begin{pmatrix} \mathbf{b}_1^* \\ \vdots \\ \vdots \\ \neq \\ \neq \\ \vdots \\ \mathbf{b}_n^* \end{pmatrix}$$

How can we prove the termination of the algorithm?

$$
\begin{pmatrix} \|\mathbf{b}_1\|^2 \\ \vdots \\ \vdots \\ \|\mathbf{b}_i\|^2 \\ \|\mathbf{b}_{i-1}\|^2 \\ \vdots \\ \|\mathbf{b}_n\|^2 \end{pmatrix}
=
\begin{array}{cc}
\begin{array}{r} \\ \\ \\ i-1 \\ i \\ \\ \\ \end{array} &
\begin{pmatrix}
1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
\mu_{2,1}^2 & \ddots & \ddots & & & & \vdots \\
\vdots & \ddots & \ddots & \ddots & & & \vdots \\
\neq^2 & \cdots & \neq^2 & 1 & \ddots & & \vdots \\
\neq^2 & \cdots & \neq^2 & \neq^2 & \ddots & \ddots & \vdots \\
\vdots & & \vdots & \vdots & \ddots & \ddots & 0 \\
\mu_{n,1}^2 & \cdots & \neq^2 & \neq^2 & \cdots & \mu_{n,n-1}^2 & 1
\end{pmatrix} \\
 & \begin{array}{ccccccc} \quad & \quad & i-1 & i & \quad & \quad & \quad \end{array}
\end{array}
\times
\begin{pmatrix} \|\mathbf{b}_1^*\|^2 \\ \vdots \\ \vdots \\ \|\neq\|^2 \\ \|\neq\|^2 \\ \vdots \\ \|\mathbf{b}_n^*\|^2 \end{pmatrix}
$$

How can we prove the termination of the algorithm?

$$
\begin{pmatrix}
\|\mathbf{b}_1\|^2 \\ \vdots \\ \vdots \\ \|\mathbf{b}_i\|^2 \\ \|\mathbf{b}_{i-1}\|^2 \\ \vdots \\ \|\mathbf{b}_n\|^2
\end{pmatrix}
=
\begin{array}{c}
\\ \\ \\ i-1 \\ i \\ \\ \\ \\
\end{array}
\begin{array}{cccccc}
1 & 0 & \cdots & \cdots & \cdots & 0 \\
\mu_{2,1}^2 & \ddots & & & & \vdots \\
\vdots & & \ddots & & & \vdots \\
\neq^2 & \cdots & \neq^2 & 1 & \ddots & \vdots \\
\neq^2 & \cdots & \neq^2 & \neq^2 & \ddots & \vdots \\
\vdots & & \vdots & \vdots & \ddots & 0 \\
\mu_{n,1}^2 & \cdots & \neq^2 & \neq^2 & \cdots\ \mu_{n,n-1}^2 & 1 \\
 & & i-1 & i & &
\end{array}
\times
\begin{pmatrix}
\|\mathbf{b}_1^*\|^2 \\ \vdots \\ \vdots \\ \leq \frac{3}{4}\|\mathbf{b}_{i-1}^*\|^2 \\ \leq \|\mathbf{b}_{i-1}^*\|^2 \\ \vdots \\ \|\mathbf{b}_n^*\|^2
\end{pmatrix}
$$

How can we prove the termination of the algorithm?

How can we prove the termination of the algorithm?

Let $_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$.

How can we prove the termination of the algorithm?

Let ${}_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det({}_k \cdot {}^t_k)$.

How can we prove the termination of the algorithm?

Let ${}_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det({}_k \cdot {}^t_k)$.

→ will be used to control the progress of the algorithm.

How can we prove the termination of the algorithm?

Let ${}_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det({}_k \cdot {}_k^t)$.

$\rightarrow$ will be used to control the progress of the algorithm.

We have

How can we prove the termination of the algorithm?

Let $_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det(_k \cdot _k^t)$.

→ will be used to control the progress of the algorithm.

We have

$$d_k$$

How can we prove the termination of the algorithm?

Let $_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det(_k \cdot {}^t_k)$.

→ will be used to control the progress of the algorithm.

We have

$$d_k = \det\left({}_k{}^t_k\right)$$

How can we prove the termination of the algorithm?

Let ${}_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det({}_k \cdot {}_k^t)$.

$\rightarrow$ will be used to control the progress of the algorithm.

We have

$$d_k = \det\left({}_k {}_k^t\right) = \det\left(U_k {}_k^* ({}_k^*)^t U_k^t\right)$$

How can we prove the termination of the algorithm?

