Lattice Basis Reduction

Bounds and Algorithms

Matthew Cary

October 15, 2003

GCD

$$\text{gcd}(a,b) = \text{min}^{+}\{|x \cdot a + y \cdot b| : x, y \in \mathbb{Z}\}$$

- ⋆ GCD is the *minimum* nonzero element of a discrete set
- ⋆ Euclidean algorithm computes this by iteratively subtracting a and b from each other

A Generalized GCD

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix}$$

$$\lambda(B) = \mathrm{min}^{+}\{\|B \cdot x\| : x \in \mathbb{Z}^n\}$$

- ⋆ The set $\{B \cdot x : x \in \mathbb{Z}^n\}$ is called a *lattice*
- ⋆ Computing $\lambda(B)$ is NP-hard
- ⋆ *Approximation* is active field of research
  - ◇ NP-hard to approximate to a constant [Micciancio '98, Ajtai '98].
  - ◇ Polynomial-time algorithms to find a reduced basis that approximates the shortest vector to $(1+\epsilon)^n$ [LLL '82, Schnorr '87]

Lattice Applications

- ⋆ Direct application
  - ◇ Knapsack cryptosystems
  - ◇ Integer programming with a fixed number of variables
- ⋆ Linear approximation of nonlinear systems
  - ◇ Small roots of modular polynomials
  - ◇ Truncated linear congruential generators
- ⋆ Number theory
  - ◇ Factoring integer polynomials
  - ◇ Small integer relations

Applications of Lattice Basis Reduction

- ⋆ [Shamir '82] used fixed-dimension IP algorithm [Lenstra '83] based on LLL to break Merkle-Hellman Knapsack cryptosystem ['78].

- ⋆ The Chor-Rivest Subset Sum cryptosystem ['84] was broken in dimension 103 [Schnorr, Horner '95], using low-density subset sum lattices (suggested dimension is ~ 200).

- ⋆ Low-density subset sum can be solved up to density $0.9408\ldots$

- ⋆ Other classic applications: Factoring polynomials over $\mathbb{Z}$, finding small integer relations, attacking low-degree RSA, breaking truncated linear congruential pseudo-random number generators, bounding bits leaked by RSA.

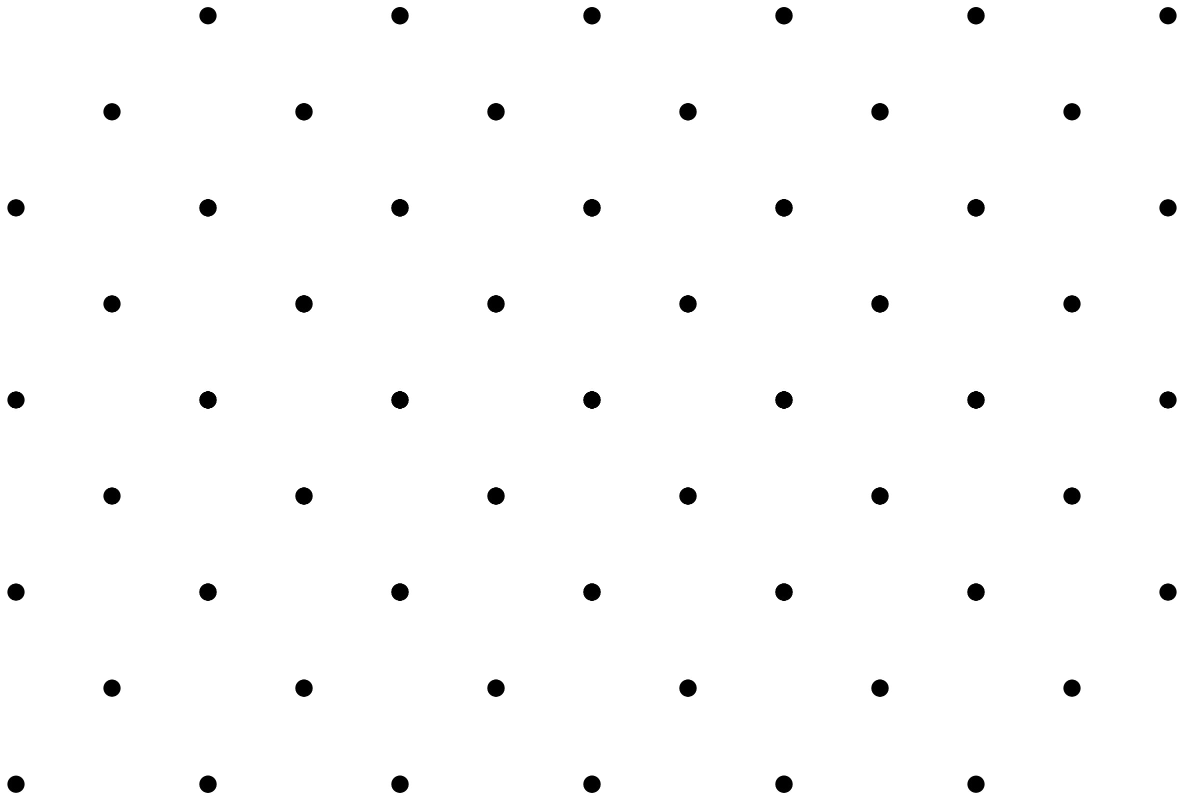
Outline

1. Elementary bounds

2. Reduction algorithms

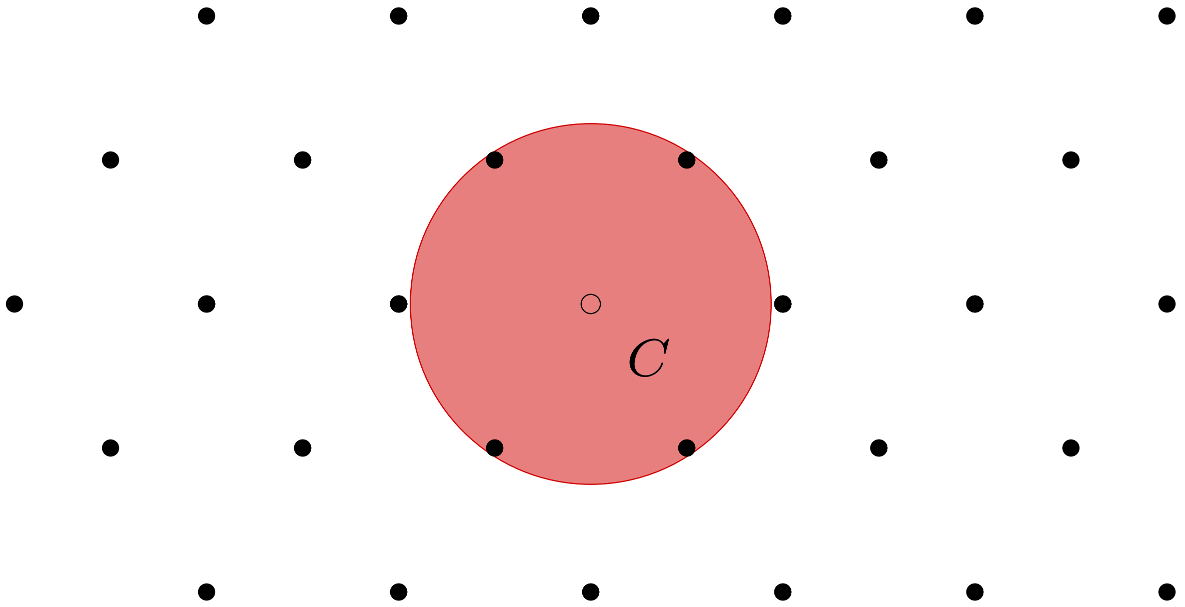
3. (My) current research

A Lattice: Geometrically



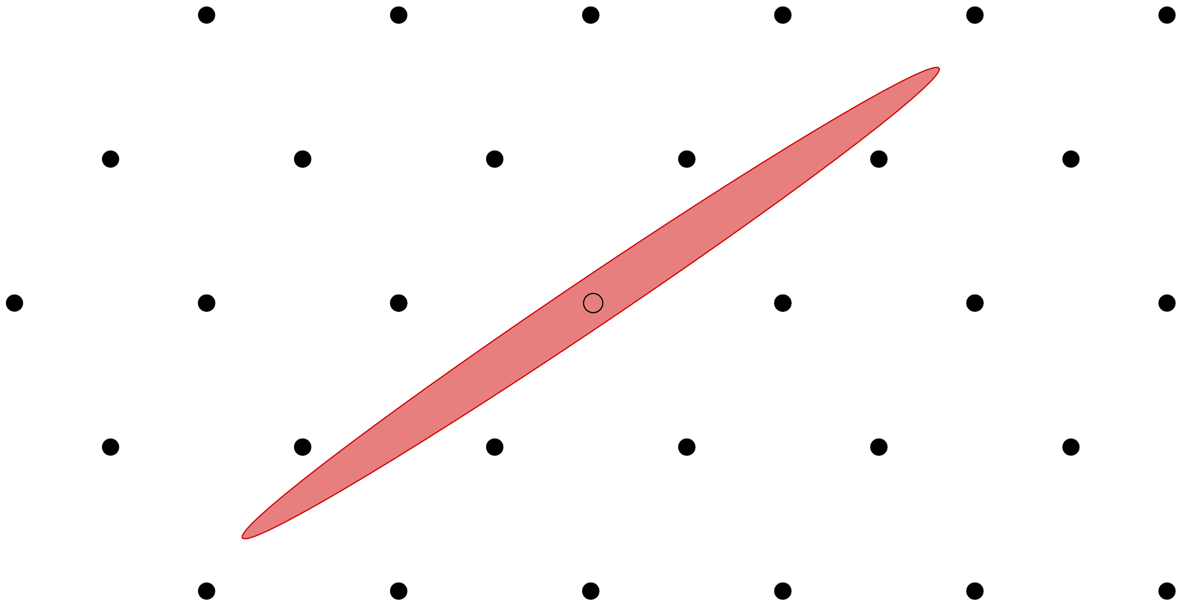
$$\mathcal{L} = \mathcal{L}(B)$$

A Symmetric Convex Body in a Lattice

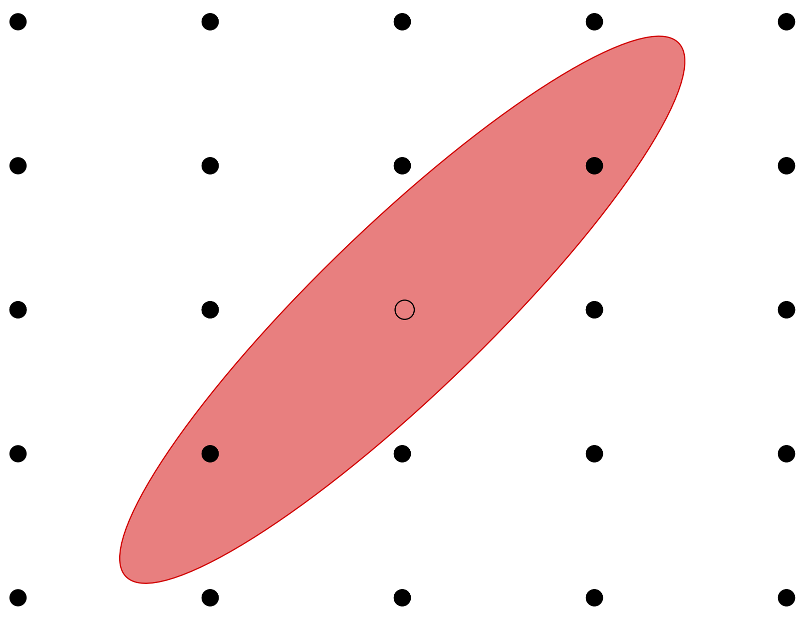


How big can C be before containing a lattice point (other than the origin)?

