Lattice Basis Reduction and the LLL Algorithm

Curtis Bright

May 21, 2009

Point Lattices

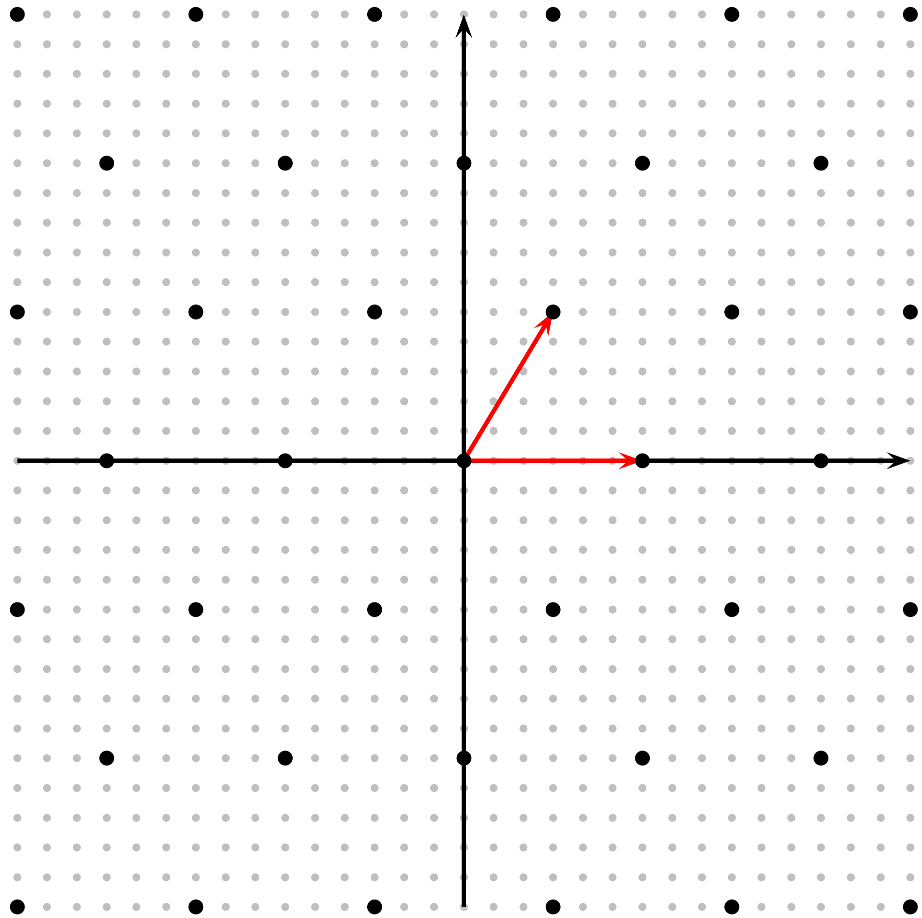
- A *point lattice* is a discrete additive subgroup of $\mathbb{R}^n$.
- A *basis* for a lattice $L \subset \mathbb{R}^n$ is a set of linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d \in \mathbb{R}^n$ whose 'integer span' generates L:

$$L = \left\{ \sum_{i=1}^{d} x_i \boldsymbol{b}_i : x_i \in \mathbb{Z} \right\}$$

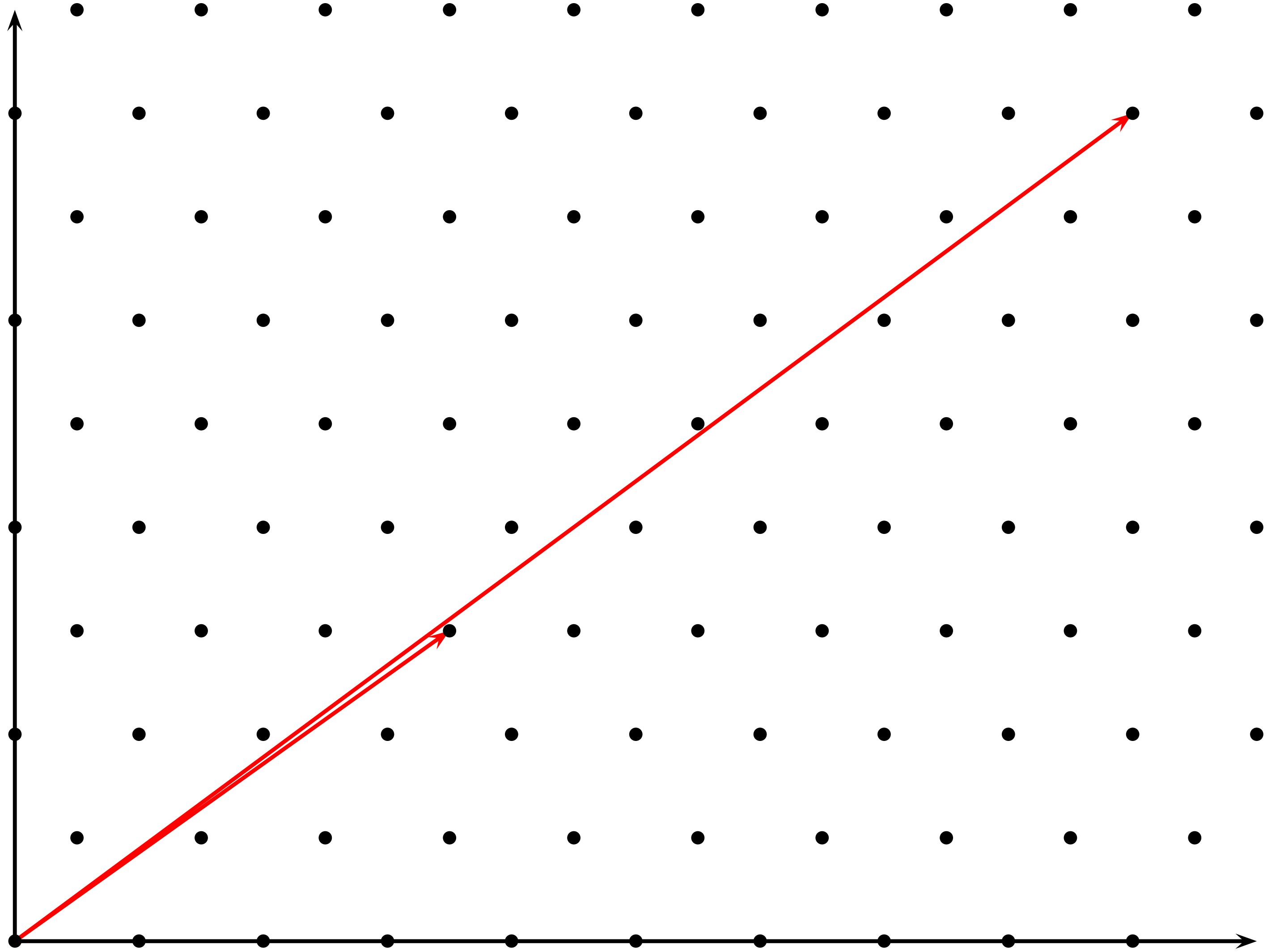
- In particular, we will be concerned about the case when $\boldsymbol{b}_i \in \mathbb{Z}^n$, so $L \subseteq \mathbb{Z}^n$.
- d is the *dimension* of the lattice.

