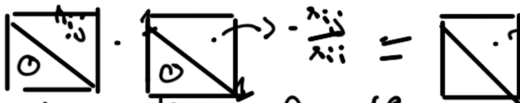
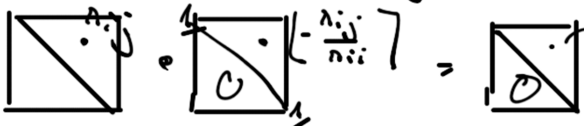
Size-reducing:  $\to -\frac{r_{ij}}{r_{ii}}$ ... $\to r_{ij}^{(\text{new})} = 0$.

This is not an admissible transformation, as we may not have $-\frac{r_{ij}}{r_{ii}} \in \mathbb{Z}$.

Instead:  $\left\lfloor -\frac{r_{ij}}{r_{ii}} \right\rceil$ ... $\to r_{ij}^{(\text{new})} = r_{ij}^{\text{old}} + \left\lfloor -\frac{r_{ij}}{r_{ii}} \right\rceil r_{ii}$

$\in [-r_{ii}/2, +r_{ii}/2]$.

unimodular

$$B = QR \longrightarrow B\cdot T = Q\underbrace{R\cdot T}_{\text{upTr.}}$$

orth. upTr. $\Rightarrow$ this is the new R.

How to make the whole i-th column small? Go from bottom to top, as T_{ij} modifies $r_{i'j}$ for $i' \leq i$ only.

Compute $B = Q.R$
For $i = j-1$ down to 1 do
  $b_j \leftarrow b_j + \lfloor -r_{ij}/r_{ii} \rceil b_i$
  $r_{i'j} \leftarrow r_{i'j} + \lfloor -r_{ij}/r_{ii} \rceil r_{i'i}$ for $i' = 1 \ldots i$.

$\Big\}$ $O(n^2)$ ~~operations~~

This is called size-reduction, SIC, Babai's nearest plane algorithm, etc.

$\Rightarrow$ At this stage we see that what matters is the r_{ii}'s

But $\prod r_{ii} = \det B = \det L$ is cst.

③ The LLL algorithm ['82]

We want to make the r_{ii}'s not decrease too fast, so that the first is rather small.

Observation:
$$\begin{bmatrix} r_{11} & r_{12} \\ & r_{22} \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = Q \cdot \begin{bmatrix} (r_{12}^2 + r_{22}^2)^{1/2} & \text{ble} \\ 0 & \frac{r_{11} r_{22}}{(r_{12}^2 + r_{22}^2)^{1/2}} \end{bmatrix}.$$

If $\binom{r_{12}}{r_{22}}$ is shorter than $\binom{r_{11}}{0}$, we've made some progress.

If $\left\| \binom{r_{12}}{r_{22}} \right\| \leq r_{11} \cdot \underset{<1}{\delta}$, we've made significant progress.

If it is size-reduced, it has a better chance to work...

LLL: size-reduce
While $\exists i$ s.t. $\left\| \begin{pmatrix} r_{i\,i+1} \\ r_{i+1\,i+1} \end{pmatrix} \right\| \leq \delta \cdot r_{ii}$, swap $b_i \leftrightarrow b_{i+1}$
update R.
size-reduce.

Return B.

Cost driven by the # of iterations. How can we bound it?

Look at $\Pi = \prod_{i=1}^{n} \Big(\underbrace{\prod_{j=1}^{i} r_{jj}}_{\text{starting determinant}} \Big)^2.$

For a swap at i: $\prod_{j=1}^{i'} r_{jj}$ unchanged except for $i'=i$.
then $r_{11}, \ldots, r_{i-1\,i-1}$ unchanged and r_{ii} decreases by the factor δ

$\Rightarrow \Pi^{new} \leq \delta^2 \cdot \Pi^{old}$.

$\Pi \in \mathbb{Z}^+$: $\forall i$: $\left(\prod_{j=1}^{i} r_{jj}\right)^2 = \det\left((b_1, \ldots, b_i)^T (b_1 \ldots b_i)\right) \in \mathbb{Z}$.

Initially, $\Pi \leq (\max \|b_i^{init}\|)^{n^2}$.

Comment on sandpile model

$\Rightarrow$ # iterations $\leq n^2 \log \max \|b_i^{init}\| / \log 1/\delta$.

Quality: If a swap is not possible at index i, we have

$r_{i\,i+1}^2 + r_{i+1\,i+1}^2 \geq \delta^2 r_{ii}^2$

$\Rightarrow r_{i+1\,i+1}^2 \geq (\delta^2 - 1/4) r_{ii}^2$ by size-reduction.

$\Rightarrow r_{i+1\,i+1} \geq (\delta^2 - 1/4)^{1/2} r_{ii}$ $\forall i$. $\boxed{\text{PICTURE}}$.

