Lecture 11: Finding Short Vectors in a Lattice

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science
rafael.oliveira.teaching@gmail.com

February 22, 2021
Overview

- Short Vectors in a Lattice
- Algorithm Idea: Find Good Basis
- Gram-Schmidt Orthogonalization
- Lenstra-Lenstra-Lovasz (LLL) Basis Reduction Algorithm
- Conclusion
- Acknowledgements
Short Vectors in a Lattice

- **Input:** linearly independent vectors $b_1, \ldots, b_n \in \mathbb{Z}^n$, bound $M \in \mathbb{N}$

  \[ \mathcal{L} = \{ \alpha_1 b_1 + \cdots + \alpha_n b_n \mid \alpha_i \in \mathbb{Z} \} \]

- **Output:** A vector $v \in \mathcal{L}$ such that $\|v\| \leq M$
Short Vectors in a Lattice

- **Input:** linearly independent vectors $b_1, \ldots, b_n \in \mathbb{Z}^n$, bound $M \in \mathbb{N}$

  $$\mathcal{L} = \{ \alpha_1 b_1 + \cdots + \alpha_n b_n \mid \alpha_i \in \mathbb{Z} \}$$

- **Output:** A vector $v \in \mathcal{L}$ such that $\|v\| \leq M$

  The problem above is NP-hard, as it would allow us to find the *shortest vector* in a lattice (which is an NP-hard problem).
Short Vectors in a Lattice

- **Input:** linearly independent vectors \(b_1, \ldots, b_n \in \mathbb{Z}^n\), bound \(M \in \mathbb{N}\)

  \[\mathcal{L} = \{\alpha_1 b_1 + \cdots + \alpha_n b_n \mid \alpha_i \in \mathbb{Z}\}\]

- **Output:** A vector \(v \in \mathcal{L}\) such that \(\|v\| \leq M\)

The problem above is NP-hard, as it would allow us to find the *shortest vector* in a lattice (which is an NP-hard problem).

So we will settle for the approximation version:

1. **Input:** linearly independent vectors \(b_1, \ldots, b_n \in \mathbb{Z}^n\), approximation bound \(M \in \mathbb{N}\)

   \[\mathcal{L} = \{\alpha_1 b_1 + \cdots + \alpha_n b_m \mid \alpha_i \in \mathbb{Z}\}\]

2. **Output:** A vector \(v \in \mathcal{L}\) such that \(\|v\| \leq M \cdot \lambda(\mathcal{L})\), where \(\lambda(\mathcal{L})\) is the length of the shortest vector in \(\mathcal{L}\).
Short Vectors in a Lattice

**Input:** linearly independent vectors \( b_1, \ldots, b_n \in \mathbb{Z}^n \), bound \( M \in \mathbb{N} \)

\[ \mathcal{L} = \{ \alpha_1 b_1 + \cdots + \alpha_n b_n \mid \alpha_i \in \mathbb{Z} \} \]

**Output:** A vector \( v \in \mathcal{L} \) such that \( \|v\| \leq M \)

The problem above is NP-hard, as it would allow us to find the *shortest vector* in a lattice (which is an NP-hard problem).

So we will settle for the approximation version:

1. **Input:** linearly independent vectors \( b_1, \ldots, b_n \in \mathbb{R}^n \), approximation bound \( M \in \mathbb{N} \)

\[ \mathcal{L} = \{ \alpha_1 b_1 + \cdots + \alpha_n b_m \mid \alpha_i \in \mathbb{Z} \} \]

2. **Output:** A vector \( v \in \mathcal{L} \) such that \( \|v\| \leq M \cdot \lambda(\mathcal{L}) \), where \( \lambda(\mathcal{L}) \) is the length of the shortest vector in \( \mathcal{L} \)

Today we will see a polynomial time algorithm when \( M = 2^{\frac{n-1}{2}} \)

(*quite ok for our purposes*)
Observations on our Lattice Problem

- In previous lecture, we wrote the problem with input vectors

\[ b_1, \ldots, b_m \in \mathbb{Z}^n \]

where \( m, n \) could be distinct. Why isn’t the problem from the previous slide less general?  

\(^1\)See homework and practice exercises for this.
Observations on our Lattice Problem

- In previous lecture, we wrote the problem with input vectors
  \[ b_1, \ldots, b_m \in \mathbb{Z}^n \]
  where \( m, n \) could be distinct. Why is\textbf{n’t} the problem from the previous slide \textit{less general}?

- If \( m < n \), can simply make \( m = n \) by reducing the dimension of ambient space

---

\(^1\)See homework and practice exercises for this.
Observations on our Lattice Problem

- In previous lecture, we wrote the problem with input vectors

\[ b_1, \ldots, b_m \in \mathbb{Z}^n \]

where \( m, n \) could be distinct. Why isn’t the problem from the previous slide *less general*?

- If \( m < n \), can simply make \( m = n \) by reducing the dimension of ambient space

- If \( m > n \), we can simply take a linearly independent subset of the vectors \( b_i \) which span the lattice.

---

1See homework and practice exercises for this.
Observations on our Lattice Problem

- In previous lecture, we wrote the problem with input vectors

\[ b_1, \ldots, b_m \in \mathbb{Z}^n \]

where \( m, n \) could be distinct. Why isn’t the problem from the previous slide less general?

- If \( m < n \), can simply make \( m = n \) by reducing the dimension of ambient space orthogonal projections\(^1\)

- If \( m > n \), we can simply take a linearly independent subset of the vectors \( b_i \) which span the lattice.

- Given previous bullets, we can indeed assume that \( m = n \).

\(^1\)See homework and practice exercises for this.
Reducing to a basis of $\mathbb{R}^n$

- Suppose we have $b_1, \ldots, b_m \in \mathbb{Z}^n$ where $m > n$ and we know that $b_1, \ldots, b_m$ span $\mathbb{R}^n$. 

Reducing to a basis of $\mathbb{R}^n$

- Suppose we have $b_1, \ldots, b_m \in \mathbb{Z}^n$ where $m > n$ and we know that $b_1, \ldots, b_m$ span $\mathbb{R}^n$.
- Let $B = (b_1 \ b_2 \ \cdots \ b_m)$ be matrix with $b_k$'s as columns. Let $b_k = (b_{k1}, b_{k2}, \ldots, b_{kn})^T$. 

\[ B = \begin{pmatrix} \vdots & \vdots & \cdots & \vdots \\ b_1 & b_2 & \cdots & b_m \end{pmatrix} \]

\[ b_k = \begin{pmatrix} b_{k1} \\ b_{k2} \\ \vdots \\ b_{kn} \end{pmatrix} \]
Reducing to a basis of $\mathbb{R}^n$

- Suppose we have $b_1, \ldots, b_m \in \mathbb{Z}^n$ where $m > n$ and we know that $b_1, \ldots, b_m$ span $\mathbb{R}^n$.
- Let $B = \begin{pmatrix} b_1 & b_2 & \cdots & b_m \end{pmatrix}$ be a matrix with $b_k$'s as columns. Let $b_k = (b_{k1}, b_{k2}, \ldots, b_{kn})^T$.
- Compute $g = \gcd(b_{11}, b_{21}, \ldots, b_{m1})$ and integers $a_1, \ldots, a_m$ such that $\sum_{i=1}^m a_i b_{1i} = g$.
Reducing to a basis of $\mathbb{R}^n$

- Suppose we have $b_1, \ldots, b_m \in \mathbb{Z}^n$ where $m > n$ and we know that $b_1, \ldots, b_m$ span $\mathbb{R}^n$.
- Let $B = \begin{pmatrix} b_1 & b_2 & \cdots & b_m \end{pmatrix}$ be matrix with $b_k$’s as columns. Let $b_k = (b_{k1}, b_{k2}, \ldots, b_{kn})^T$.