2D Example Lattice

- The lattice generated by $\boldsymbol{b}_1 = \begin{bmatrix} 3 & 5 \end{bmatrix}$ and $\boldsymbol{b}_2 = \begin{bmatrix} 6 & 0 \end{bmatrix}$ in $\mathbb{Z}^2$:



A Bad Basis



Changing Bases

- The lattices in $\mathbb{Z}^4$ generated by the rows of

$$B = \begin{bmatrix} -32 & 27 & 99 & 92 \\ -74 & 8 & 29 & -31 \\ -4 & 69 & 44 & 67 \end{bmatrix}$$

$$B' = \begin{bmatrix} -4339936 & -682927 & -2330272 & -6748685 \\ 268783718 & 42311760 & 144378994 & 418036006 \\ 47833660 & 7038229 & 23910075 & 72218282 \end{bmatrix}$$

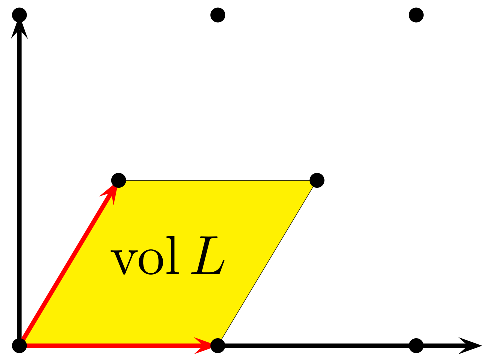
  are the same. This is shown by writing each row in B as a $\mathbb{Z}$-linear combination of the rows of B', and vice versa.

- That is, there exist change-of-basis matrices U and U' with integer entries such that $B' = UB$ and $B = U'B'$.

- Since U and $U' = U^{-1}$ both have integer entries, $\det U$ and $\det U^{-1} = 1/\det U$ are both integers.

- Therefore $\det U = \pm 1$ (U is *unimodular*).

Lattice Volume



- We define the *volume* of a lattice L with basis B to be the volume of the $[0,1)$-span of its basis vectors.
- If B is square then $\operatorname{vol} L = |\det B|$, and in general $\operatorname{vol} L = \sqrt{\det(BB^T)}$.
- This is well defined: if B' is some other basis of L then

$$\sqrt{\det(B'B'^T)} = \sqrt{\det(UBB^TU^T)} = \sqrt{\det(BB^T)}$$

since U is unimodular.

Lattice Reduction

- Some bases are much easier to work with than others. This suggests we try to find:
  - A method of ranking the bases of a lattice in some desirable order.
  - An efficient way to find desirable bases of a lattice when given one of its other bases.

The Best Basis

- The best possible basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ of L would have $\boldsymbol{b}_1$ the shortest possible nonzero vector in L and in general $\boldsymbol{b}_i$ the shortest possible nonzero vector such that $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i$ are linearly independent.
- Of course such vectors always exist, but perhaps surprisingly for $d \geq 4$ they do not necessarily form a basis of L.

- For example, the lattice generated by the following basis:

$$\begin{bmatrix} 2 & & & \\ & \ddots & & \\ & & 2 & \\ 1 & \cdots & 1 & 1 \end{bmatrix} \in \mathbb{Z}^{n \times n}$$

- For $n \geq 5$ the last vector is no longer the shortest possible vector in the lattice; in this case the shortest possible vector has norm 2 and there are exactly n vectors (up to sign) which reach the minimum.
- These vectors are linearly independent but generate $(2\mathbb{Z})^n$ instead.