A Tricky Symmetric Convex Body in a Lattice

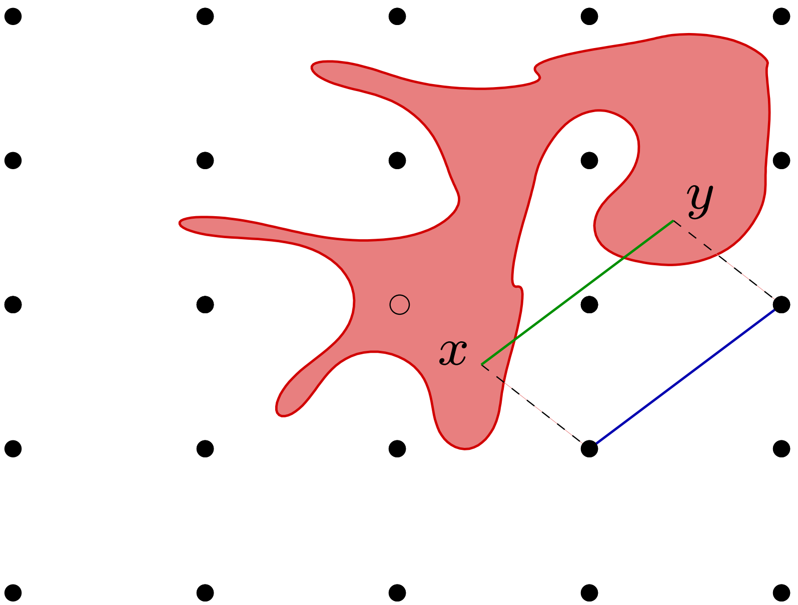


Minkowski's Convex Body Theorem



Any convex body symmetric about the origin in $\mathbb{R}^n$ with volume greater than 2^n, contains a nonzero point of $\mathbb{Z}^n$

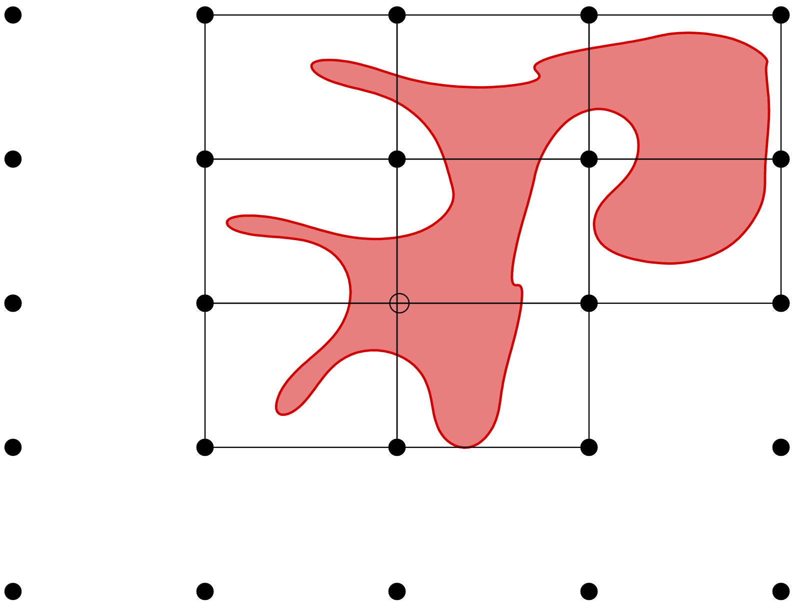
Blichfeldt's Lemma



Let $\mathcal{M}$ be any bounded open set with volume > 1. Then $\mathcal{M}$ contains two points x and y with $x - y \in \mathbb{Z}^n$