1. Compute $g = \gcd(b_{11}, b_{21}, \ldots, b_{m1})$ and integers $a_1, \ldots, a_m$ such that $\sum_{i=1}^m a_i b_{1i} = g$.

2. Construct a new basis $C = (c_1, \ldots, c_m)$ as follows:

$$c_1 = a_1 b_1 + \cdots + a_m b_m$$

$$c_k = b_k - \frac{b_{k1}}{g} \cdot c_1$$

(If $c_{k1} = 0$)

Note that new basis also spans the same lattice $\mathcal{L}$ and $c_{k1} = 0$ for all $k > 1$.

$$c_1 = \sum_{k=1}^m a_k b_k$$

$$c = \begin{pmatrix} a_1 b_{11} + a_2 b_{21} + \cdots + a_m b_{m1} \end{pmatrix} = g$$

$$\mathcal{L}(b_1, \ldots, b_n) = \mathcal{L}(c_1, \ldots, c_n)$$
Reducing to a basis of $\mathbb{R}^n$

- Suppose we have $b_1, \ldots, b_m \in \mathbb{Z}^n$ where $m > n$ and we know that $b_1, \ldots, b_m$ span $\mathbb{R}^n$.
- Let $B = (b_1 \ b_2 \ \cdots \ b_m)$ be matrix with $b_k$’s as columns. Let $b_k = (b_{k1}, b_{k2}, \ldots, b_{kn})^T$.

1. Compute $g = \gcd(b_{11}, b_{21}, \ldots, b_{m1})$ and integers $a_1, \ldots, a_m$ such that $\sum_{i=1}^m a_i b_{1i} = g$.

2. Construct a new basis $C = (c_1, \ldots, c_m)$ as follows:
   
   $c_1 = a_1 b_1 + \cdots + a_m b_m$
   
   $c_k = b_k - \frac{b_{k1}}{g} \cdot c_1$

   Note that new basis also spans the same lattice $\mathcal{L}$ and $c_{k1} = 0$ for all $k > 1$.

3. Repeat step (1) for $(c_2, \ldots, c_m)$ recursion.
Reducing to a basis of $\mathbb{R}^n$

- Suppose we have $b_1, \ldots, b_m \in \mathbb{Z}^n$ where $m > n$ and we know that $b_1, \ldots, b_m$ span $\mathbb{R}^n$
- Let $B = \begin{pmatrix} b_1 & b_2 & \cdots & b_m \end{pmatrix}$ be matrix with $b_k$’s as columns. Let $b_k = (b_{k1}, b_{k2}, \ldots, b_{kn})^T$.
  1. compute $g = \gcd(b_{11}, b_{21}, \ldots, b_{m1})$ and integers $a_1, \ldots, a_m$ such that $\sum_{i=1}^m a_ib_{1i} = g$
  2. Construct a new basis $C = (c_1, \ldots, c_m)$ as follows:

\[
c_1 = a_1b_1 + \cdots + a_mb_m
\]
\[
c_k = b_k - \frac{b_{k1}}{g} \cdot c_1
\]

Note that new basis also spans the same lattice $\mathcal{L}$ and $c_{k1} = 0$ for all $k \geq 1$.

- Repeat step (1) for $(c_2, \ldots, c_m)$ recursion
- Note that by the end of this process, we will have a matrix

\[
M = \begin{pmatrix} A & 0 \end{pmatrix}
\]

where $A \in \mathbb{Z}^{n \times n}$ is integral, full rank, and the column vectors of $A$ span the same lattice $\mathcal{L}$. 
Example

\[
\begin{pmatrix}
  2 & 5 & 4 \\
  3 & -1 & 4 \\
\end{pmatrix}
\]

\[
\begin{align*}
\mathbf{c}_1 &= \begin{pmatrix} 2 \\ 3 \\ b_1 \end{pmatrix} \cdot 0 + \begin{pmatrix} 5 \\ -1 \\ b_2 \end{pmatrix} \cdot 1 + (-1) \begin{pmatrix} 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -5 \end{pmatrix} \\
\mathbf{c}_2 &= b_2 - \frac{5}{1} \cdot \mathbf{c}_1 = \begin{pmatrix} 5 \\ -1 \end{pmatrix} - 5 \begin{pmatrix} 1 \\ -5 \end{pmatrix} = \begin{pmatrix} 0 \\ 24 \end{pmatrix} \\
\mathbf{c}_3 &= b_3 - \frac{4}{1} \cdot \mathbf{c}_1 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} - 4 \begin{pmatrix} 1 \\ -5 \end{pmatrix} = \begin{pmatrix} 0 \\ 24 \end{pmatrix}
\end{align*}
\]

\[
\begin{pmatrix}
  1 & 0 & 0 \\
  -5 & 24 & 24 \\
\end{pmatrix}
\begin{pmatrix}
  1 & 0 & 0 \\
  -5 & 24 & 0 \\
\end{pmatrix}
\]

\[1 = \gcd(2, 5, 4) = 2 \cdot 0 + 5 \cdot 1 + 4 \cdot (-1)\]
Example
Example
Determinant of a Lattice

Now that we clarified the assumption that $m = n$ and that $b_1, \ldots, b_n$ form a basis of $\mathbb{R}^n$, we can define an invariant of our lattice: the determinant

$$\det(\mathcal{L}) = | \det (b_1 \ b_2 \ \cdots \ b_n) |$$

$$B = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} \quad \det(B) = | \det(B) |$$
Determinant of a Lattice

- Now that we clarified the assumption that \( m = n \) and that \( b_1, \ldots, b_n \) form a basis of \( \mathbb{R}^n \), we can define an invariant of our lattice: the determinant

\[
\det(\mathcal{L}) = | \det (b_1 \ b_2 \ \cdots \ b_n) |
\]

- The definition above is basis independent: if \( (c_1, c_2, \ldots, c_n) \) is another basis for \( \mathcal{L} \), we have that

\[
| \det (b_1 \ b_2 \ \cdots \ b_n) | = | \det (c_1 \ c_2 \ \cdots \ c_n) |
\]
Determinant of a Lattice

- Now that we clarified the assumption that \( m = n \) and that \( b_1, \ldots, b_n \) form a basis of \( \mathbb{R}^n \), we can define an **invariant** of our lattice: the **determinant**

  \[
  \det(\mathcal{L}) = |\det (b_1 \ b_2 \ \cdots \ b_n)|
  \]