Minkowski Reduction

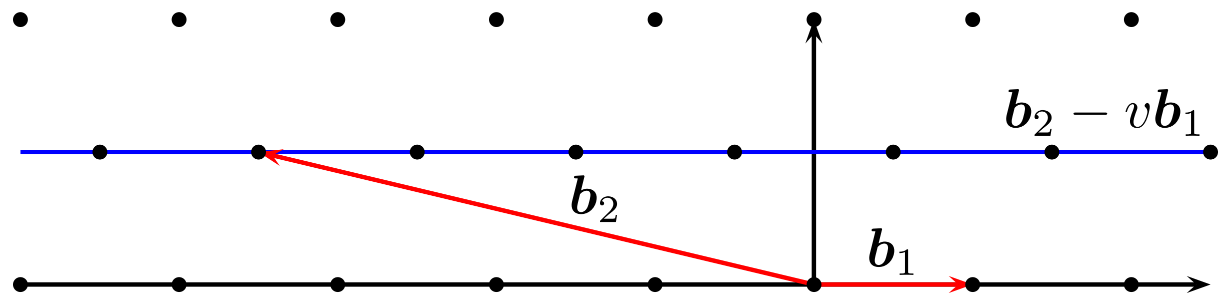
- The next best thing:

  Definition. *A basis* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ *of* L *is Minkowski reduced if* $\boldsymbol{b}_i$ *is the shortest possible vector such that* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i$ *may be extended into a basis of* L *for each* $1 \leq i \leq d$*.*

- This is a greedy definition: it may concede a large increase in later $\boldsymbol{b}_i$ for a small decrease in an early $\boldsymbol{b}_i$.
- Computationally, finding a Minkowski reduced basis leads to a combinatorial problem with a search space exponential in d.
- Even just computing $\boldsymbol{b}_1$ (the *Shortest Vector Problem*) is NP-hard when the maximum norm is used.

Lagrange Reduction

- Historically the first lattice reduction considered (by Lagrange in 1773) was in two dimensions.
- It gives rise to a simple algorithm, rather similar in style to Euclid's famous gcd algorithm: the norms of the input vectors are continually decreased by subtracting appropriate multiples of one vector from the other.
- If $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$ then we want to replace $\boldsymbol{b}_2$ with $\boldsymbol{b}_2 - v\boldsymbol{b}_1$ for some v such that $\|\boldsymbol{b}_2 - v\boldsymbol{b}_1\|$ is minimized.



- Optimally, the new value of $\|\boldsymbol{b}_2 - v\boldsymbol{b}_1\|$ would be

$$\left\|\boldsymbol{b}_2 - \mathrm{proj}_{\boldsymbol{b}_1}(\boldsymbol{b}_2)\right\| = \left\|\boldsymbol{b}_2 - \tfrac{\langle \boldsymbol{b}_2, \boldsymbol{b}_1\rangle}{\|\boldsymbol{b}_1\|^2}\boldsymbol{b}_1\right\|.$$

- But it is essential that $v \in \mathbb{Z}$, so take

$$v := \left\lfloor \tfrac{\langle \boldsymbol{b}_2, \boldsymbol{b}_1\rangle}{\|\boldsymbol{b}_1\|^2} \right\rceil.$$

- In the case $\left|\frac{\langle \boldsymbol{b}_2, \boldsymbol{b}_1\rangle}{\|\boldsymbol{b}_1\|^2}\right| \leq \frac{1}{2}$ there is no multiplier we can use to strictly decrease the norm.

Definition. *A basis $\boldsymbol{b}_1, \boldsymbol{b}_2$ of L is Lagrange reduced if $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$ and $\left|\frac{\langle \boldsymbol{b}_2, \boldsymbol{b}_1\rangle}{\|\boldsymbol{b}_1\|^2}\right| \leq \frac{1}{2}$.*

- Repeatedly applying this form of reduction yields Algorithm 1.3.14 in Cohen's text:

  Input: A basis $\boldsymbol{b}_1, \boldsymbol{b}_2$ of a lattice L

  Output: A Lagrange reduced basis of L

  repeat

  if $\|\boldsymbol{b}_1\| > \|\boldsymbol{b}_2\|$ **then** swap $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$

  $\boldsymbol{b}_2 := \boldsymbol{b}_2 - \left\lfloor \frac{\langle \boldsymbol{b}_2, \boldsymbol{b}_1 \rangle}{\|\boldsymbol{b}_1\|^2} \right\rceil \boldsymbol{b}_1$

  until $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$

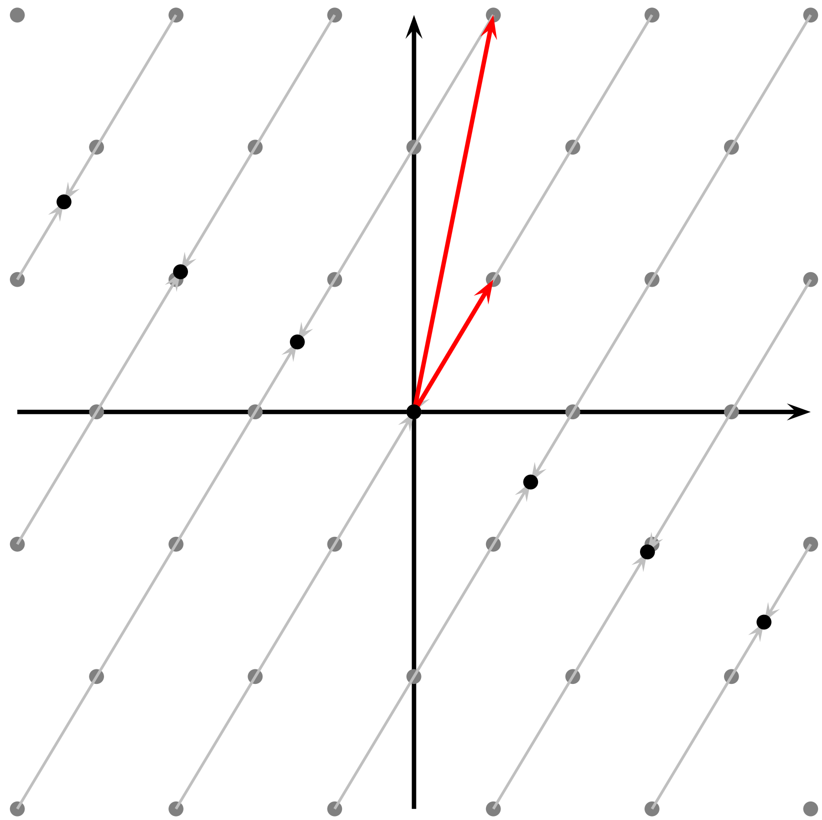
  return $(\boldsymbol{b}_1, \boldsymbol{b}_2)$