Proof of Blichfeldt's Lemma

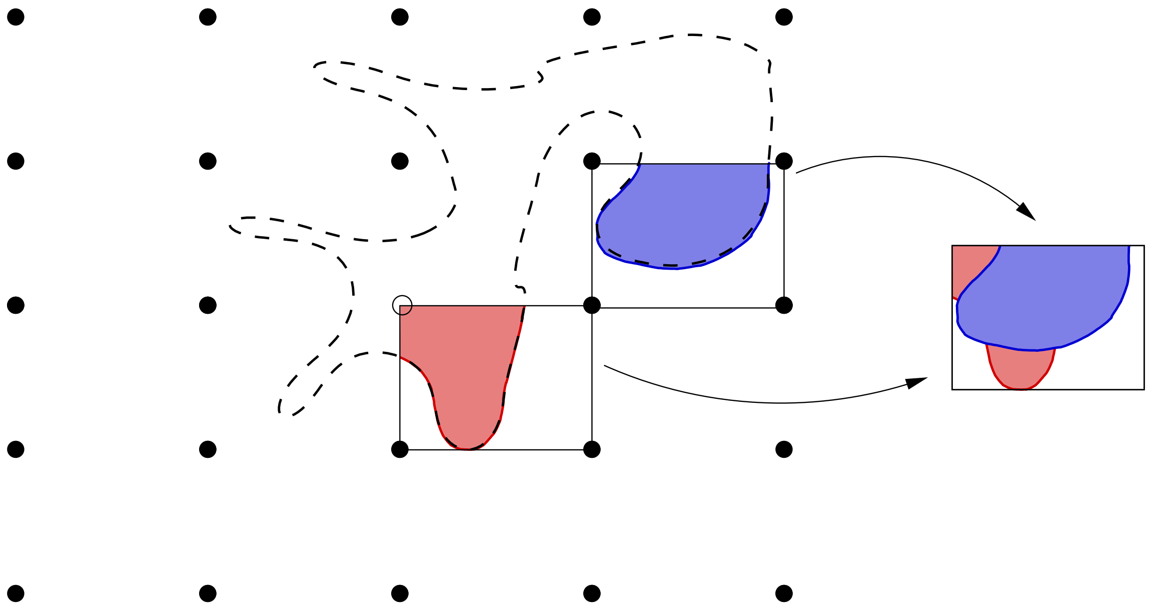
Step 1



⋆ Divide $\mathcal{M}$ based on unit squares

Proof of Blichfeldt's Lemma

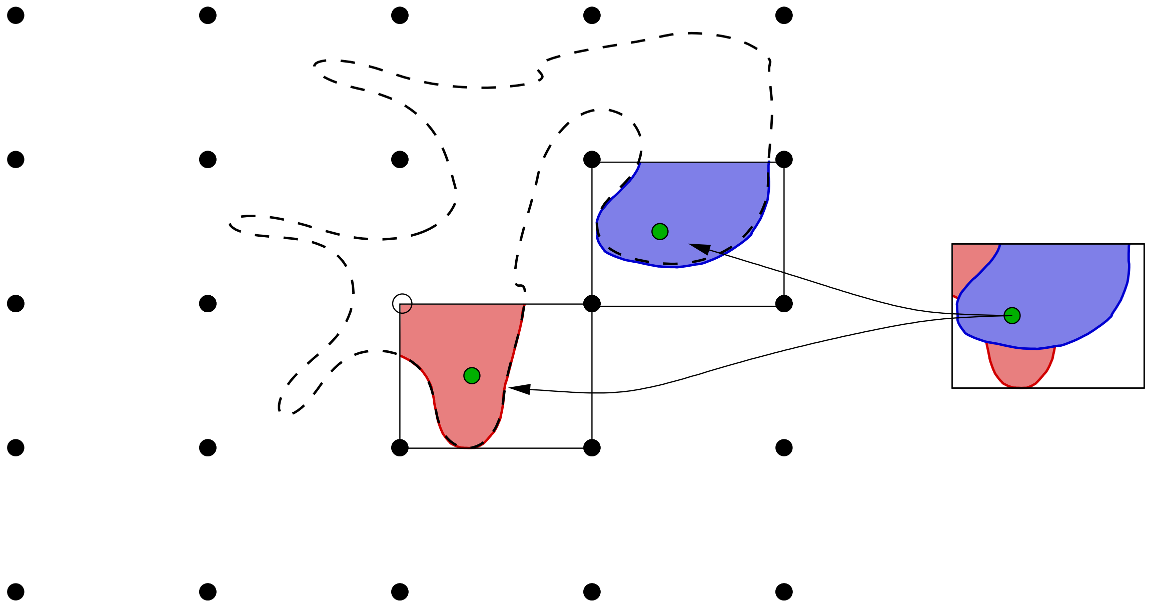
Step 2



⋆ As volume > 1, two regions must overlap

Proof of Blichfeldt's Lemma

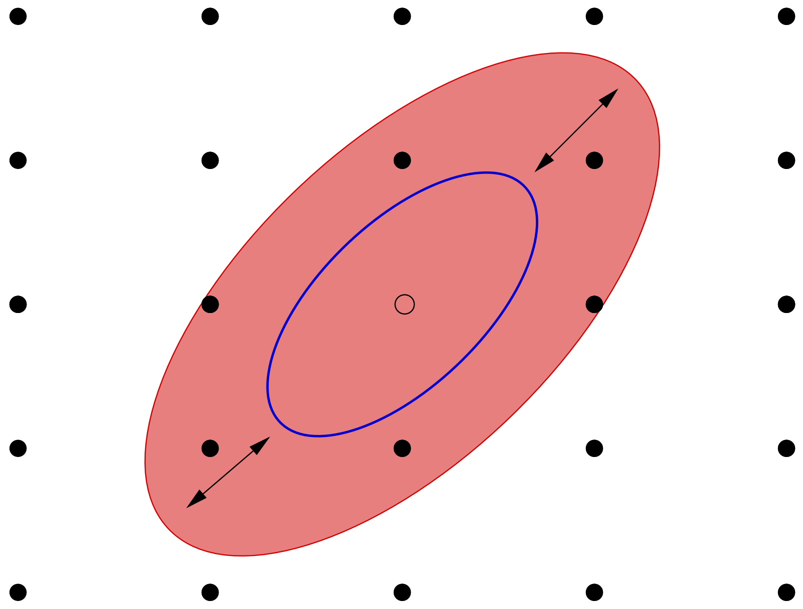
Step 3



⋆ The overlap points differ by a vector in $\mathbb{Z}^n$ □

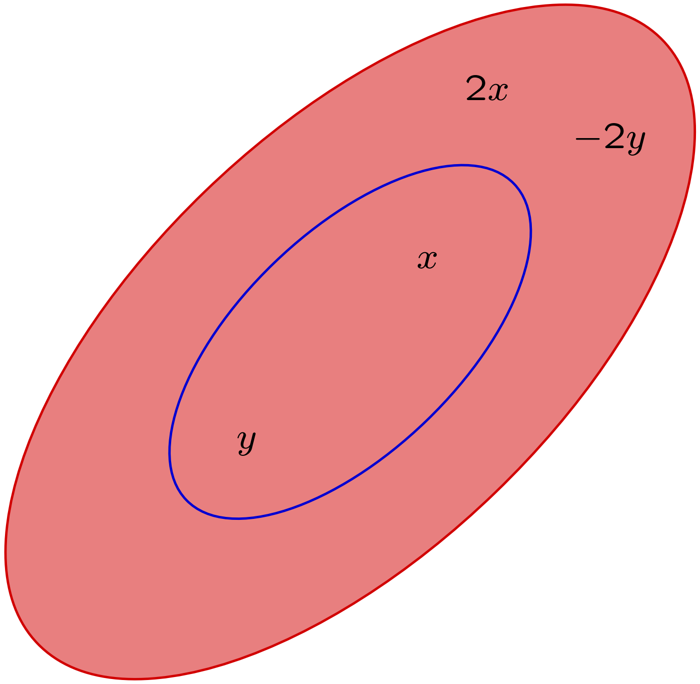
Minkowski's Convex Body Theorem

Any convex body symmetric about the origin in $\mathbb{R}^n$ with volume greater than 2^n, contains a nonzero point of $\mathbb{Z}^n$



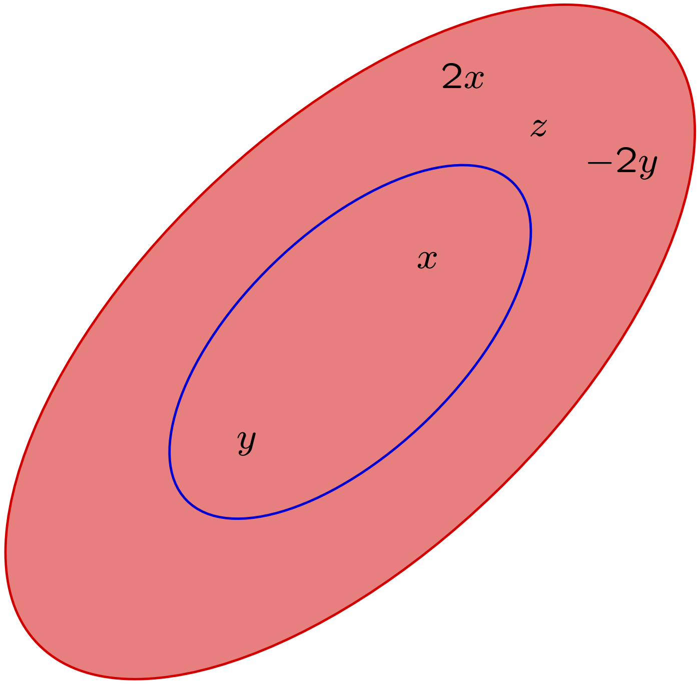
- $\star$ Shrink C by factor of 2 in every direction
- $\star$ Resulting volume > 1, so Blichfeldt's lemma applies

Minkowski's Convex Body Theorem



- ⋆ $x - y \in \mathbb{Z}^n$
- ⋆ $2x, 2y, -2y \in C$ as factor of 2 larger and symmetric

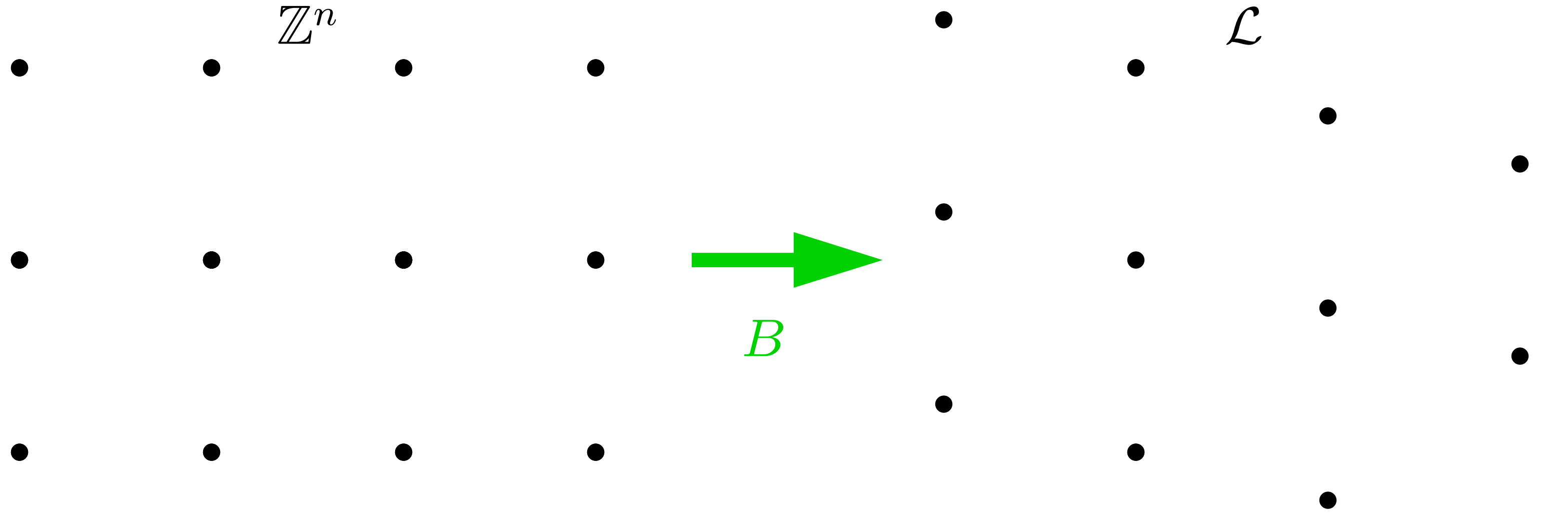
Minkowski's Convex Body Theorem



- ⋆ $2x, -2y \in C$ as factor of 2 larger and symmetric
- ⋆ The midpoint z of $2x$ and $-2y$ also in C as convex
- ⋆ $z = \frac{1}{2}(2x - 2y) = x - y \in \mathbb{Z}^n \quad \square$