(the r_{ii}'s don't decrease too fast).

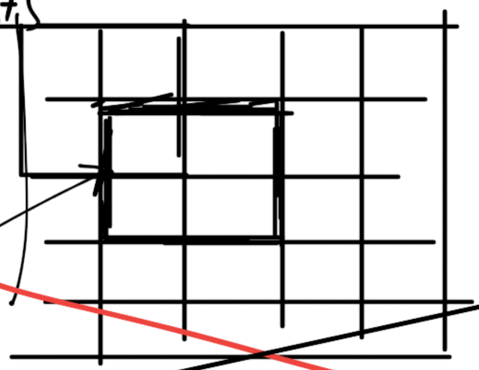
Thm.: B LLL reduced $\Rightarrow$ $\|b_1\| \leq \alpha^{n-1/2} (\det)^{1/n}$. $\alpha = \left(\frac{1}{\delta^2 - 1/4}\right)^{1/2}$.
$\|b_1\| \leq \alpha^{n-1} \cdot \lambda_1$

Pf: $\lambda_1 \geq \min r_{ii}$ $\Rightarrow$ 2nd statement.
For the 1st statement: $\det = \prod r_{ii} \leq \prod(\alpha^{i-1} r_{11}) = r_{11}^n \alpha^{-(\ldots)/2}$ $\square$.

④ Schnorr's algorithm [1987]

Bck size 2 → Bck size k.

Assume $k|n$ and $k \ll n$



$\prod_{j\in block_i} \lambda_{jj}$ vs $\prod_{j\in block_{i+1}} \lambda_{jj}$

to be defined.

If $[block_i, block_{i+1}]$ is HKZ-reduced, then:

$$\prod_{j\in block_i} \lambda_{jj} \Big/ \prod_{j\in block_{i+1}} \lambda_{jj} \le k^{c\cdot k} \text{ for some } c.$$

Proof: $\lambda_{11} \le \sqrt{n}\,\left(\prod_i \lambda_{ii}\right)^{1/n} \Rightarrow \lambda_{11}^{n-1/n} \le \sqrt{n}\cdot\left(\prod_{i>1}\lambda_{ii}\right)^{1/n}$

$\Rightarrow \lambda_{11} \le \sqrt{n}^{\,n/n-1}\left(\prod \lambda_{ii}\right)^{1/n-1}$.

$$\prod_{j\le n/2} \lambda_{jj} \le \sqrt{n}^{\frac{n}{n-1}} \left(\prod_{j=2}^{n/2}\lambda_{jj}\right)^{\frac{n}{n-1}} \cdot \left(\prod_{j>n/2}\lambda_{jj}\right)^{\frac{1}{n-1}}$$

$$\lambda_{22} \le \sqrt{n-1}^{\frac{n-1}{n-2}} \left(\prod_{i\ge 3}\lambda_{ii}\right)^{1/n-2}$$

$$\frac{1}{n-1}+\frac{n}{n-1}\cdot\frac{1}{n-2} = \frac{1}{n-1}\left[\frac{2(n-1)}{n-2}\right] = \frac{2}{n-2}$$

$$\prod_{j\le n/2}\lambda_{jj} \le (\sqrt{n})^{\frac{n}{n-1}}\cdot(\sqrt{n-1})^{\frac{n}{n-2}}\cdot\left(\prod_{j=3}^{n/2}\lambda_{jj}\right)^{\frac{n}{n-2}}\cdot\left(\prod_{j>n/2}\lambda_{jj}\right)^{\frac{2}{n-2}}.$$

$$\le \dots$$

$$\le \underbrace{(\sqrt{n})^{n\sum_{k=n/2}^{n}\frac{1}{k}}}_{n^{O(n)}} \cdot \left(\prod_{j>n/2}\lambda_{jj}\cdot\right).$$

Adaptation of LLL: Find d: s.t. $\prod_{block_{i+1}} \le \delta\, k^{c.k} \cdot \prod_{block_i}$.

At the end, HKZ-reduce each block.

New potential: $\prod_i \left(\prod_{j\le i} block_j\right)$.

Quality/Time: time $2^{\Theta(k)}$. for $\frac{\|b_1\|}{\lambda_1} \le k^{\Theta(n/k)}$.

$$\frac{\|b_1\|}{\det^{1/n}} \le k^{O(n/k)}.$$

5) The BKZ algorithm. (SchEu91)

For $i = 1$ to n do $HKZ_{\min(\beta, n-i)}$ on proj. basis starting at index i

Repeat many times.

Better output quality than Schnorr (better cst in the exponent)

Best known algorithm in practice.

H.S'07: we may stop after a few tours.

How to analyze it?

Sand-pile model.

↳ open problem even for $\beta = 2$.

BKZ2.0 [Chen-Nguyen]

↳ early abort

\+ extreme pruning

\+ pre-processing.