- $\|\boldsymbol{b}_2\|$ decreases by at least a factor of $\sqrt{3}$ on every iteration (except possibly the first and last).

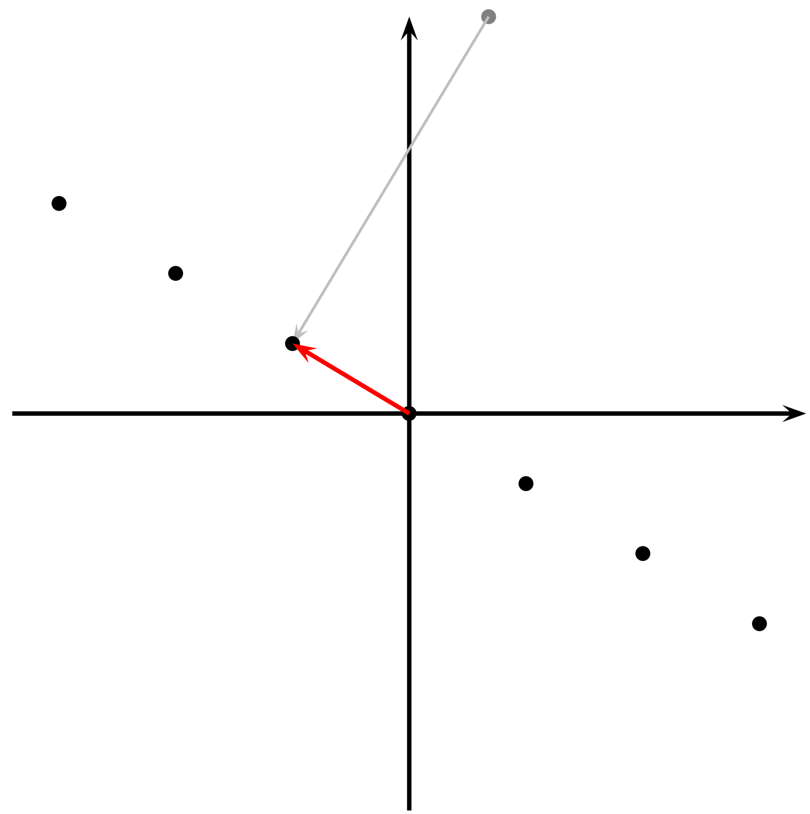
- Since $\|\boldsymbol{b}_2\|$ is always at least 1, there are $O(\log_{\sqrt{3}}\|\boldsymbol{b}_2\|)$ iterations.

- The arithmetic operations in each loop take $O(\log^2\|\boldsymbol{b}_2\|)$, so this algorithm runs in time $O(\log^3\|\boldsymbol{b}_2\|)$.

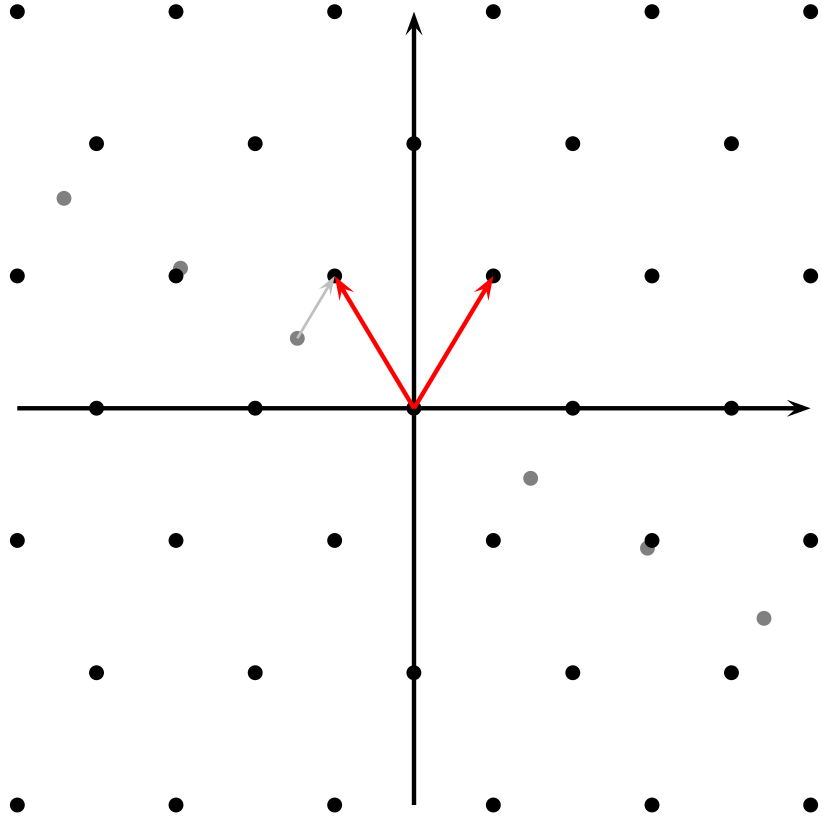
- Equivalently, we may consider Lagrange's algorithm as if it was using a *projected lattice*:



- Let L' be the lattice L projected orthogonally to $\boldsymbol{b}_1$. Then $d = 1$, so L' has only one basis up to sign:



- Now ‘lift’ the basis for L' into L. Of course, there are an infinite number ways to lift; we choose the shortest.