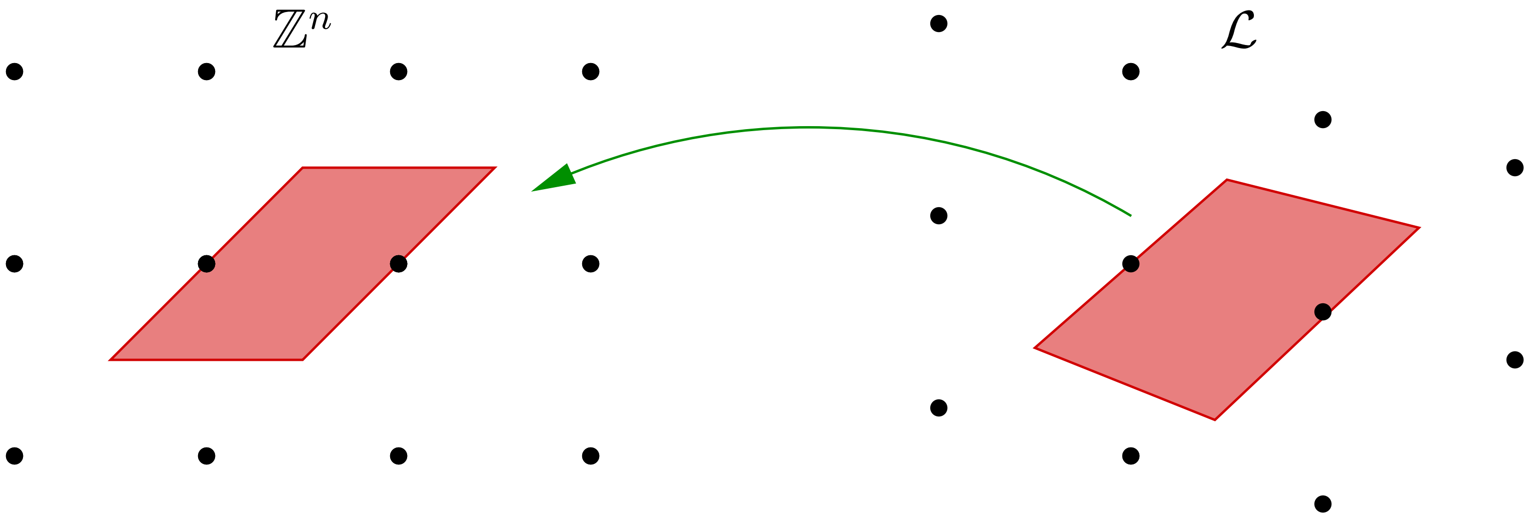
General Lattices

Any lattice $\mathcal{L}$ is a linear transformation B of $\mathbb{Z}^n$



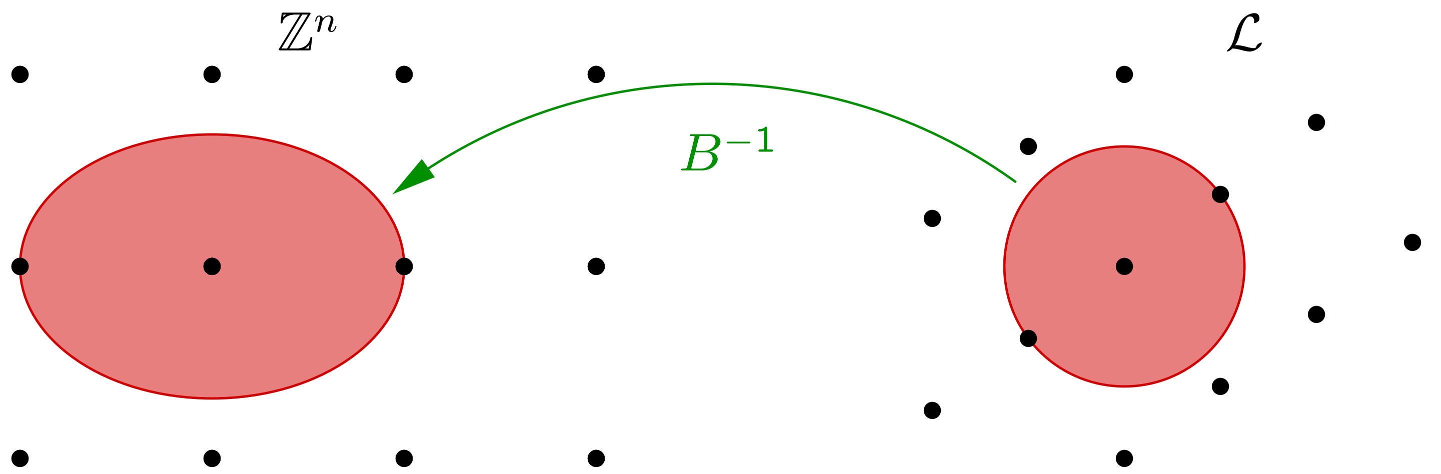
$\det T$ tells how volume scales between $\mathbb{Z}^n$ and $\mathcal{L}$

General Lattices



Any *(convex, symmetric)* body in $\mathcal{L}$ is related to a *(convex, symmetric)* body in $\mathbb{Z}^n$

Using Minkowski's First Theorem



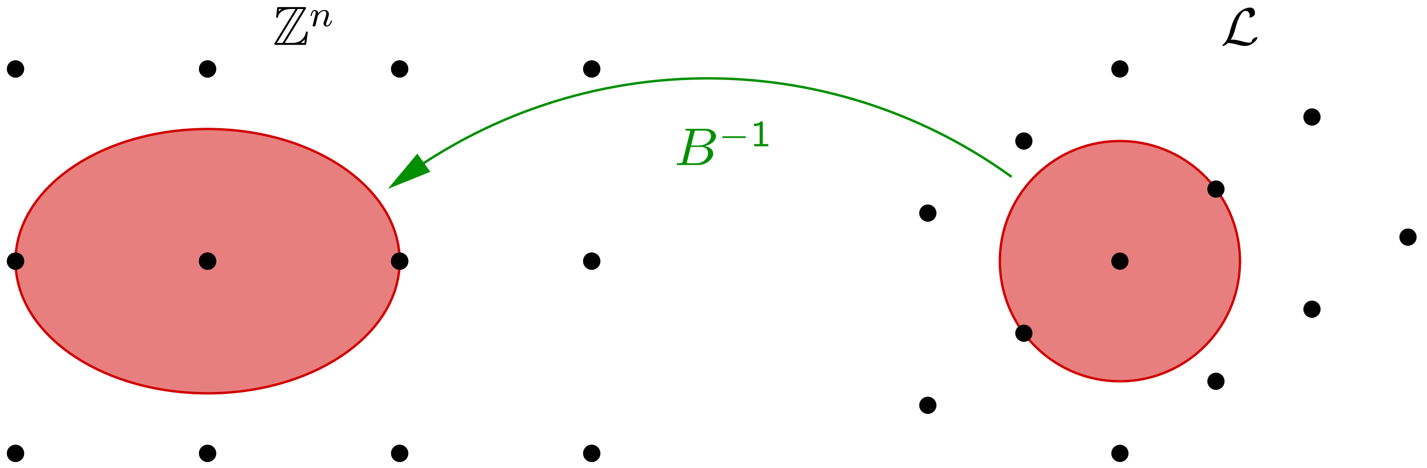
- ⋆ Let $\lambda(\mathcal{L})$ be the length of a shortest nonzero vector in $\mathcal{L}$

- ⋆ The sphere in $\mathcal{L}$ just containing a shortest vector has volume $\lambda(\mathcal{L})^n \cdot V_n$

- ⋆ Minkowski's Theorem says

$$\underbrace{\lambda(\mathcal{L})^n \cdot V_n}_{\text{volume of sphere}} \quad / \quad \underbrace{\det B}_{\text{change in volume for ellipse}} \quad \leq 2^n$$

Using Minkowski's First Theorem



- ⋆ Minkowski's Theorem says

$$\underbrace{\lambda(\mathcal{L})^n \cdot V_n}_{\text{volume of sphere}} \quad / \quad \underbrace{\det B}_{\text{change in volume for ellipse}} \quad \leq 2^n$$

- ⋆ Rearranging,

$$\lambda(\mathcal{L}) \leq 2\left(\det(B)/V_n\right)^{1/n} \leq \sqrt{n}\,\det(B)^{1/n}$$