- The definition above is **basis independent**: if \((c_1, c_2, \ldots, c_n)\) is another basis for \( \mathcal{L} \), we have that

  \[
  |\det (b_1 \ b_2 \ \cdots \ b_n)| = |\det (c_1 \ c_2 \ \cdots \ c_n)|
  \]

- Proof: invertible linear transformation taking one basis to another.

\[
C_k = \sum_{j=1}^{n} A_{kj} b_j \quad A_{kj} \in \mathcal{L}
\]

\[
(c_1 \ c_2 \ \cdots \ c_n) = (b_1 \ \cdots \ b_n) (A_{11} \ A_{12} \ \cdots \ A_{1n})
\]

\[
B = C \tilde{A}
\]
\[
\begin{pmatrix}
  c_1 & c_2 & \cdots & c_n \\
  \end{pmatrix}
= \begin{pmatrix}
  b_1 & \cdots & b_n \\
  \end{pmatrix}
\begin{pmatrix}
  A_{11} & A_{12} & \cdots & A_{1n} \\
  A_{21} & A_{22} & \cdots & A_{2n} \\
  \vdots & & & \vdots \\
  A_{n1} & A_{n2} & \cdots & A_{nn} \\
  \end{pmatrix}
\]

\[c_1 = A_{11}b_1 + A_{21}b_2 + \cdots + A_{n1}b_n\]

\[A_{ij} \in \mathbb{Z}\]

\[
\text{det}(c) = \frac{\text{det}(B) \cdot \text{det}(A)}{\text{det}(BA)} \in \mathbb{Z}
\]

\[\Rightarrow \text{det}(B) | \text{det}(c) \quad \text{and} \quad \text{det}(B) = \pm \text{det}(c)\]
Determinant of a Lattice

- Now that we clarified the assumption that \( m = n \) and that \( b_1, \ldots, b_n \) form a basis of \( \mathbb{R}^n \), we can define an invariant of our lattice: the determinant
  \[
  \det(\mathcal{L}) = \left| \det \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} \right|
  \]
- The definition above is basis independent: if \( (c_1, c_2, \ldots, c_n) \) is another basis for \( \mathcal{L} \), we have that
  \[
  \left| \det \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} \right| = \left| \det \begin{pmatrix} c_1 & c_2 & \cdots & c_n \end{pmatrix} \right|
  \]
- Proof: invertible linear transformation taking one basis to another.
- To go from one basis to another, we can do elementary column operations, that is, if we have basis \( b_1, \ldots, b_n \) then we can do
  \[
  c_k = b_k - \alpha b_i, \quad \alpha \in \mathbb{Z} \quad \text{and} \quad c_\ell = b_\ell \quad \text{for} \quad \ell \neq k
  \]
Short Vectors in a Lattice

**Algorithm Idea: Find Good Basis**

Gram-Schmidt Orthogonalization

Lenstra-Lenstra-Lovasz (LLL) Basis Reduction Algorithm

Conclusion

Acknowledgements
Algorithm idea: a good basis will contain a short vector!

Let’s work this out for $n = 2$. Suppose we have $a, b \in \mathbb{Z}^2$ which form a basis for the lattice $\mathcal{L} = \mathbb{Z}a + \mathbb{Z}b$. Also, assume $\|a\| \leq \|b\|$. 
Algorithm idea: a good basis will contain a short vector!

- Let’s work this out for $n = 2$. Suppose we have $a, b \in \mathbb{Z}^2$ which form a basis for the lattice $L = \mathbb{Z}a + \mathbb{Z}b$. Also, assume $\|a\| \leq \|b\|$.
- If we have that $\|a\| \leq \|b\| \leq \|b + \alpha a\|$ for all $\alpha \in \mathbb{Z}$, then we have that $a$ is the shortest vector in our lattice!
Algorithm idea: a good basis will contain a short vector!

- Let’s work this out for $n = 2$. Suppose we have $a, b \in \mathbb{Z}^2$ which form a basis for the lattice $\mathcal{L} = \mathbb{Z}a + \mathbb{Z}b$. Also, assume $\|a\| \leq \|b\|$.

- If we have that $\|a\| \leq \|b\| \leq \|b + \alpha a\|$ for all $\alpha \in \mathbb{Z}$, then we have that $a$ is the \textit{shortest vector} in our lattice!

- Proof: let $z = \beta a + \gamma b$, where $\beta, \gamma \in \mathbb{Z}$. Can assume $\beta, \gamma \neq 0$

\[
\begin{align*}
\gamma = 0 & \implies z = \beta a \implies \|z\| = \|\beta a\| \\
\beta = 0 & \implies z = \gamma b \implies \|z\| \geq \|\gamma b\|
\end{align*}
\]
Algorithm idea: a good basis will contain a short vector!

- Let’s work this out for $n = 2$. Suppose we have $a, b \in \mathbb{Z}^2$ which form a basis for the lattice $\mathcal{L} = \mathbb{Z}a + \mathbb{Z}b$. Also, assume $\|a\| \leq \|b\|$.
- If we have that $\|a\| \leq \|b\| \leq \|b + \alpha a\|$ for all $\alpha \in \mathbb{Z}$, then we have that $a$ is the shortest vector in our lattice!
- Proof: let $z = \beta a + \gamma b$, where $\beta, \gamma \in \mathbb{Z}$. Can assume $\beta, \gamma \neq 0$.
- Case 1: $\beta > \gamma$  
  $$(a+b)^2 = \|a\|^2 + \|b\|^2 + 2\langle a, b \rangle \geq \|b\|^2 = \langle b, b \rangle$$
  $$\Rightarrow 2 \langle a, b \rangle \geq -\langle a, a \rangle$$

$$\|z\|^2 = \langle \beta a + \gamma b, \beta a + \gamma b \rangle = \beta^2 \|a\|^2 + \gamma^2 \|b\|^2 + 2\beta\gamma \langle a, b \rangle \geq \beta^2 \|a\|^2 + \gamma^2 \|b\|^2 - \beta \gamma \|a\|^2 = \beta (\beta - \gamma) \|a\|^2 + \gamma^2 \|b\|^2 \geq \frac{\beta (\beta - \gamma)}{\gamma} \|a\|^2 \geq \|a\|^2.$$
Algorithm idea: a good basis will contain a short vector!

- Let’s work this out for $n = 2$. Suppose we have $a, b \in \mathbb{Z}^2$ which form a basis for the lattice $\mathcal{L} = \mathbb{Z}a + \mathbb{Z}b$. Also, assume $\|a\| \leq \|b\|$.
- If we have that $\|a\| \leq \|b\| \leq \|b + \alpha a\|$ for all $\alpha \in \mathbb{Z}$, then we have that $a$ is the shorted vector in our lattice!
- Proof: let $z = \beta a + \gamma b$, where $\beta, \gamma \in \mathbb{Z}$. Can assume $\beta, \gamma \neq 0$
- Case 1: $\beta > \gamma$
- Case 2: $\beta \leq \gamma$

$$\|a + b\|^2 \geq \|a\|^2$$ similar to previous slide
Algorithm idea: a good basis will contain a short vector!