Korkin-Zolotarev Reduction

- The advantage to considering Lagrange's algorithm this way is that it generalizes to higher dimensions.
- Let $\boldsymbol{b}'_i$ be the component of $\boldsymbol{b}_i$ orthogonal to $\boldsymbol{b}_1$, i.e.,

$$\boldsymbol{b}'_i = \mathrm{proj}_{\mathrm{span}(\boldsymbol{b}_1)^\perp}(\boldsymbol{b}_i) = \boldsymbol{b}_i - \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_1 \rangle}{\|\boldsymbol{b}_1\|^2}\boldsymbol{b}_1 = \boldsymbol{b}_i - \mu_{i,1}\boldsymbol{b}_1.$$

  Definition. *A basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ of L is Korkin-Zolotarev reduced if*

  - *$\boldsymbol{b}_1$ is the shortest possible nonzero vector of L*
  - *$\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_d$ is a Korkin-Zolotarev reduced basis of L'*
  - *$\boldsymbol{b}_2, \ldots, \boldsymbol{b}_d$ are lifted from L' minimally: $|\mu_{i,1}| \leq \frac{1}{2}$ for $2 \leq i$*

- Once again, this reduction notion requires solving SVP to find a Korkin-Zolotarev reduced basis—not good computationally.

- There are d recursive lattices in this definition:

$$L \text{ with basis } \boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$$

$$L' \text{ with basis } \boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_d$$

$$L^{(2)} \text{ with basis } \boldsymbol{b}_3^{(2)}, \ldots, \boldsymbol{b}_d^{(2)}$$

$$\vdots$$

$$L^{(d-1)} \text{ with basis } \boldsymbol{b}_d^{(d-1)}$$

- Denote $\boldsymbol{b}_i^{(i-1)}$ by $\boldsymbol{b}_i^*$. By induction it may be shown

$$\boldsymbol{b}_i^* = \mathrm{proj}_{\mathrm{span}(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{i-1}^*)^\perp}(\boldsymbol{b}_i).$$

- These are the *Gram-Schmidt orthogonalization* vectors. $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_i^*$ is an orthogonal basis for $\mathrm{span}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i)$.

Orthogonality Defect

- By the Gram-Schmidt orthogonalization,

$$\operatorname{vol} L = \prod_{i=1}^{d} \|\boldsymbol{b}_i^*\| \leq \prod_{i=1}^{d} \|\boldsymbol{b}_i\|$$

  with equality if and only if the $\boldsymbol{b}_i$ are orthogonal.

- The larger $\prod_{i=1}^{d} \|\boldsymbol{b}_i\|$ is compared to $\operatorname{vol} L$ the less orthogonal the $\boldsymbol{b}_i$ are. So $\prod_{i=1}^{d} \|\boldsymbol{b}_i\| / \operatorname{vol} L$ is known as the *orthogonality defect*, and is a method of ranking the bases of a lattice.

- We would like a guarantee that the reductions we consider have an orthogonality defect bounded by some function of d:

$$\prod_{i=1}^{d} \|\boldsymbol{b}_i\| \leq f(d) \operatorname{vol} L.$$

Hermite Reduction

- Historically, Hermite was the first to consider lattice reduction in arbitrary dimension in two letters sent to Jacobi in 1845.
- Hermite reduction is weaker than Korkin-Zolotarev reduction, but stronger than LLL reduction.
- Nevertheless, the properties we will show for Hermite reduced bases also apply to LLL reduced bases (with small modifications).

Definition. *A basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ of L is Hermite reduced if*

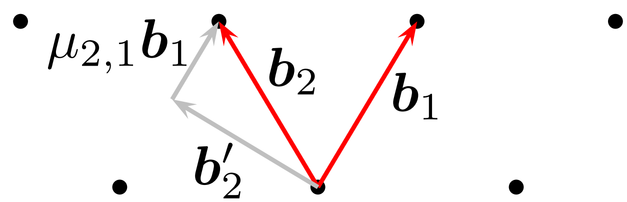
- $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_i\|$ *for all* i
- $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_d$ *is a Hermite reduced basis of* L'
- $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_d$ *are lifted from* L' *minimally:* $|\mu_{i,1}| \leq \frac{1}{2}$ *for* $2 \leq i$

A Nice Bound

- Hermite reduced bases satisfy the following bound:

$$\|\boldsymbol{b}_i\|^2 \leq \frac{4}{3}\|\boldsymbol{b}'_i\|^2$$

- Intuitively this says that the projected vector $\boldsymbol{b}'_i$ isn't that much smaller than the original $\boldsymbol{b}_i$.
- Actually follows from the Pythagorean Theorem in d dimensions and the fact $\|\mu_{i,1}\boldsymbol{b}_1\| \leq \frac{1}{2}\|\boldsymbol{b}_i\|$.