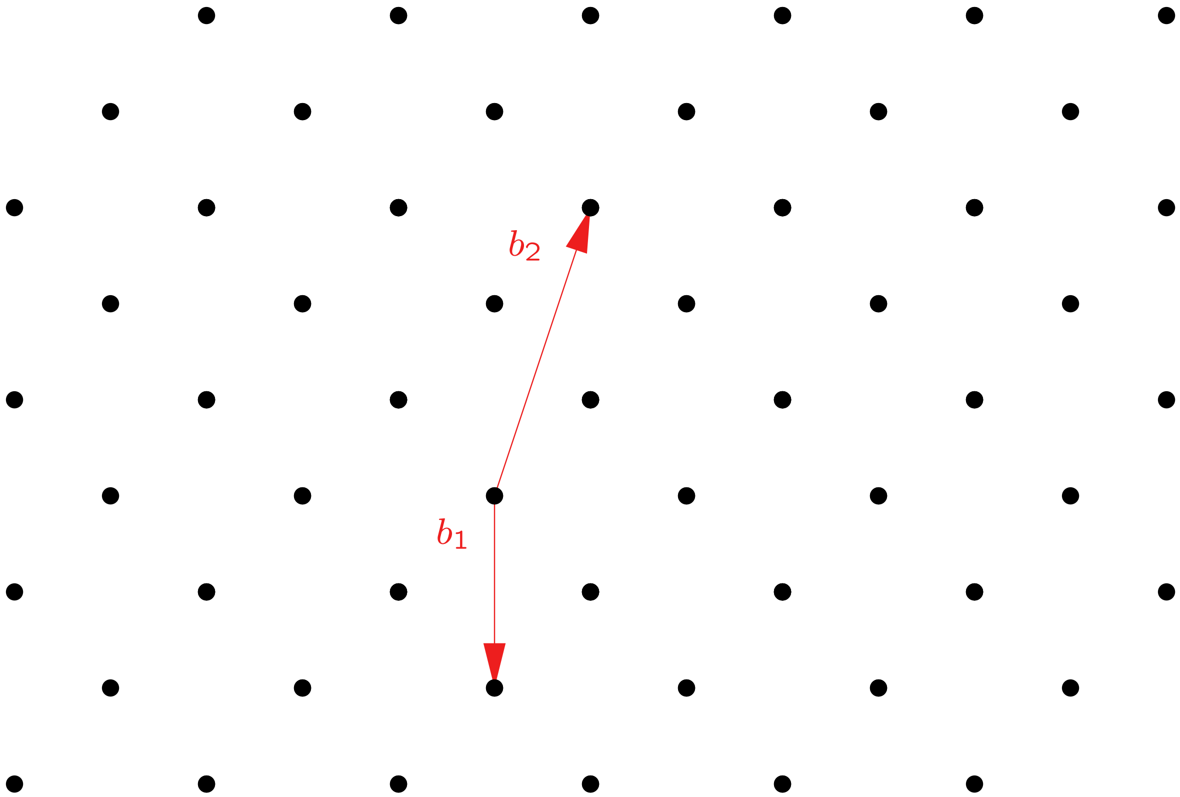
Outline

1. Elementary bounds $\surd$

2. Reduction algorithms

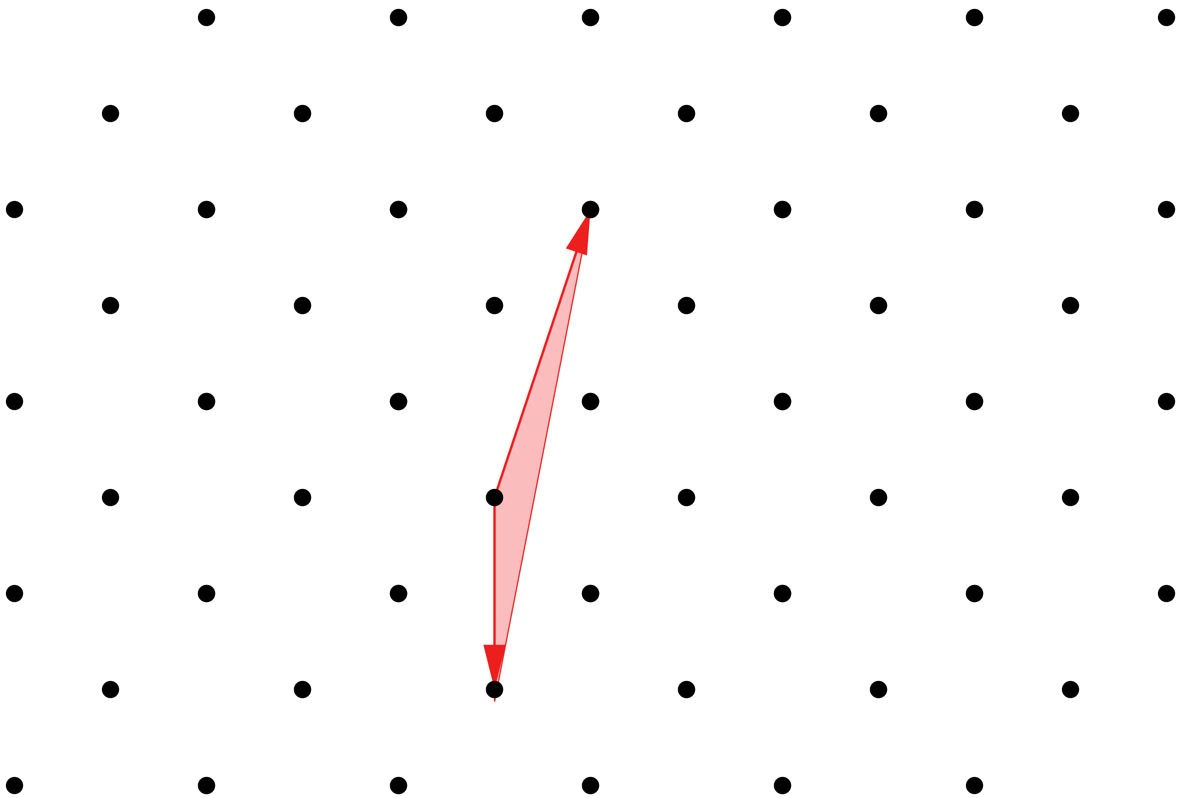
3. (My) current research

Lattice Basis



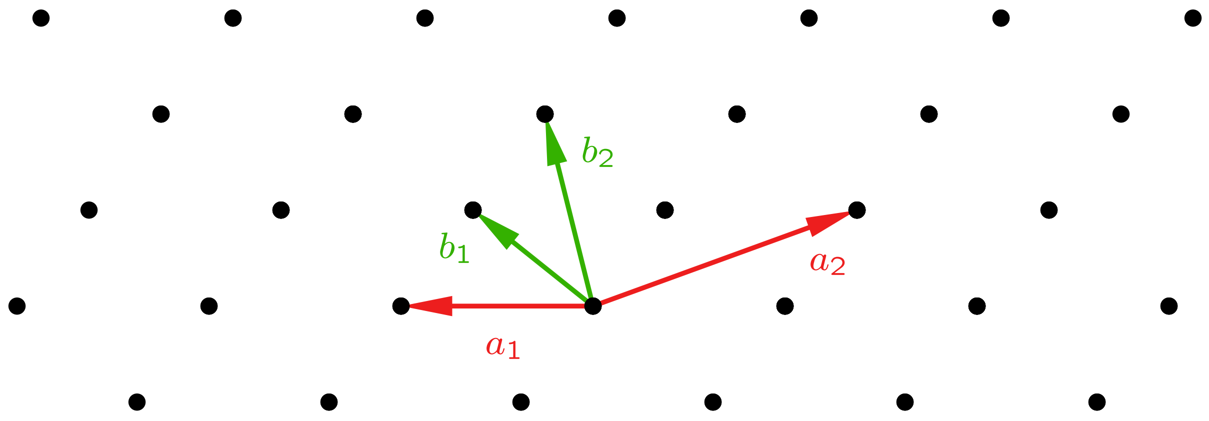
$$B = [b_1 b_2]$$
$$\mathbb{Z}^2 \mapsto_B \mathcal{L}$$
$$\mathcal{L} = \{x_1 b_1 + x_2 b_2 : x_1, x_2 \in \mathbb{Z}\}$$

Why B is a Basis



The triangle spanned by B contains no lattice points except the vertices

Basis

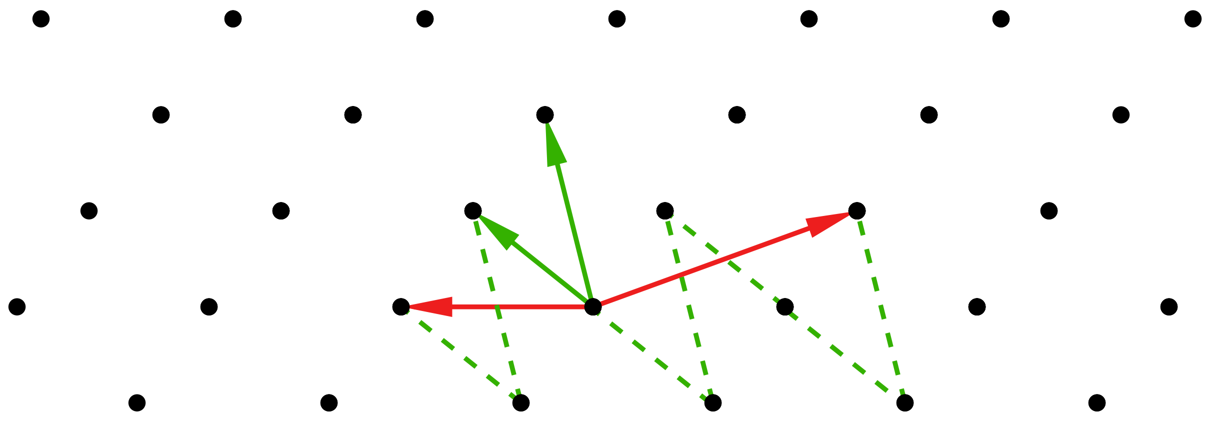


$$A = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ -4 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} -1 & 4 \\ -2\frac{1}{2} & 2 \end{pmatrix}$$

There are many bases for the same lattice

Why the Bases are the Same

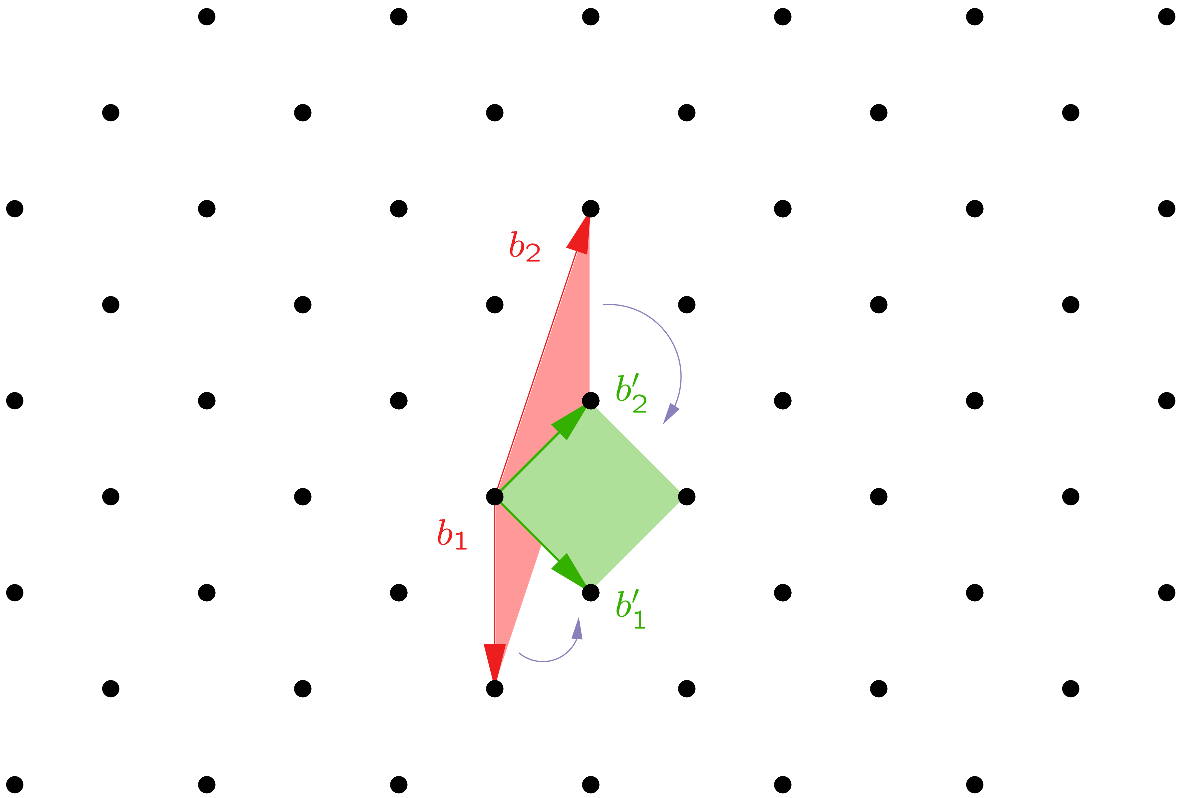


⋆ The red basis can be expressed in the green basis, and vice-versa

⋆ Integer unimodular transformation U with

$$A = UB$$

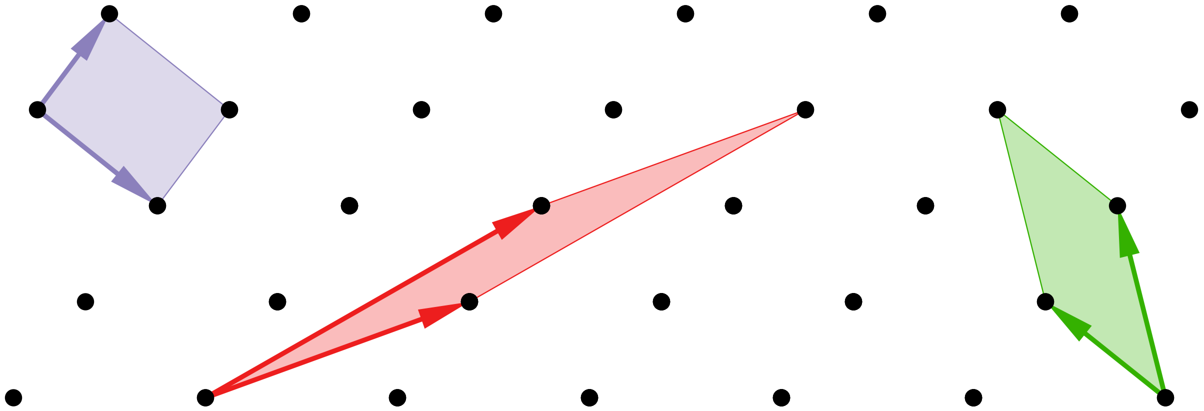
We Like Some Bases Better Than Others



Transform given basis to one with short vectors:

Basis Reduction

Geometry of Determinant



- ⋆ $d(\mathcal{L}) \triangleq \det B =$ volume of *fundamental region*

- ⋆ If B is not square, $d(\mathcal{L}) = \sqrt{\det BB^T}$

Results on Shortest Vectors

- $\star$ Recall that $\lambda(\mathcal{L})$ is the length of a shortest non-zero vector of $\mathcal{L}$.
- $\star$ Theory tells us:

$$\lambda(\mathcal{L}) \leq \sqrt{n} \cdot d(\mathcal{L})^{1/n}$$

[Minkowski 1896]