- Let's work this out for \( n = 2 \). Suppose we have \( a, b \in \mathbb{Z}^2 \) which form a basis for the lattice \( \mathcal{L} = \mathbb{Z}a + \mathbb{Z}b \). Also, assume \( \|a\| \leq \|b\| \).

- If we have that \( \|a\| \leq \|b\| \leq \|b + \alpha a\| \) for all \( \alpha \in \mathbb{Z} \), then we have that \( a \) is the shortest vector in our lattice!

- Proof: let \( z = \beta a + \gamma b \), where \( \beta, \gamma \in \mathbb{Z} \). Can assume \( \beta, \gamma \neq 0 \)

  - Case 1: \( \beta > \gamma \)
  - Case 2: \( \beta \leq \gamma \)

  - How do we find such a basis \((a, b)\) with the property from the second bullet? An orthogonal basis does it.
Algorithm idea: a good basis will contain a short vector!

- Let’s work this out for $n = 2$. Suppose we have $a, b \in \mathbb{Z}^2$ which form a basis for the lattice $L = \mathbb{Z}a + \mathbb{Z}b$. Also, assume $\|a\| \leq \|b\|$.
- If we have that $\|a\| \leq \|b\| \leq \|b + \alpha a\|$ for all $\alpha \in \mathbb{Z}$, then we have that $a$ is the **shortest vector** in our lattice!
- Proof: let $z = \beta a + \gamma b$, where $\beta, \gamma \in \mathbb{Z}$. Can assume $\beta, \gamma \neq 0$
- Case 1: $\beta > \gamma$
- Case 2: $\beta \leq \gamma$
- How do we find such a basis $(a, b)$ with the property from the second bullet? An orthogonal basis does it.
- It will not always be the case that a lattice has orthogonal basis. For instance

\[
\begin{pmatrix}
0 & 1 \\
3 & -1
\end{pmatrix}
\]
Algorithm idea: a good basis will contain a short vector!

- Let's work this out for \( n = 2 \). Suppose we have \( a, b \in \mathbb{Z}^2 \) which form a basis for the lattice \( \mathcal{L} = \mathbb{Z}a + \mathbb{Z}b \). Also, assume \( \|a\| \leq \|b\| \).

- If we have that \( \|a\| \leq \|b\| \leq \|b + \alpha a\| \) for all \( \alpha \in \mathbb{Z} \), then we have that \( a \) is the \textit{shortest vector} in our lattice!

- Proof: let \( z = \beta a + \gamma b \), where \( \beta, \gamma \in \mathbb{Z} \). Can assume \( \beta, \gamma \neq 0 \)

  - Case 1: \( \beta > \gamma \)
  
  - Case 2: \( \beta \leq \gamma \)

- How do we find such a basis \((a, b)\) with the property from the second bullet? An orthogonal basis does it.

- It will not always be the case that a lattice has orthogonal basis. For instance

\[
\begin{pmatrix}
0 & 1 \\
3 & -1
\end{pmatrix}
\]

- “Close enough” to orthogonal does it!
Looking at Counterexample

\[ B = \begin{pmatrix} 0 & 1 \\ 3 & -1 \end{pmatrix} \quad U = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \text{ orthogonal} \]

\[ |\det(B)| = |\det(U)| \text{ invariant} \]

\[ |0 \cdot (-1) - 1 \cdot 3| = 3 = |\det(U)| \]

\[ = \|u_1\| \cdot \|u_2\| \]

\[ U \cdot U^T = \det(U)^2 \cdot I \quad (U \text{ orthogonal}) \]
\[ \|u_1\| \cdot \|u_2\| = 3 \quad U \quad B = \begin{pmatrix} 0 & 1 \\ 3 & -1 \end{pmatrix} \]

\[ u_1, u_2 \in L^2 \left( \begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right) \]

\[ \Rightarrow u_1, u_2 \in L^2 \]

\[ \|u_1\|^2 \cdot \|u_2\|^2 = 9 \quad \Rightarrow \quad \|u_1\| = 3 \]

\[ \|u_2\| = \frac{1}{\sqrt{e_1, e_2}} \]

\[ \|u_1\| = \|u_2\| = \sqrt{3} \]
Gauss’ Reduction Algorithm

- The LLL algorithm is a generalization of 2D basis reduction due to Gauss.
- Idea: given two vectors \( u, v \), s.t. \( \|u\| \leq \|v\| \), subtract off as much of \( u \)'s projection from \( v \), while staying in the lattice.

Making \( u, v \) “as orthogonal as possible”

\[ \|v - \alpha u\| \text{ smallest } \alpha \in \mathbb{Z} \]
Gauss’ Reduction Algorithm

- The LLL algorithm is a generalization of 2D basis reduction due to Gauss.
- Idea: given two vectors $u, v$, s.t. $\|u\| \leq \|v\|$ subtract off as much of $u$’s projection from $v$, while staying in the lattice.
- There is $\alpha \in \mathbb{Z}$ such that

$$|\langle v - \alpha u, u \rangle| \leq \frac{1}{2} \|u\|^2$$
Gauss’ Reduction Algorithm

- The LLL algorithm is a generalization of 2D basis reduction due to Gauss.
- Idea: given two vectors $u, v$, s.t. $\|u\| \leq \|v\|$ subtract off as much of $u$’s projection from $v$, while staying in the lattice.
- There is $\alpha \in \mathbb{Z}$ such that

\[
\|v - \alpha u, u\| \leq \frac{1}{2} \|u\|^2
\]

- Proof: if $\beta = \frac{\langle u, v \rangle}{\|u\|}$, take $\alpha \in \mathbb{Z}$ closest to $\beta$. Thus $|\alpha - \beta| \leq 1/2$

\[
\langle v - \alpha u, u \rangle = \langle v - \beta u, u \rangle + \langle (\beta - \alpha)u, u \rangle \leq \frac{1}{2} \|u\|^2
\]

$\quad v - \beta u + (\beta - \alpha)u$

$\quad v - \beta u \perp u$
Gauss’ Reduction Algorithm

- The LLL algorithm is generalization of 2D basis reduction due to Gauss
- Idea: given two vectors $u, v$, s.t. $\|u\| \leq \|v\|$ subtract off as much of $u$'s projection from $v$, while staying in the lattice
- There is $\alpha \in \mathbb{Z}$ such that

$$|\langle v - \alpha u, u \rangle| \leq \frac{1}{2} \|u\|^2$$

- Proof: if $\beta = \frac{\langle u, v \rangle}{\|u\|}$, take $\alpha \in \mathbb{Z}$ closest to $\beta$. Thus $|\alpha - \beta| \leq 1/2$

$$|\langle v - \alpha u, u \rangle| = |\langle v - \beta u, u \rangle + \langle (\beta - \alpha)u, u \rangle| \leq \frac{1}{2} \|u\|^2$$

- If $\|v - \alpha u\| \geq \|u\|$ stop. Otherwise swap the vectors and continue.
Gauss’ Reduction Algorithm

- The LLL algorithm is generalization of 2D basis reduction due to Gauss.
- Idea: given two vectors $u, v$, s.t. $\|u\| \leq \|v\|$ subtract off as much of $u$’s projection from $v$, while staying in the lattice.
- There is $\alpha \in \mathbb{Z}$ such that
  \[ |\langle v - \alpha u, u \rangle| \leq \frac{1}{2} \|u\|^2 \]