- Using the Pythagorean Theorem,

$$\begin{aligned} \|\boldsymbol{b}_i\|^2 &= \|\boldsymbol{b}'_i\|^2 + \|\mu_{i,1}\boldsymbol{b}_1\|^2 \\ &\le \|\boldsymbol{b}'_i\|^2 + \tfrac{1}{4}\|\boldsymbol{b}_i\|^2 \\ \tfrac{3}{4}\|\boldsymbol{b}_i\|^2 &\le \|\boldsymbol{b}'_i\|^2 \\ \|\boldsymbol{b}_i\|^2 &\le \tfrac{4}{3}\|\boldsymbol{b}'_i\|^2 \\ &\le \left(\tfrac{4}{3}\right)^2 \left\|\boldsymbol{b}_i^{(2)}\right\|^2 \\ &\vdots \\ &\le \left(\tfrac{4}{3}\right)^{i-1} \|\boldsymbol{b}_i^*\|^2 \end{aligned}$$

by repeated application of the bound.

- Intuitively, as i increases $\boldsymbol{b}_i^*$ is allowed to become increasingly smaller than $\boldsymbol{b}_i$, but not arbitrarily smaller.

- From $\|\boldsymbol{b}_i\| \leq \left(\frac{4}{3}\right)^{(i-1)/2} \|\boldsymbol{b}_i^*\|$ we can bound the orthogonality defect:

$$\prod_{i=1}^{d} \|\boldsymbol{b}_i\| \leq \prod_{i=1}^{d} \left(\tfrac{4}{3}\right)^{(i-1)/2} \|\boldsymbol{b}_i^*\| = \left(\tfrac{4}{3}\right)^{\sum_{i=1}^{d}(i-1)/2} \operatorname{vol} L = \left(\tfrac{4}{3}\right)^{d(d-1)/4} \operatorname{vol} L$$

Approximate Shortest Vector Problem

- Hermite reduced bases can also be used to *approximate* a solution to SVP.
- Let $\boldsymbol{x} = \sum_{i=1}^{k} r_i \boldsymbol{b}_i$ be a shortest nonzero vector in L (i.e., a solution to SVP), where $r_i \in \mathbb{Z}$ and $r_k \neq 0$.
- It is difficult to bound a sum of $\boldsymbol{b}_i$ directly since they are not orthogonal. So we rewrite using Gram-Schmidt:

$$\boldsymbol{x} = \sum_{i=1}^{k} r_i \left(\boldsymbol{b}_i^* + \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^* \right) = r_k \boldsymbol{b}_k^* + \sum_{i=1}^{k-1} s_i \boldsymbol{b}_i^*$$

for some $s_i \in \mathbb{Q}$.

- Now we can use a generalization of the Pythagorean Theorem,

$$\|\boldsymbol{x}\|^2 = \|r_k \boldsymbol{b}_k^*\|^2 + \sum_{i=1}^{k-1} \|s_i \boldsymbol{b}_i^*\|^2 \geq r_k^2 \|\boldsymbol{b}_k^*\|^2 \geq \|\boldsymbol{b}_k^*\|^2.$$

- Using previous bounds on $\boldsymbol{b}_i$ with $i = k$,

$$\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_k\| \leq \left(\tfrac{4}{3}\right)^{(k-1)/2} \|\boldsymbol{b}_k^*\| \leq \left(\tfrac{4}{3}\right)^{(d-1)/2} \|\boldsymbol{x}\|.$$

- So $\boldsymbol{b}_1$ is at most a factor of $\left(\tfrac{4}{3}\right)^{(d-1)/2}$ longer than the shortest possible nonzero vector in L.

Optimal-LLL Reduction

- There is no algorithm known which can provably compute a Hermite reduced basis efficiently (polynomial time in d). So, we weaken the conditions again:

  Definition. *A basis* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ *of* L *is optimal-LLL reduced if*

  - $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$
  - $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_d$ *is an optimal-LLL reduced basis of* L'
  - $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_d$ *are lifted from* L' *minimally:* $|\mu_{i,1}| \leq \frac{1}{2}$ *for* $2 \leq i$

- Optimal-LLL reduced bases no longer satisfy the nice bound $\|\boldsymbol{b}_i\|^2 \leq \frac{4}{3}\|\boldsymbol{b}'_i\|^2$, but do satisfy a similar one,

$$\|\boldsymbol{b}_i^*\|^2 \leq \tfrac{4}{3}\|\boldsymbol{b}_{i+1}^*\|^2.$$

- In fact, with a little more work we can derive the same properties as in the Hermite case:

$$\|\boldsymbol{b}_i\| \leq \left(\tfrac{4}{3}\right)^{(i-1)/2} \|\boldsymbol{b}_i^*\|$$

$$\prod_{i=1}^{d}\|\boldsymbol{b}_i\| \leq \left(\tfrac{4}{3}\right)^{d(d-1)/4} \operatorname{vol} L$$

$$\|\boldsymbol{b}_1\| \leq \left(\tfrac{4}{3}\right)^{(d-1)/2} \|\boldsymbol{x}\|$$

- There is no algorithm known which can provably compute an optimal-LLL reduced basis efficiently (polynomial time in d).

LLL Reduction

- We weaken optimal-LLL reduction by allowing some slack room in the $\|\boldsymbol{b}_1\| \leq \|\boldsymbol{b}_2\|$ condition:

  Definition. *A basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ of L is LLL reduced with quality parameter $c \in (1, 4)$ if*

  - $\|\boldsymbol{b}_1\| \leq \sqrt{c}\,\|\boldsymbol{b}_2\|$
  - $\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_d$ *is an LLL reduced basis of L' (with quality c)*
  - $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_d$ *are lifted from L' minimally: $|\mu_{i,1}| \leq \frac{1}{2}$ for $2 \leq i$*

- The smaller c is, the less slack room and the better the reduction.