- $\star$ Polynomial-time LLL algorithm finds $v \in \mathcal{L}$ with:

$$\begin{aligned} |v| &\leq 2^{n/4} \cdot d(\mathcal{L})^{1/n} \\ |v| &\leq 2^{n/2} \cdot \lambda(\mathcal{L}) \end{aligned}$$

[Lenstra, Lenstra, Lovasz '82]

- $\star$ Block Korkine-Zolotareff reduction replaces 2 with $(1 + \epsilon)$

Outline

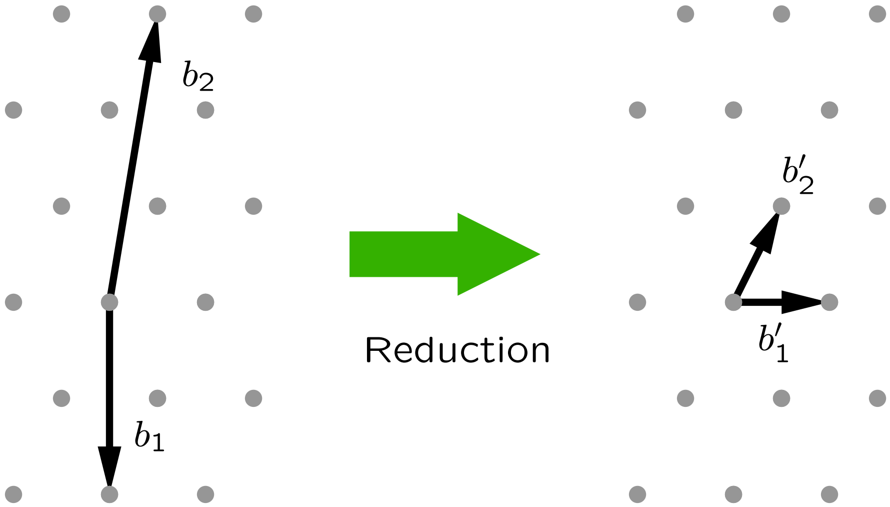
1. Elementary bounds $\surd$

2. Reduction algorithms

   - $\star$ 2-D Gaussian reduction
   - $\star$ LLL reduction
   - $\star$ Block Korkine-Zolotareff reduction

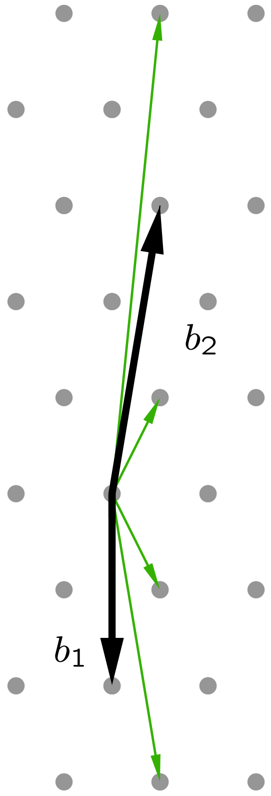
3. (My) current research

The Two-Dimensional Case



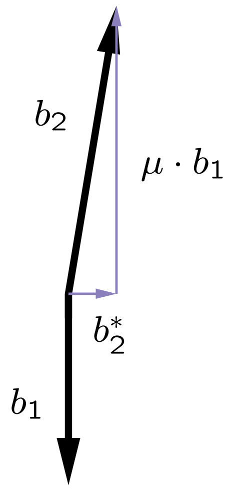
b'_1 is a shortest vector

2D Reduction



If $|b_1| < |b_2|$, shrink b_2 by adding multiples of b_1

Gram-Schmidt Orthogonalization



$$b_2 = b_2^* + \mu b_1$$

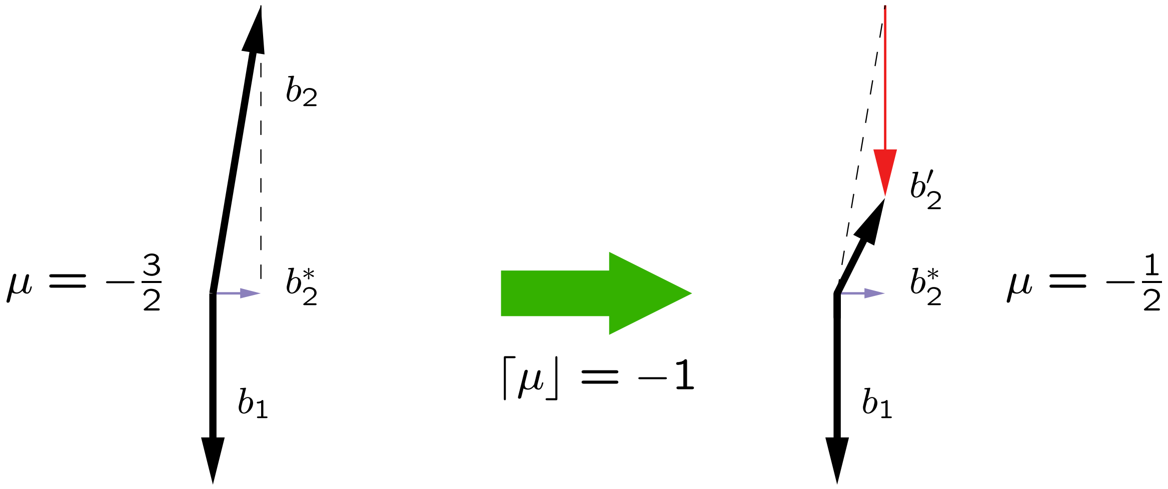
$$b_2^* \perp b_1$$

$$\mu = \frac{\langle b_2, b_1 \rangle}{|b_1|^2}$$

b_2^*, μ *rational* quantities

How Much to Add?

Size Reduction

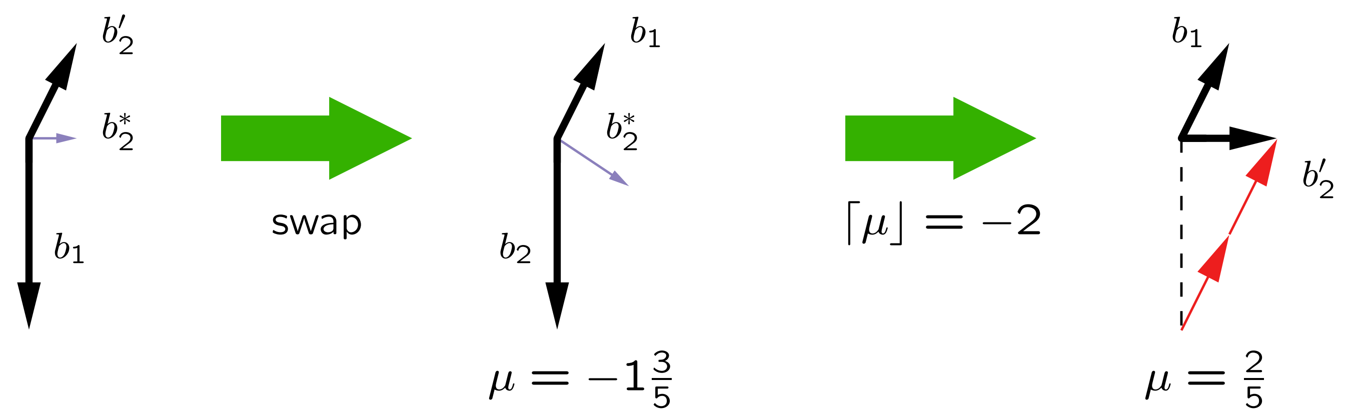


$$b_2' = b_2^* + (\mu - \lceil \mu \rfloor) b_1 = b_2^* + \mu' b_1$$

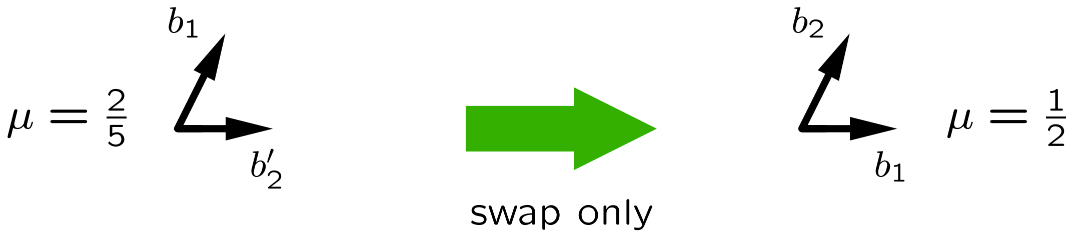
$$|\mu'| \leq \frac{1}{2}$$

2D Reduction

$|b'_2| < |b_1|$, so swap and continue. . .



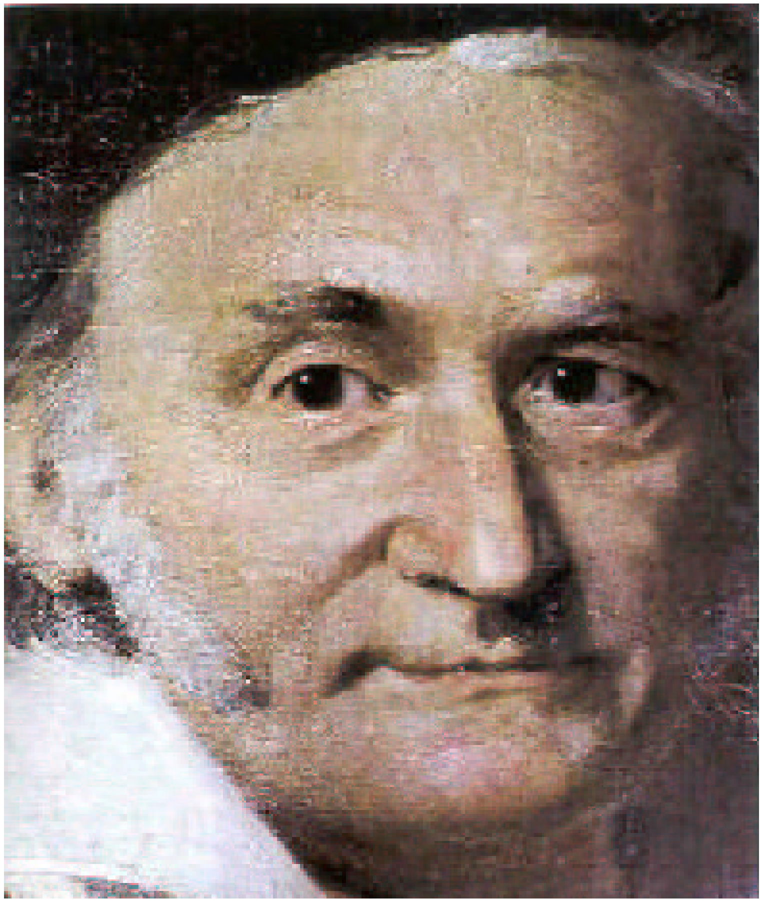
Gaussian Reduction Conditions



. . . until no more improvement possible:

$$|b_1| \leq |b_2|$$
$$|\mu| \leq \frac{1}{2}$$

2D Reduction



Carl Friedrich Gauss (1777-1855)