- Proof: if $\beta = \frac{\langle u, v \rangle}{\|u\|}$, take $\alpha \in \mathbb{Z}$ closest to $\beta$. Thus $|\alpha - \beta| \leq 1/2$
  \[ |\langle v - \alpha u, u \rangle| = |\langle v - \beta u, u \rangle + \langle (\beta - \alpha)u, u \rangle| \leq \frac{1}{2} \|u\|^2 \]

- If $\|v - \alpha u\| \geq \|u\|$ stop. Otherwise swap the vectors and continue.
- Note that at each iteration we are decreasing the norm of the smallest basis vector. When we cannot decrease further, previous slide gives us that $u$ is the shortest vector!
Short Vectors in a Lattice

Algorithm Idea: Find Good Basis

Gram-Schmidt Orthogonalization

Lenstra-Lenstra-Lovasz (LLL) Basis Reduction Algorithm

Conclusion

Acknowledgements
Orthogonal Bases and Short Vectors

- Note that if \( b_1, \ldots, b_n \in \mathbb{Z}^n \) were an orthogonal basis for the lattice, then one of these vectors must be the shortest!
Orthogonal Bases and Short Vectors

- Note that if \( b_1, \ldots, b_n \in \mathbb{Z}^n \) were an orthogonal basis for the lattice, then one of these vectors must be the shortest!

- It will not always be the case that a lattice has orthogonal basis. For instance

\[
\begin{pmatrix}
    0 & 1 \\
    3 & -1
\end{pmatrix}
\]
Orthogonal Bases and Short Vectors

- Note that if $b_1, \ldots, b_n \in \mathbb{Z}^n$ were an orthogonal basis for the lattice, then one of these vectors must be the shortest!
- It will not always be the case that a lattice has orthogonal basis. For instance

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

- But we could still attempt to get something “almost as good”
- Let us compare to “the best” we could hope for: Gram-Schmidt
Orthogonal Bases and Short Vectors

- Note that if $b_1, \ldots, b_n \in \mathbb{Z}^n$ were an orthogonal basis for the lattice, then one of these vectors must be the shortest!
- It will not always be the case that a lattice has orthogonal basis. For instance

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

- But we could still attempt to get something “almost as good”
- Let us compare to “the best” we could hope for: Gram-Schmidt

**Input:** basis $b_1, \ldots, b_n \in \mathbb{R}^n$

**Output:** Set of orthogonal basis $u_1, \ldots, u_n$
Orthogonal Bases and Short Vectors

- Note that if \( b_1, \ldots, b_n \in \mathbb{Z}^n \) were an orthogonal basis for the lattice, then one of these vectors must be the shortest!
- It will not always be the case that a lattice has orthogonal basis. For instance
  \[
  \begin{pmatrix}
  0 & 1 \\
  3 & -1
  \end{pmatrix}
  \]
- But we could still attempt to get something “almost as good”
- Let us compare to “the best” we could hope for: **Gram-Schmidt**

**Input:** basis \( b_1, \ldots, b_n \in \mathbb{R}^n \)

**Output:** Set of orthogonal basis \( u_1, \ldots, u_n \)

1. Set \( u_1 = b_1 \)
Orthogonal Bases and Short Vectors

Note that if \( b_1, \ldots, b_n \in \mathbb{Z}^n \) were an orthogonal basis for the lattice, then one of these vectors must be the **shortest**!

It will not always be the case that a lattice has orthogonal basis. For instance

\[
\begin{pmatrix}
0 & 1 \\
3 & -1 \\
\end{pmatrix}
\]

But we could still attempt to get something “almost as good”

Let us compare to “the best” we could hope for: **Gram-Schmidt**

**Input:** basis \( b_1, \ldots, b_n \in \mathbb{R}^n \)

**Output:** Set of orthogonal basis \( u_1, \ldots, u_n \)

1. Set \( u_1 = b_1 \)
2. Repeat the following for \( 2 \leq k \leq n \)

\[
u_k = b_k - \sum_{i=1}^{k-1} \frac{\langle b_k, u_i \rangle}{\|u_i\|^2} \cdot u_i
\]

\[\langle u_k, u_i \rangle = 0 \quad \text{for } i < k\]
Orthogonal Bases and Short Vectors

- Note that if \( b_1, \ldots, b_n \in \mathbb{Z}^n \) were an orthogonal basis for the lattice, then one of these vectors must be the shortest!
- It will not always be the case that a lattice has orthogonal basis. For instance

\[
\begin{pmatrix}
0 & 1 \\
3 & -1
\end{pmatrix}
\]

- But we could still attempt to get something “almost as good”
- Let us compare to “the best” we could hope for: Gram-Schmidt

**Input:** basis \( b_1, \ldots, b_n \in \mathbb{R}^n \)

**Output:** Set of orthogonal basis \( u_1, \ldots, u_n \)

1. Set \( u_1 = b_1 \)
2. Repeat the following for \( 2 \leq k \leq n \)

\[
u_k = b_k - \sum_{i=1}^{k-1} \frac{\langle b_k, u_i \rangle}{\| u_i \|^2} \cdot u_i
\]

- Orthogonal basis *not necessarily* a basis for our lattice!
Properties of Gram-Schmidt Basis

Gram-Schmidt algorithm:

1. Set $u_1 = b_1$
2. Repeat the following for $2 \leq k \leq n$

$$u_k = b_k - \sum_{i=1}^{k-1} \frac{\langle b_k, u_i \rangle}{\|u_i\|^2} \cdot u_i$$
Properties of Gram-Schmidt Basis

- Gram-Schmidt algorithm:
  1. Set $u_1 = b_1$
  2. Repeat the following for $2 \leq k \leq n$

\[
u_k = b_k - \sum_{i=1}^{k-1} \frac{\langle b_k, u_i \rangle}{\|u_i\|^2} \cdot u_i\]

- Can write

\[
b_k = \sum_{i=1}^{k} \mu_{ki} \cdot u_i\]

with $\mu_{kk} = 1$. 
Properties of Gram-Schmidt Basis

- **Gram-Schmidt algorithm:**
  1. Set \( u_1 = b_1 \)
  2. Repeat the following for \( 2 \leq k \leq n \)

\[
u_k = b_k - \sum_{i=1}^{k-1} \frac{\langle b_k, u_i \rangle}{\| u_i \|^2} \cdot u_i
\]

- Can write

\[
b_k = \sum_{i=1}^{k} \mu_{ki} \cdot u_i
\]

  with \( \mu_{kk} = 1 \).

- If don’t change the order but make some \( b_k = b_k + \alpha b_j \) with \( j < k \) the GSO basis stays the same
Properties of Gram-Schmidt Basis

- Gram-Schmidt algorithm:
  1. Set $u_1 = b_1$
  2. Repeat the following for $2 \leq k \leq n$

$$u_k = b_k - \sum_{i=1}^{k-1} \frac{\langle b_k, u_i \rangle}{\|u_i\|^2} \cdot u_i$$

- Can write

$$b_k = \sum_{i=1}^{k} \mu_{ki} \cdot u_i$$

with $\mu_{kk} = 1$.