- Define $C = \frac{4c}{4-c}$; note that $C > \frac{4}{3}$ for $c > 1$ but we can set C arbitrarily close to $\frac{4}{3}$.
- Analogously to the Hermite case, LLL reduced bases satisfy:

$$\|\boldsymbol{b}_i\| \leq C^{(i-1)/2} \|\boldsymbol{b}_i^*\|$$

$$\prod_{i=1}^{d} \|\boldsymbol{b}_i\| \leq C^{d(d-1)/4} \operatorname{vol} L$$

$$\|\boldsymbol{b}_1\| \leq C^{(d-1)/2} \|\boldsymbol{x}\|$$

- In the original LLL paper $c = \frac{4}{3}$ was used, so $C = 2$.

The Punchline

- The straightforward way of applying the definition of an LLL reduced basis gives an algorithm for computing an LLL reduced basis efficiently (polynomial time in d).

Input: A basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ of a lattice L; a quality parameter c
Output: An LLL reduced basis of L (with quality c)

  if $d = 1$ **then return** $(\boldsymbol{b}_1)$
  repeat
    if $\|\boldsymbol{b}_1\| > \sqrt{c}\,\|\boldsymbol{b}_2\|$ **then** swap $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$
    $(\boldsymbol{b}_2, \ldots, \boldsymbol{b}_d) := \mathrm{lift}_{\boldsymbol{b}_1}(\mathrm{LLLR{\small EDUCE}}_c(\boldsymbol{b}'_2, \ldots, \boldsymbol{b}'_d))$
  until $\|\boldsymbol{b}_1\| \le \sqrt{c}\,\|\boldsymbol{b}_2\|$
  return $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$

The Iterative LLL Definition: Size Reduction

- The shortest-lift condition in the jth recursive lattice is $\left|\mu_i^{(j)}\right| \leq \frac{1}{2}$ for $j+1 < i$, where:

$$\mu_i^{(j)} = \frac{\left\langle \boldsymbol{b}_i^{(j)}, \boldsymbol{b}_{j+1}^{(j)} \right\rangle}{\left\| \boldsymbol{b}_{j+1}^{(j)} \right\|^2} = \frac{\left\langle \boldsymbol{b}_i - \sum_{k=1}^{j} \mu_{i,k} \boldsymbol{b}_k^*, \boldsymbol{b}_{j+1}^* \right\rangle}{\left\| \boldsymbol{b}_{j+1}^* \right\|^2}$$

$$= \frac{\left\langle \boldsymbol{b}_i, \boldsymbol{b}_{j+1}^* \right\rangle}{\left\| \boldsymbol{b}_{j+1}^* \right\|^2}$$

$$= \mu_{i,j+1}$$

- So the shortest-lift condition implies $|\mu_{i,j}| \leq \frac{1}{2}$ for $j < i$.
- This is called *size-reduction*.

The Iterative LLL Definition: Lovász Condition

- The $\|\boldsymbol{b}_1\| \leq \sqrt{c}\,\|\boldsymbol{b}_2\|$ condition in the ith recursive lattice:

$$\begin{aligned} \left\|\boldsymbol{b}_{i+1}^{(i)}\right\| &\leq \sqrt{c}\,\left\|\boldsymbol{b}_{i+2}^{(i)}\right\| \\ &= \sqrt{c}\,\left\|\boldsymbol{b}_{i+2} - \sum_{j=1}^{i} \mu_{i+2,j}\boldsymbol{b}_j^*\right\| \\ &= \sqrt{c}\,\|\boldsymbol{b}_{i+2}^* + \mu_{i+2,i+1}\boldsymbol{b}_{i+1}^*\| \end{aligned}$$

- So the $\boldsymbol{b}_1$-bound condition implies $\|\boldsymbol{b}_i^*\| \leq \sqrt{c}\,\|\boldsymbol{b}_{i+1}^* + \mu_{i+1,i}\boldsymbol{b}_i^*\|$ for $i \geq 1$.

- This is called the *Lovász condition*.

Non-recursive LLL Reduction

- Putting these conditions together gives Definition 2.6.1 in Cohen's text:

  Definition. *A basis* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ *is LLL reduced with quality parameter* $c \in (1, 4)$ *if*

  - $|\mu_{i,j}| \leq \frac{1}{2}$ *for* $1 \leq j < i \leq d$
  - $\|\boldsymbol{b}^*_{i-1}\| \leq \sqrt{c}\,\|\boldsymbol{b}^*_i + \mu_{i,i-1}\boldsymbol{b}^*_{i-1}\|$ *for* $1 < i \leq d$

- Say we have some basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ such that the first $k - 1$ vectors form an LLL reduced basis. If

  - $\boldsymbol{b}_k$ is size-reduced against the first $k - 1$ vectors
  - the Lovász condition holds for $i = k$

  then $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ is also an LLL reduced basis.

The Iterative LLL Algorithm

Input: A basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ of a lattice L; a quality parameter c

Output: An LLL reduced basis of L (with quality c)

$k := 2$

while $k \leq d$ **do**

size-reduce $\boldsymbol{b}_k$ against $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k-1}$

if $\|\boldsymbol{b}_{k-1}^*\| \leq \sqrt{c}\,\|\boldsymbol{b}_k^* + \mu_{k,k-1}\boldsymbol{b}_{k-1}^*\|$ **then**

$k := k + 1$

else

swap $\boldsymbol{b}_{k-1}$ and $\boldsymbol{b}_k$

$k := \max(k - 1, 2)$

end if

end while

return $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$

- At the start of the loop, $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k-1}$ is an LLL reduced basis.