The Gaussian Reduction Algorithm

```
GaussianReduce($b_1, b_2$)
   do
      if $|b_1| > |b_2|$ then
         swap $b_1$, $b_2$
      $\mu \leftarrow \frac{\langle b_2, b_1 \rangle}{|b_1|^2}$
      $b_2 \leftarrow b_2 - \lceil \mu \rfloor b_1$
   while $|b_1| > |b_2|$
   return $(b_1, b_2)$
```

```
GCD($x, y$)
   do
      if $x > y$ then
         swap $x, y$
      $(x, y) \leftarrow (y \bmod x, x)$
   while $x > 0$
   return $y$
```

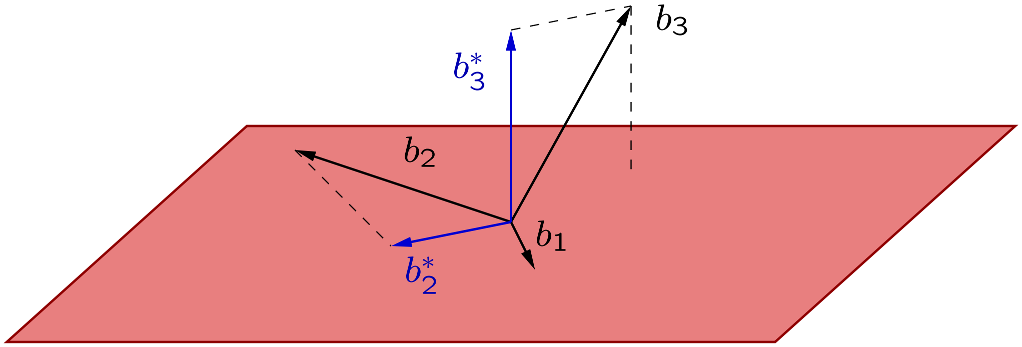
Gram-Schmidt Orthogonalization in Arbitrary Dimension

$$b_1^* = b_1$$

b_2^* is component of b_2 perpendicular to b_1.

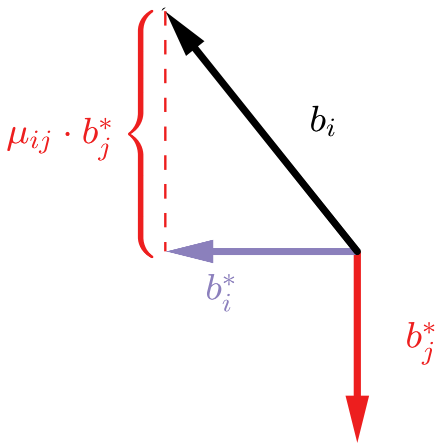
b_3^* is component of b_3 perpendicular to span(b_1, b_2).

$\vdots$

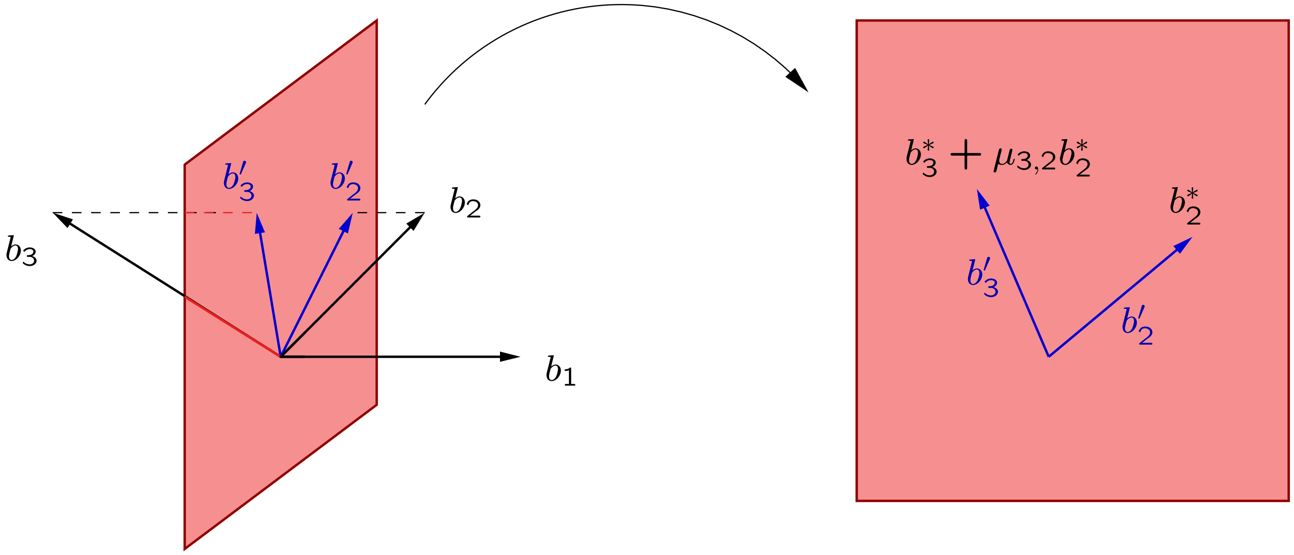


Gram-Schmidt Orthogonalization

$$\begin{aligned} b_1^* &= b_1 \\ \mu_{ij} &= \frac{\langle b_i, b_j^* \rangle}{|b_j^*|^2} \\ b_i^* &= b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^* \end{aligned}$$



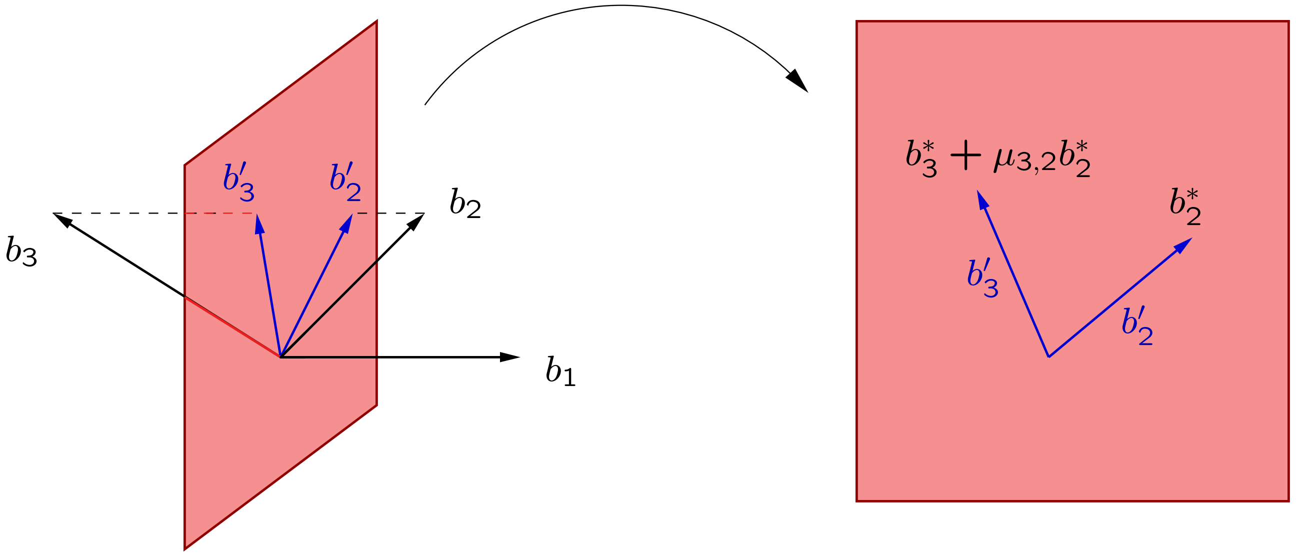
Projecting Lattices



$$\{b_1, b_2, b_3\} \rightarrow \{b'_2, b'_3\}$$

Project b_2 and b_3 to subspace $\perp b_1$

When to Swap



Apply Gaussian Reduction to b'_2, b'_3:

Swap if $|b'_2|^2 > \frac{4}{3}|b'_3|^2$

The Algorithm

```
$\texttt{General} - \texttt{Reduction}(B = b_1, \ldots, b_n)$
    while $|b_i^*|^2 > \frac{4}{3}|b_{i+1}^* + \mu_{i+1,i}b_i^*|^2$ for some $i$,
            {some pair not Gaussian reduced}
        $\texttt{GaussianReduce}(b_i^*, (b_{i+1}^* + \mu_{i+1,i}b_i^*))$
        update $\mu_{hk}$ and $b_k^*$ for all $h, k$.
        $B \leftarrow \texttt{SizeReduce}(B)$
    return $B$
```

This is the famous *LLL Basis Reduction* of Lenstra, Lenstra and Lovász

The Gaussian Reduction Algorithm

```
GaussianReduce($b_1, b_2$)
  do
      if $|b_1| > |b_2|$ then
          swap $b_1$, $b_2$
      $\mu \leftarrow \frac{\langle b_2, b_1 \rangle}{|b_1|^2}$
      $b_2 \leftarrow b_2 - \lceil \mu \rfloor \, b_1$          { size-reduction }
  while $|b_1|^2 > \frac{4}{3}|b_2|^2$
  return $(b_1, b_2)$
```

Size-Reduction with Gram-Schmidt

```
SizeReduce($B = b_1, \ldots, b_n$)
  for $j = 2, \ldots, n$
      for $i = j - 1, \ldots, 1$
          $b_j \leftarrow b_j - \lceil \mu_{ji} \rfloor b_i$
          $\mu_{jk} \leftarrow \mu_{jk} - \lceil \mu_{ji} \rfloor \mu_{ik}$ for $k = 1, \ldots, i$
  return $B$
```

- ⋆ Now $|\mu_{ij}| \leq \frac{1}{2}$ for all $j < i$
- ⋆ b_i^* are unchanged

Algorithms

- ⋆ $O\big(n^4 \log S\big)$ operations on $O\big(n \log S\big)$-bit numbers, on $n \times n$ input matrix with S-bit coefficients.