- If don’t change the order but make some $b_k = b_k + \alpha b_j$ with $j < k$ the GSO basis stays the same.

- If input basis is *integral* (or rational) then the output basis is *rational*.
Shortest vector & Gram-Schmidt Orthogonalization (GSO)

From now on, given any basis \((b_1, \ldots, b_n)\) we can refer to its GSO \((u_1, \ldots, u_n)\)
Shortest vector & Gram-Schmidt Orthogonalization (GSO)

- From now on, given any basis \((b_1, \ldots, b_n)\) we can refer to its GSO \((u_1, \ldots, u_n)\)

- Relationship between GSO basis and shortest vector in \(\mathcal{L}(b_1, \ldots, b_n)\)
  Shortest vector in GSO basis \textit{lower bounds shortest vector} in \(\mathcal{L}\).
Shortest vector & Gram-Schmidt Orthogonalization (GSO)

- From now on, given any basis \( (b_1, \ldots, b_n) \) we can refer to its GSO \( (u_1, \ldots, u_n) \).
- Relationship between GSO basis and shortest vector in \( \mathcal{L}(b_1, \ldots, b_n) \):
  - Shortest vector in GSO basis *lower bounds* shortest vector in \( \mathcal{L} \).
- Proof: let \( v \in \mathcal{L} \). Then we can write \( v = \alpha_1 b_1 + \cdots + \alpha_n b_n, \ \alpha_j \in \mathbb{Z} \).
Shortest vector & Gram-Schmidt Orthogonalization (GSO)

- From now on, given any basis \((b_1, \ldots, b_n)\) we can refer to its GSO \((u_1, \ldots, u_n)\)
- Relationship between GSO basis and shortest vector in \(\mathcal{L}(b_1, \ldots, b_n)\)
  
  Shortest vector in GSO basis lower bounds shortest vector in \(\mathcal{L}\).

Proof: let \(v \in \mathcal{L}\). Then we can write \(v = \alpha_1 b_1 + \cdots + \alpha_n b_n, \alpha_j \in \mathbb{Z}\)

- By GSO, we can write \(b_k = \sum_{i=1}^{k} \mu_{ki} \cdot u_i\), with \(\mu_{kk} = 1\)
Shortest vector & Gram-Schmidt Orthogonalization (GSO)

- From now on, given any basis \( (b_1, \ldots, b_n) \) we can refer to its GSO \((u_1, \ldots, u_n)\)
- Relationship between GSO basis and shortest vector in \( L(b_1, \ldots, b_n) \)
  - Shortest vector in GSO basis lower bounds shortest vector in \( L \).
- Proof: let \( v \in L \). Then we can write \( v = \alpha_1 b_1 + \cdots + \alpha_n b_n, \alpha_j \in \mathbb{Z} \)
- By GSO, we can write \( b_k = \sum_{i=1}^{k} \mu_{ki} \cdot u_i \), with \( \mu_{kk} = 1 \)
- Thus, if \( \alpha_t \neq 0 \) and \( \alpha_\ell = 0 \) for all \( \ell > t \):
  \[
  v = \beta_1 u_1 + \cdots + \beta_t u_t
  \]
  With \( \beta_t = 1 \), as no other \( u_i \) depends on \( u_t \).

\[
\mathcal{U} = \alpha_1 b_1 + \cdots + \alpha_t b_t
\]

\[
\mathcal{U} = \sum_{i=1}^{t} \beta_i u_i
\]

\( b_t = u_t + (-c) \)
Shortest vector & Gram-Schmidt Orthogonalization (GSO)

- From now on, given any basis \((b_1, \ldots, b_n)\) we can refer to its GSO \((u_1, \ldots, u_n)\)
- Relationship between GSO basis and shortest vector in \(L(b_1, \ldots, b_n)\)
  Shortest vector in GSO basis **lower bounds shortest vector** in \(L\).
- Proof: let \(v \in L\). Then we can write \(v = \alpha_1 b_1 + \cdots + \alpha_n b_n, \alpha_j \in \mathbb{Z}\)
- By GSO, we can write \(b_k = \sum_{i=1}^{k} \mu_{ki} \cdot u_i, \text{ with } \mu_{kk} = 1\)
- Thus, if \(\alpha_t \neq 0\) and \(\alpha_\ell = 0\) for all \(\ell > t\):
  \[
  v = \beta_1 u_1 + \cdots + \beta_t u_t
  \]
  With \(\beta_t = \alpha_t\), as no other \(u_i\) depends on \(u_t\).
- And the norm is given by:
  \[
  \|v\| = |\beta_1| \cdot \|u_1\| + \cdots + |\beta_t| \cdot \|u_t\| \geq \|u_t\|
  \]
Now we are ready to define what a “good basis” is:
Let \((u_1, \ldots, u_n)\) be the GSO basis from \((b_1, \ldots, b_n)\)

\[ b_k = \sum_{i=1}^{k} \mu_{ki} u_i \]
Reduced Basis

- Now we are ready to define what a “good basis” is:
- Let \((u_1, \ldots, u_n)\) be the GSO basis from \((b_1, \ldots, b_n)\)

\[
b_k = \sum_{i=1}^{k} \mu_{ki} u_i
\]

- A basis \((b_1, \ldots, b_n)\) is a reduced basis if
  1. Each \(|\mu_{ki}| \leq 1/2\) when \(i \neq k\)  
  2. For each \(k\),

\[
\|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2
\]

\[
b_k = u_k + \sum_{i < k} \mu_{ki} u_i
\]

- Orthogonality of \(b_i\)’s
- G so basis does not have “spikes”
Reduced Basis

- Now we are ready to define what a “good basis” is:
- Let \((u_1, \ldots, u_n)\) be the GSO basis from \((b_1, \ldots, b_n)\)

\[
b_k = \sum_{i=1}^{k} \mu_{ki} u_i
\]

- A basis \((b_1, \ldots, b_n)\) is a reduced basis if
  1. each \(\mu_{ki} \leq 1/2\) when \(i \neq k\)
  2. For each \(k\),

\[
\|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2
\]

- The LLL basis reduction algorithm will simply construct a reduced basis iteratively, much like Gauss’ reduction algorithm.
Short Vectors in a Lattice

Algorithm Idea: Find Good Basis

Gram-Schmidt Orthogonalization

Lenstra-Lenstra-Lovasz (LLL) Basis Reduction Algorithm

Conclusion

Acknowledgements
A basis $(b_1, \ldots, b_n)$ is a **reduced basis** if

1. each $|\mu_{ki}| \leq 1/2$ when $i \neq k$
2. For each $k$,

$$
\|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2
$$

**GSO** $(u_1, \ldots, u_n)$

$$
b_k = u_k + \sum_{i < k} \lambda_{ki} u_i
$$
LLL Basis Reduction Algorithm

- A basis \((b_1, \ldots, b_n)\) is a reduced basis if
  1. each \(\mu_{ki} \leq 1/2\) when \(i \neq k\)
  2. For each \(k\),
     \[
     \|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2
     \]