The Gram-Schmidt Vectors During LLL

- Size reduction does not change the $\boldsymbol{b}_i^*$.
- If $\boldsymbol{c}_i^*$ are the Gram-Schmidt vectors after a swap, then:

Before		After
$\boldsymbol{b}_1$	$\|\boldsymbol{b}_1^*\| = \|\boldsymbol{c}_1^*\|$	$\boldsymbol{b}_1$
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{b}_{k-1}$	$\|\boldsymbol{b}_{k-1}^*\| = \|\boldsymbol{c}_{k-1}^*\|$	$\boldsymbol{b}_{k-1}$
$\boldsymbol{b}_k$	$\|\boldsymbol{b}_k^*\| > \sqrt{c}\,\|\boldsymbol{c}_k^*\|$	$\boldsymbol{b}_{k+1}$
$\boldsymbol{b}_{k+1}$	$\|\boldsymbol{b}_{k+1}^*\| < \sqrt{c}\,\|\boldsymbol{c}_{k+1}^*\|$	$\boldsymbol{b}_k$
$\boldsymbol{b}_{k+2}$	$\|\boldsymbol{b}_{k+2}^*\| = \|\boldsymbol{c}_{k+2}^*\|$	$\boldsymbol{b}_{k+2}$
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{b}_d$	$\|\boldsymbol{b}_d^*\| = \|\boldsymbol{c}_d^*\|$	$\boldsymbol{b}_d$

Bounding the Number of Swaps

- Let B_k be the basis consisting of the first k basis vectors, L_k the lattice formed by the basis B_k, and

$$d_k = (\mathrm{vol}\, L_k)^2 = \det(B_k B_k^T) = \prod_{i=1}^{k} \|\boldsymbol{b}_i^*\|^2.$$

- If the $\boldsymbol{b}_i$ are integer vectors then $d_k \in \mathbb{Z}^+$.
- During LLL, a swap of $\boldsymbol{b}_k$ and $\boldsymbol{b}_{k+1}$ decreases d_k by a factor of at least c, and doesn't change d_i for $i \neq k$.
- Thus, if we define

$$D = \prod_{i=1}^{d} d_i$$

then D decreases by a factor of at least c after every swap.

- Thus, there are at most $\log_c(D)$ swaps. Since

$$D = \prod_{i=1}^{d} \|\boldsymbol{b}_i^*\|^{2(d-i+1)} \leq \prod_{i=1}^{d} \|\boldsymbol{b}_i\|^{2(d-i+1)} \leq \max_i \|\boldsymbol{b}_i\|^{d(d+1)}$$

  there are $O(\log D) = O(d^2 \log B)$ swaps, where $B = \max_i \|\boldsymbol{b}_i\|$ for the original $\boldsymbol{b}_i$.

- The size of the numbers involved remain reasonable throughout the algorithm:
  - $\|\boldsymbol{b}_i^*\| \leq B$.
  - The denominators of $\boldsymbol{b}_i^*$ and $\mu_{i,j}$ divide $\operatorname{vol} L$.
  - $\log\|\boldsymbol{b}_i\|$ and $\log|\mu_{i,j}|$ are $O(d \log B)$.

- Size-reduction requires $O(n)$ arithmetic operations, and there are $O(d)$ vectors to size-reduce against.

- Total cost of LLL is therefore $O(nd^5(\log B)^3)$ without fast arithmetic.

Factoring Polynomials over the Integers

- If f is an integer polynomial with an algebraic root, if we can find the minimal polynomial of that root then we have an irreducible factor of f.
- Let $\alpha \in \mathbb{C}$ be an approximation to a algebraic root of f with a minimal polynomial h of degree m.

- For some constant N let L be the lattice generated by the rows of the following basis:

$$\begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_m \end{bmatrix} = \begin{bmatrix} 1 & & & & & N\,\Re(\alpha^0) & N\,\Im(\alpha^0) \\ & 1 & & & & N\,\Re(\alpha^1) & N\,\Im(\alpha^1) \\ & & 1 & & & N\,\Re(\alpha^2) & N\,\Im(\alpha^2) \\ & & & \ddots & & \vdots & \vdots \\ & & & & 1 & N\,\Re(\alpha^m) & N\,\Im(\alpha^m) \end{bmatrix}$$

- Any $\boldsymbol{x} \in L$ has form $\boldsymbol{x} = \sum_{i=0}^{m} g_i \boldsymbol{b}_i$ for some $g_i \in \mathbb{Z}$.
- Can think of $(g_0, \ldots, g_m)$ as $\boldsymbol{g} \in \mathbb{Z}^m$ or an integer polynomial $g(x) = \sum_{i=0}^{m} g_i x^i$.

- Any $\boldsymbol{x} \in L$ has the form

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{g}^T & N\,\Re(g(\alpha)) & N\,\Im(g(\alpha)) \end{bmatrix},$$

and it follows $\|\boldsymbol{x}\|^2 = \|\boldsymbol{g}\|^2 + N^2|g(\alpha)|^2$.

- We can make $h(\alpha)$ arbitrarily small by increasing the precision of α.

- So by taking N large enough, we can make the shortest nonzero vector in L be

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{h}^T & N\,\Re(h(\alpha)) & N\,\Im(h(\alpha)) \end{bmatrix}.$$

- And then increasing N by a factor $\approx 2^{m/2}$ ensures that any vector $\boldsymbol{x} \in L$ not a multiple of $\boldsymbol{s}$ will have $\|\boldsymbol{x}\|^2 > 2^m \|\boldsymbol{s}\|^2$.

- LLL will always find a vector $\|\boldsymbol{b}_0\|^2 \leq 2^m \|\boldsymbol{s}\|^2$.