Let ${}_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det({}_k \cdot {}_k^t)$.

→ will be used to control the progress of the algorithm.

We have

$$d_k = \det\left({}_k{}_k^t\right) = \det\left(U_k{}_k^*({}_k^*)^t U_k^t\right) = \det\left({}_k^*({}_k^*)^t\right)$$

How can we prove the termination of the algorithm?

Let ${}_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det({}_k \cdot {}_k^t)$.

$\rightarrow$ will be used to control the progress of the algorithm.

We have

$$d_k = \det\left({}_k{}_k^t\right) = \det\left(U_k{}_k^*({}_k^*)^t U_k^t\right) = \det\left({}_k^*({}_k^*)^t\right) = \prod_{1\leq l\leq k} \|\mathbf{g}_l^*\|^2$$

How can we prove the termination of the algorithm?

Let ${}_k = \begin{pmatrix} bg_1 \\ \\ bg_2 \\ \vdots \\ \\ bg_n \end{pmatrix}$. We define $d_k := \det({}_k \cdot {}_k^t)$.

→ will be used to control the progress of the algorithm.

We have

$$d_k = \det\left({}_k{}_k^t\right) = \det\left(U_k{}_k^*({}_k^*)^t U_k^t\right) = \det\left({}_k^*({}_k^*)^t\right) = \prod_{1 \le l \le k} \|\mathbf{g}_l^*\|^2$$

If we **swap**
bg$_i$ and
bg$_{i-1}$:
$\|\mathbf{d}_{i-1}^*\|$ decrease by a $\frac{3}{4}$ factor, so $\mathbf{d_{i-1}}$ decrease by a $\frac{3}{4}$ factor.

How can we prove the termination of the algorithm?

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 =$$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k =$$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k = \prod_{k=1}^{n-1} \prod_{1 \leq l \leq k} \|$$

bg$_l^* \|^2 =$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \displaystyle\prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k = \prod_{k=1}^{n-1} \prod_{1 \leq l \leq k} \|$$

bg$_l^* \|^2 = \prod_{k=1}^{n-1} \|$
bg$_k^* \|^{2(n-k)}$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k = \prod_{k=1}^{n-1} \prod_{1 \leq l \leq k} \|$$

bg$_l^*\|^2 = \prod_{k=1}^{n-1} \|$
bg$_k^*\|^{2(n-k)}$

$$\leq \prod_{k=1}^{n-1} \|$$

bg$_k\|^{2(n-k)}$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k = \prod_{k=1}^{n-1} \prod_{1 \leq l \leq k} \|$$

bg$_l^*\|^2 = \prod_{k=1}^{n-1} \|$
bg$_k^*\|^{2(n-k)}$

$$\leq \prod_{k=1}^{n-1} \|$$

bg$_k\|^{2(n-k)} \leq \prod_{k=1}^{n-1} (\max_{1 \leq i \leq n} \|$
bg$_i\|^{2(n-k)}$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

→ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k = \prod_{k=1}^{n-1} \prod_{1 \leq l \leq k} \|$$

bg$_l^* \|^2 = \prod_{k=1}^{n-1} \|$
bg$_k^* \|^{2(n-k)}$

$$\leq \prod_{k=1}^{n-1} \|$$

bg$_k \|^{2(n-k)} \leq \prod_{k=1}^{n-1} (\max_{1 \leq i \leq n} \|$
bg$_i \|^{2(n-k)} \leq (\max_{1 \leq i \leq n} \|$
bg$_i \|^{n(n-1)}$

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \displaystyle\prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k = \prod_{k=1}^{n-1} \prod_{1 \leq l \leq k} \|$$

bg$_l^*\|^2 = \prod_{k=1}^{n-1} \|$
bg$_k^*\|^{2(n-k)}$

$$\leq \prod_{k=1}^{n-1} \|$$

bg$_k\|^{2(n-k)} \leq \prod_{k=1}^{n-1} (\max_{1 \leq i \leq n} \|$
bg$_i\|^{2(n-k)} \leq (\max_{1 \leq i \leq n} \|$
bg$_i\|^{n(n-1)}$

Termination proof

How can we prove the termination of the algorithm?

We define $\mathbb{Z} \ni D := \prod_{k=1}^{n-1} d_k > 1$

$\rightarrow$ After each swap, D decrease by a $\frac{3}{4}$ factor.