- Start with input basis \((b_1, \ldots, b_n)\) sorted by increasing norm, then get GSO \((u_1, \ldots, u_n)\)
LLL Basis Reduction Algorithm

- A basis \((b_1, \ldots, b_n)\) is a **reduced basis** if
  1. each \(\mu_{ki} \leq 1/2\) when \(i \neq k\)
  2. For each \(k\),
     \[
     \|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2
     \]
- Start with input basis \((b_1, \ldots, b_n)\) sorted by increasing norm, then get GSO \((u_1, \ldots, u_n)\)
- If condition 1 fails, then apply Gauss’ reduction to the vectors.

\[
\begin{align*}
|\mu_{ni}| &> \frac{1}{2} \\
(i < k) \\
(b_i, b_n) & \quad \text{Gauss’ reduction} \\
\end{align*}
\]

\[
\begin{align*}
 b_k &\leftarrow b_k - \alpha b_i \\
|\mu_{ni}| &\leq \frac{1}{2}
\end{align*}
\]
LLL Basis Reduction Algorithm

- A basis \((b_1, \ldots, b_n)\) is a **reduced basis** if
  1. each \(\mu_{ki} \leq 1/2\) when \(i \neq k\)
  2. For each \(k\),
     \[\|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2\]

- Start with input basis \((b_1, \ldots, b_n)\) sorted by increasing norm, then get GSO \((u_1, \ldots, u_n)\)
- If condition 1 fails, then apply Gauss’ reduction to the vectors.
- If condition 2 fails for \(k\), then swap vectors \((b_k, b_{k+1})\) and recompute the GSO.
LLL Basis Reduction Algorithm

- A basis \((b_1, \ldots, b_n)\) is a reduced basis if
  1. each \(\mu_{ki} \leq 1/2\) when \(i \neq k\)
  2. For each \(k\),
     \[
     \|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2
     \]

- Start with input basis \((b_1, \ldots, b_n)\) sorted by increasing norm, then get GSO \((u_1, \ldots, u_n)\)
- If condition 1 fails, then apply Gauss’ reduction to the vectors.
- If condition 2 fails for \(k\), then swap vectors \((b_k, b_{k+1})\) and recompute the GSO.
- Check once again both conditions. Stop only when both are satisfied.
LLL Basis Reduction Algorithm

- A basis \((b_1, \ldots, b_n)\) is a **reduced basis** if
  1. each \(\mu_{ki} \leq 1/2\) when \(i \neq k\)
  2. For each \(k\),
     \[ \|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k}u_k\|^2 \]

- Start with input basis \((b_1, \ldots, b_n)\) sorted by increasing norm, then get GSO \((u_1, \ldots, u_n)\)
- If condition 1 fails, then apply Gauss’ reduction to the vectors.
- If condition 2 fails for \(k\), then swap vectors \((b_k, b_{k+1})\) and recompute the GSO.
- Check once again both conditions. Stop only when both are satisfied.
- We will now take a deeper look into the first routine
Step 1 – Gauss Reduction

Given basis \((b_1, \ldots, b_n)\) with GSO basis \((u_1, \ldots, u_n)\), we can get a new basis \((c_1, \ldots, c_n)\) where

\[ c_k = \sum_{i=1}^{k} \gamma_{ki} u_i \quad \text{with} \quad |\gamma_{ki}| \leq 1/2 \quad \text{for} \quad i < k \]

\((c_1, \ldots, c_n)\) has some GSO basis as \((b_1, \ldots, b_n)\)
Step 1 – Gauss Reduction

- Given basis \((b_1, \ldots, b_n)\) with GSO basis \((u_1, \ldots, u_n)\), we can get a new basis \((c_1, \ldots, c_n)\) where

\[
c_k = \sum_{i=1}^{k} \gamma_{ki} u_i \quad \text{with} \quad |\gamma_{ki}| \leq 1/2
\]

- If \((b_1, \ldots, b_n)\) does not have desired property, take maximum pair \((k, i)\) such that \(|\mu_{ki}| > 1/2\).

\[
b'_k := b_k - \alpha b_i \quad \text{from Gauss reduction}
\]

\[
\Rightarrow \quad |\mu_{ki}| \leq 1/2
\]
Step 1 – Gauss Reduction

- Given basis \((b_1, \ldots, b_n)\) with GSO basis \((u_1, \ldots, u_n)\), we can get a new basis \((c_1, \ldots, c_n)\) where

\[
c_k = \sum_{i=1}^{k} \gamma_{ki} u_i \quad \text{with} \quad |\gamma_{ki}| \leq 1/2
\]

- If \((b_1, \ldots, b_n)\) does not have desired property, take maximum pair \((k, i)\) such that \(|\mu_{ki}| > 1/2\).

\[
b'_k := b_k - \alpha b_i \quad \text{from Gauss reduction}
\]

- Why maximum? Because we don’t mess up the higher \(\mu\)'s (but we may mess up the lower ones)

\[
(\kappa_{ii'}) > (\kappa_{ii}) \quad \text{not affected}
\]
Step 1 – Gauss Reduction

• Given basis \((b_1, \ldots, b_n)\) with GSO basis \((u_1, \ldots, u_n)\), we can get a new basis \((c_1, \ldots, c_n)\) where

\[
c_k = \sum_{i=1}^{k} \gamma_{ki} u_i \quad \text{with} \quad |\gamma_{ki}| \leq 1/2
\]

• If \((b_1, \ldots, b_n)\) does not have desired property, take maximum pair \((k, i)\) such that \(|\mu_{ki}| > 1/2\).

\[
b'_k := b_k - \alpha b_i \quad \text{from Gauss reduction}
\]

• Why maximum? Because we don’t mess up the higher \(\mu\)'s (but we may mess up the lower ones)

• Gauss reduction will make \(|\mu_{ki}| \leq 1/2\) but it may change \(\mu_{kj}\) for \(j < i\)
Step 1 – Gauss Reduction

- Given basis \((b_1, \ldots, b_n)\) with GSO basis \((u_1, \ldots, u_n)\), we can get a new basis \((c_1, \ldots, c_n)\) where

\[c_k = \sum_{i=1}^{k} \gamma_{ki} u_i \quad \text{with} \quad |\gamma_{ki}| \leq 1/2\]

- If \((b_1, \ldots, b_n)\) does not have desired property, take maximum pair \((k, i)\) such that \(|\mu_{ki}| > 1/2\).

\[b'_k := b_k - \alpha b_i\quad \text{from Gauss reduction}\]

- Why maximum? Because we don’t mess up the higher \(\mu\)’s (but we may mess up the lower ones)

- Gauss reduction will make \(|\mu_{ki}| \leq 1/2\) but it may change \(\mu_{kj}\) for \(j < i\)

- After we go through all pairs \((k, i)\) in decreasing order, the new coefficients \(\gamma_{ki}\) will satisfy 1 do this \(O(n^2)\) times
Runtime Analysis

- We need to prove that our algorithm will terminate, and will do so quickly
Runtime Analysis

- We need to prove that our algorithm will terminate, and will do so quickly.
- Let

\[ D(b_1, \ldots, b_n) := \prod_{i=1}^{n} \| u_i \|^ {n-i} \]

\[(u_1, \ldots, u_n)\]
Runtime Analysis

- We need to prove that our algorithm will terminate, and will do so quickly.