- ⋆ Improved to $O\big(n^3 \log S\big)$ operations on $O\big(n + \log S\big)$-bit integers and floating point numbers [Schnorr, Koy '01].

- ⋆ By reducing blocks rather than pairs of vectors, get $(1+\epsilon)^{n/2}$ approximation [Schnorr '89] ($\sim$ 1.5 is practical).

- ⋆ The current standard is block-reduction, sped up with *pruning heuristic* and floating-point Gram-Schmidt calculations, iterating several stages over the basis to be reduced. Lattices of dimension 800 and similar bit-length are practical.

An Asymptotically Bad Basis for LLL

$$B = \begin{bmatrix} \alpha & & & & \\ \rho & \alpha\rho & & & \\ \rho^2 & \rho^2 & \alpha\rho^2 & & \\ & \vdots & & \ddots & \\ \rho^{n-1} & \cdots & \cdots & \cdots & \alpha\rho^{n-1} \end{bmatrix}$$

$$|b_i^*| = \alpha\rho^{i-1}$$
$$\mu_{ji} = \tfrac{1}{\alpha}\rho^{j-i} \text{ for } j > i$$
$$|b_i| = \rho^{i-1}(\alpha + i - 1)$$
$$\frac{|b_{i+1}(i)|^2}{|b_i(i)|^2} = \frac{\alpha^2\rho^{2i} + \frac{1}{4}\alpha^2\rho^{2i-2}}{\alpha^2\rho^{2i-2}} = \rho^2 = \frac{1}{4}$$
$$(\det B)^{1/n} = \alpha\rho^{(n-1)/2}$$

If we take $\alpha = \sqrt{3}$ and $\rho = \alpha/2$, then $|b_{i+1}(i)|^2/|b_i(i)|^2 = 1$ and $\mu_{ji} = 1/2$ for $j > i$, hence B is LLL reduced. But the last row has length

$$\sqrt{n\rho^{n-1}\alpha^2} = \sqrt{n}\rho^{(n-1)/2}\alpha \ldots \qquad \ldots \text{while } |b_1| = \alpha.$$

Permute this order, and it's no longer reduced. No bad basis known if rows are permuted before performing the reduction.

Towards Schnorr's Algorithm

- ⋆ LLL reduction finds shortest vectors in projected 2D blocks, and iterates

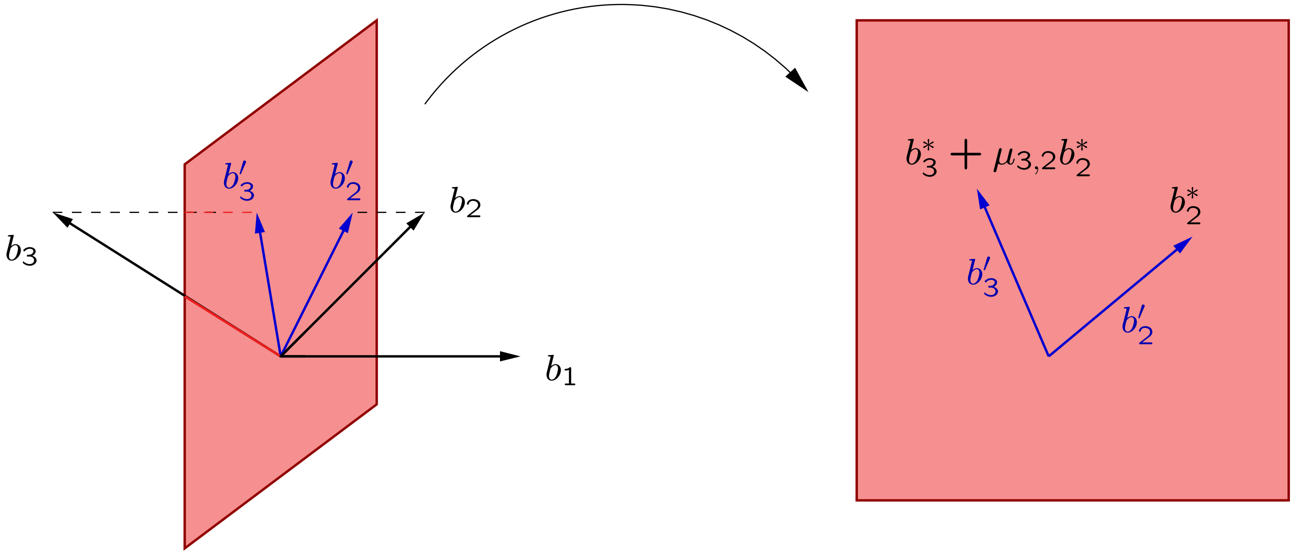
$$b_1 \quad b_2 \quad b_3 \quad \underbrace{b_4 \quad b_5} \quad b_6 \quad b_7 \quad b_8 \quad b_9 \quad b_{10}$$

- ⋆ Could we improve by finding optimum of larger blocks?

$$b_1 \quad b_2 \quad b_3 \quad \underbrace{b_4 \quad b_5 \quad b_6 \quad b_7} \quad b_8 \quad b_9 \quad b_{10}$$

- ⋆ Issues:
  - ◇ Is there an "efficient" exhaustive search to find the shortest vector?
  - ◇ What's the right reduction to use so we can iterate?
  - ◇ Can we prove it works?

Korkine-Zolotareff Reduction



- ⋆ For basis B, let $B' = \{b_2', \ldots, b_n'\}$
- ⋆ B is *Korkine-Zolotareff reduced* if

$$|b_1| = \lambda(B), \text{ and}$$
$$B' \text{ is Korkine-Zolotareff reduced}$$

Why so complicated?

- ⋆ A natural notion of reduction might be:

$$\begin{gathered} |b_1| = \lambda\left(B\right), \\ |b_2| = \lambda\left(B \setminus \{b_1\}\right), \text{ etc.} \end{gathered}$$

- ⋆ But such a set may not be a basis if $n \geq 5$!

$$\begin{pmatrix} 2 & & & & \\ & 2 & & & \\ & & 2 & & \\ & & & 2 & \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

- ⋆ KZ reduction recurses on *projected* bases rather than *linearly independent* bases
  - ⋄ Easier to work with

Block KZ Reduction: The Algorithm

- ⋆ Divide basis into overlapping blocks of length k

$$\overbrace{b_1 \quad \mathrlap{\underbrace{}_{\text{Block 2}}} b_2 \quad b_3 \quad b_4}^{\text{Block 1}} \quad b_5 \quad b_6 \quad b_7 \quad b_8$$

- ⋆ While there exists a block that isn't KZ reduced, reduce it

- ⋆ As blocks overlap, reduction of one block may provide opportunity to reduce an overlapping block

- ⋆ Can prove polynomial running time (sort of. . .)

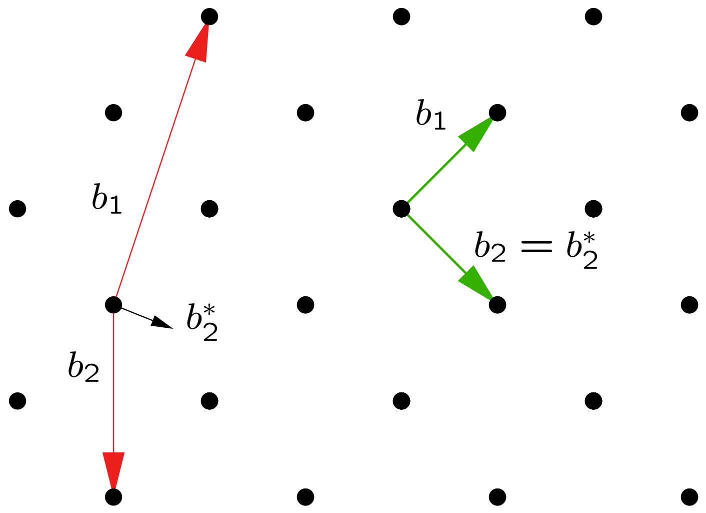
Block KZ Reduction: The Analysis

- ⋆ Define $\alpha_n = \max_{\text{KZ reduced}} |b_1|^2 / |b_n^*|^2$
- ⋆ Universal constant for KZ reduction
- ⋆ A k-block KZ reduced basis satisfies

$$|b_1|^2 \leq \alpha_k^{n/k} \lambda(\mathcal{L})^2$$

- ⋆ Minkowski's Theorem implies $\alpha_k \leq k^{1+\ln k}$
- ⋆ By setting k appropriately, $k^{(1+\ln k)/k} < (1+\epsilon)$ gives the bound we want

Digression on α_n

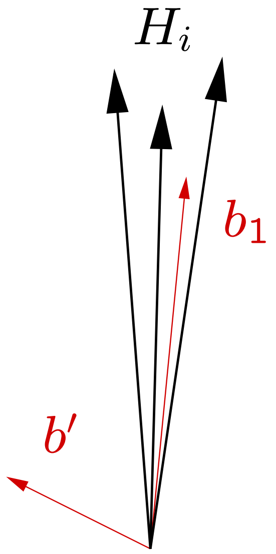


- ⋆ $|b_1|^2/|b_n^*|^2$ gives good metric for quality of reduction of basis
- ⋆ An LLL-reduced basis has $|b_1|^2/|b_n^*|^2 \sim 2^n$
- ⋆ A KZ-reduced basis has $|b_1|^2/|b_n^*|^2 = \alpha_n \sim n^{\ln n}$
- ⋆ Quality of basis reduction much deeper than shortest vector: exponential versus quasi-polynomial

Future Directions

- ⋆ Select random subspaces of the lattice and reduce there
  - ⋄ Classical results (Dvortsky's Theorem) suggest lattice will behave nicely on random subspaces
- ⋆ Problem: the subspace is likely to have a very short vector
- ⋆ Solution: reduce across many subspaces
  - ⋄ Experimental results promising

Random Subspace Reduction



- ⋆ Select lattice subspaces $H_1 \cdots H_t$ depending on basis
- ⋆ Project b_1 to each subspace, rationally
- ⋆ Subtract rounded sum of projected points from b_1 to get b'
- ⋆ Intuition:
  - ◇ If basis not well-reduced, the H_i will share common alignment
  - ◇ After subtraction, b' will be more orthogonal to this alignment
- ⋆ Seems to work in practice