Let D_0 be the value of D a the start of LLL, we have

$$D_0 = \prod_{k=1}^{n-1} d_k = \prod_{k=1}^{n-1} \prod_{1 \leq l \leq k} \|$$

bg$_l^*\|^2 = \prod_{k=1}^{n-1} \|$
bg$_k^*\|^{2(n-k)}$

$$\leq \prod_{k=1}^{n-1} \|$$

bg$_k\|^{2(n-k)} \leq \prod_{k=1}^{n-1} (\max_{1 \leq i \leq n} \|$
bg$_i\|^{2(n-k)} \leq (\max_{1 \leq i \leq n} \|$
bg$_i\|^{n(n-1)}$

Termination proof

LLL Complexity

LLL Complexity

Algorithm 0: *LLL*

Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \ldots,$
$\mathrm{bg}_n)$

```
$G \leftarrow copy(B)$
$(G^*, U) \leftarrow$ GRAM-SCHMIDT $G$
  while $i \leq n$ do
      for $j = i-1, i-2, \ldots, 1$ do
          $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$
        if  $i > 1$ and $\|$
    $bg^*_{i-1}\|^2 > 2\|$
    $bg^*_i\|^2$ then
          Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
          $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \dots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$ $\mathcal{O}(n^3)$
  while $i \leq n$ do
      for $j = i-1, i-2, \dots, 1$ do
          $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$
      if $i > 1$ and $\|$
      $bg^*_{i-1}\|^2 > 2\|$
      $bg^*_i\|^2$ then
          Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
          $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
bg$_1, \dots,$
bg$_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$ $\mathcal{O}(n^3)$
while $i \le n$ do
    for $j = i-1, i-2, \dots, 1$ do
        $\boxed{\text{bg}_i \leftarrow \text{bg}_i - \lceil \mu_{i,j} \rfloor \text{bg}_j, \text{ update } (G^*, U)}$ $\mathcal{O}(n)$
        if $i > 1$ and $\|$
    $bg^*_{i-1}\|^2 > 2\|$
    $bg^*_i\|^2$ then
        Swap bg$_{i-1}$ and bg$_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ( \mathrm{bg}_1, \ldots, \mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$ $\mathcal{O}(n^3)$
while $i \leq n$ do
    $\boxed{\text{for } j = i-1, i-2, \ldots, 1 \text{ do}}$
        $\boxed{\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j, \text{ update } (G^*, U)}$ $\mathcal{O}(n)$ $\mathcal{O}(n^2)$
    if $i > 1$ and $\| bg^*_{i-1} \|^2 > 2 \| bg^*_i \|^2$ then
        Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$
        $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = (\mathrm{bg}_1, \dots, \mathrm{bg}_n)$
$G \leftarrow copy(B)$
$\boxed{(G^*, U) \leftarrow \textsc{Gram-Schmidt}\ G}$ $\mathcal{O}(n^3)$
while $i \leq n$ do
    for $j = i-1, i-2, \dots, 1$ do
        $\boxed{\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j, \text{ update } (G^*, U)}$ $\mathcal{O}(n)$   $\mathcal{O}(n^2)$
    if $i > 1$ and $\| bg^*_{i-1} \|^2 > 2 \| bg^*_i \|^2$ then
        $\boxed{\text{Swap } \mathrm{bg}_{i-1} \text{ and } \mathrm{bg}_i, \text{ update } (G^*, U)}$ $\mathcal{O}(n)$
        $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
bg$_1, \ldots,$
bg$_n)$
$G \leftarrow copy(B)$
$(G^*, U) \leftarrow$ Gram-Schmidt $G$    $\mathcal{O}(n^3)$
while $i \leq n$ do
    for $j = i-1, i-2, \ldots, 1$ do
        bg$_i \leftarrow$bg$_i - \lceil \mu_{i,j} \rfloor$ bg$_j$, update $(G^*, U)$    $\mathcal{O}(n)$    $\mathcal{O}(n^2)$
    if $i > 1$ and $\|$
    $bg^*_{i-1}\|^2 > 2\|$
    $bg^*_i\|^2$ then
        Swap bg$_{i-1}$ and bg$_i$, update $(G^*, U)$    $\mathcal{O}(n)$
        $i \leftarrow i - 1$
```