Let

\[ D(b_1, \ldots, b_n) := \prod_{i=1}^{n} \| u_i \|^{n-i} \]

- We will show that Gauss reduction does not change the invariant above, and step 2 only decreases it.
  - Step 1 does not change the GSO basis, so \( D \) is unchanged.

\[ D(b_1, \ldots, b_n) = D(c_1, \ldots, c_n) \]  

Gauss reductions didn't change the GSO basis.
Runtime Analysis

- We need to prove that our algorithm will terminate, and will do so quickly.
- Let
  \[ D(b_1, \ldots, b_n) := \prod_{i=1}^{n} \| u_i \|^{n-i} \]

We will show that Gauss reduction does not change the invariant above, and step 2 only decreases it.
- Step 1 does not change the GSO basis, so \( D \) is unchanged.
- Step 2 decreases \( D \) by at least \( \frac{2}{\sqrt{3}} \) exercise/practice problem
Runtime Analysis

- We need to prove that our algorithm will terminate, and will do so quickly.
- Let
  \[ D(b_1, \ldots, b_n) := \prod_{i=1}^{n} \|u_i\|^{n-i} \]
- We will show that Gauss reduction does not change the invariant above, and step 2 only decreases it.
  - Step 1 does not change the GSO basis, so \( D \) is unchanged.
  - Step 2 decreases \( D \) by at least \( \frac{2}{\sqrt{3}} \).
- Upper bound on \( D(b_1, \ldots, b_n) \):
  \[ D(b_1, \ldots, b_n) \leq (\max_i \|u_i\|)^n < \frac{(\|u\|)}{\exp(n^2, b)} \]
Runtime Analysis

- We need to prove that our algorithm will terminate, and will do so quickly

- Let

\[ D(b_1, \ldots, b_n) := \prod_{i=1}^{n} \| u_i \|^{n-i+1} \]

- We will show that Gauss reduction does not change the invariant above, and step 2 only decreases it.
  - Step 1 does not change the GSO basis, so \( D \) is unchanged
  - Step 2 decreases \( D \) by at least \( \frac{2}{\sqrt{3}} \)

- Exercise/practice problem

- Upper bound on \( D(b_1, \ldots, b_n) \):

\[ D(b_1, \ldots, b_n) \leq (\max_i \| u_i \|)^{n^2} \leq \exp(n) \]

- Lower bound: let \( B = (b_1 b_2 \cdots b_n) \)

\[ 1 \leq \det(B^T B) = \prod_{i=1}^{n} \| u_i \|^2 \leq \prod_{i=1}^{n} \| u_i \|^2 \]
\[ b_k = u_k + \sum_{i < n} \gamma_{ki} u_i \]

\[ U = \begin{pmatrix} u_1, u_2, \ldots, u_n \end{pmatrix} \]

\[ B = \begin{pmatrix} u_1, \ldots, u_n \end{pmatrix} \begin{pmatrix} 1 & \gamma_{21} & \gamma_{31} \\ 0 & 1 & \gamma_{32} \\ \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \]

\[ B = U \cdot A \]

\[ B^TB = A^T U^T U A \Rightarrow \det(B^TB) = \frac{\det(A^T A)}{\det(U^T U)} \]
\[
\det(B^T B) = \det(A^T A) \cdot \det(U^T U)
\]

\[
\text{det}(B)^2 > 0 \quad \text{integer}
\]

\[
\Rightarrow \geq 1
\]
Finding Short Vector

If \((b_1, \ldots, b_n)\) is a reduced basis of \(\mathcal{L}\), then

\[
\|b_1\| \leq 2^{\frac{n-1}{2}} \lambda(\mathcal{L})
\]

where \(\lambda(\mathcal{L})\) is the length of the shortest vector in \(\mathcal{L}\)
Finding Short Vector

- If \((b_1, \ldots, b_n)\) is a reduced basis of \(\mathcal{L}\), then

\[\|b_1\| \leq 2^{\frac{n-1}{2}} \lambda(\mathcal{L})\]

where \(\lambda(\mathcal{L})\) is the length of the shortest vector in \(\mathcal{L}\).

- By reduced property of our basis, if \((u_1, \ldots, u_n)\) is the GSO basis we have:

\[
\|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu_{(k+1)k} u_k\|^2
= \frac{4}{3} \cdot \|u_{k+1}\|^2 + \frac{4}{3} \cdot \mu_{(k+1)k}^2 \cdot \|u_k\|^2
\leq \frac{4}{3} \cdot \|u_{k+1}\|^2 + \frac{1}{3} \cdot \|u_k\|^2
\Rightarrow \|u_k\|^2 \leq 2\|u_{k+1}\|^2
\]

\[
\frac{2}{3} \|u_k\|^2 \leq \frac{4}{3} \|u_{k+1}\|^2
\]
Finding Short Vector

- If \((b_1, \ldots, b_n)\) is a reduced basis of \(\mathcal{L}\), then

\[
\|b_1\| \leq 2^{\frac{n-1}{2}} \lambda(\mathcal{L})
\]

where \(\lambda(\mathcal{L})\) is the length of the shortest vector in \(\mathcal{L}\).

- By reduced property of our basis, if \((u_1, \ldots, u_n)\) is the GSO basis we have:

\[
\|b_1\|^2 \leq 2^{k-1} \|u_k\|^2
\]

\[
\|u_k\|^2 \leq \frac{4}{3} \cdot \|u_{k+1} + \mu(k+1)k u_k\|^2
\]

\[
= \frac{4}{3} \cdot \|u_{k+1}\|^2 + \frac{4}{3} \cdot \mu^2(k+1)k \cdot \|u_k\|^2
\]

\[
\leq \frac{4}{3} \cdot \|u_{k+1}\|^2 + \frac{1}{3} \cdot \|u_k\|^2
\]

\[
\Rightarrow \|u_k\|^2 \leq 2\|u_{k+1}\|^2 \quad \leftarrow \text{induction}
\]

- Then our lemma on GSO basis and shortest vector gives us

\[
\|b_1\|^2 \leq \min_k \{2^{k-1}\|u_k\|^2\} \leq 2^{n-1} \cdot \min_k \|u_k\|^2 \leq 2^{n-1} \cdot \lambda(\mathcal{L})^2
\]
Proof Details
Short Vectors in a Lattice

Algorithm Idea: Find Good Basis

Gram-Schmidt Orthogonalization

Lenstra-Lenstra-Lovasz (LLL) Basis Reduction Algorithm

Conclusion

Acknowledgements
Conclusion

In today’s lecture, we learned

- Finding short vector in a lattice
- Finished proof of factoring algorithm over \( \mathbb{Z}[x] \)
- LLL algorithm is useful way beyond factoring!
  1. breaking cryptosystems
  2. finding simultaneous Diophantine approximations
  3. refutation of Mertens’ conjecture
- Great final projects to explore here!
Acknowledgement

Based entirely on

- Lectures 10 and 11 from Madhu's notes
  http://people.csail.mit.edu/madhu/FT98/