LLL Complexity

Algorithm 0: *LLL*

```
Input: A basis $B = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$
Output: An LLL-reduced basis $G = ($
$\mathrm{bg}_1, \ldots,$
$\mathrm{bg}_n)$
$G \leftarrow copy(B)$
$(G^*, U) \leftarrow$ GRAM-SCHMIDT $G$   $\mathcal{O}(n^3)$
  while $i \leq n$ do   $\mathcal{O}(n^2 log(A))$
      for $j = i - 1, i - 2, \ldots, 1$ do
          $\mathrm{bg}_i \leftarrow \mathrm{bg}_i - \lceil \mu_{i,j} \rfloor \mathrm{bg}_j$, update $(G^*, U)$   $\mathcal{O}(n)$   $\mathcal{O}(n^2)$
      if $i > 1$ and $\|$
      $bg^*_{i-1}\|^2 > 2\|$
      $bg^*_i\|^2$ then
          Swap $\mathrm{bg}_{i-1}$ and $\mathrm{bg}_i$, update $(G^*, U)$   $\mathcal{O}(n)$
          $i \leftarrow i - 1$
```

Theorem: Complexity of BasisReduction

Theorem: Complexity of BasisReduction

Theorem.

Theorem: Complexity of BasisReduction

Theorem.

- LLL uses $\mathcal{O}\left(n^2 \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ loop iterations.

Theorem: Complexity of BasisReduction

Theorem.

- LLL uses $\mathcal{O}\left(n^2 \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ loop iterations.
- LLL uses $\mathcal{O}\left(n^2\right)$ arithmetic operations over rationals per iteration.

Theorem: Complexity of BasisReduction

Theorem.

- LLL uses $\mathcal{O}\left(n^2 \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ loop iterations.
- LLL uses $\mathcal{O}\left(n^2\right)$ arithmetic operations over rationals per iteration.
- U represented with rationals of bit-lengths $\mathcal{O}\left(n \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$

Theorem: Complexity of BasisReduction

Theorem.

- LLL uses $\mathcal{O}\left(n^2 \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ loop iterations.
- LLL uses $\mathcal{O}\left(n^2\right)$ arithmetic operations over rationals per iteration.
- U represented with rationals of bit-lengths $\mathcal{O}\left(n \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$

$\Rightarrow \mathrm{LLL}$ uses $\widetilde{\mathcal{O}}\left(n^5 \log^2\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ bit operations.

Theorem: Complexity of BasisReduction

Theorem.

- LLL uses $\mathcal{O}\left(n^2 \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ loop iterations.
- LLL uses $\mathcal{O}\left(n^2\right)$ arithmetic operations over rationals per iteration.
- U represented with rationals of bit-lengths $\mathcal{O}\left(n \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$

$\Rightarrow$ LLL uses $\widetilde{\mathcal{O}}\left(n^5 \log^2\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ bit operations.

Theorem.

Theorem: Complexity of BasisReduction

Theorem.

- LLL uses $\mathcal{O}\left(n^2 \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ loop iterations.
- LLL uses $\mathcal{O}\left(n^2\right)$ arithmetic operations over rationals per iteration.
- U represented with rationals of bit-lengths $\mathcal{O}\left(n \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$

$\Rightarrow$ LLL uses $\widetilde{\mathcal{O}}\left(n^5 \log^2\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ bit operations.

Theorem.

$\to$ LLL **compute** a reduced basis in **polynomial time**.

Theorem: Complexity of BasisReduction

Theorem.

- LLL uses $\mathcal{O}\left(n^2 \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ loop iterations.
- LLL uses $\mathcal{O}\left(n^2\right)$ arithmetic operations over rationals per iteration.
- U represented with rationals of bit-lengths $\mathcal{O}\left(n \log\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$

$\Rightarrow$ LLL uses $\widetilde{\mathcal{O}}\left(n^5 \log^2\left(\max_{1\leq i\leq n} \|\mathbf{b}_i\|\right)\right)$ bit operations.

Theorem.

$\rightarrow$ LLL **compute** a reduced basis in **polynomial time**.

$\rightarrow$ LLL **solve** $2^{\mathcal{O}(n)} - \mathrm{SVP}$ in **polynomial time**.

Thank you for your attention!

Questions?

Bibliography

Boudgoust, Katharina (Feb. 2023). *Hardness Assumptions in Lattice-Based Cryptography*. Crash-Course lecture notes, Aarhus University. Version du 2 février 2023.

Lenstra Lenstra, Lovász (Dec. 1982). “Factoring polynomials with rational coefficients”. In: *Mathematische Annalen* 261.4, pp. 515–534. ISSN: 1432-1807. DOI: 10.1007/bf01457454. URL: http://dx.doi.org/10.1007/BF01457454.