# H-LLL: Using Householder Inside LLL

Ivan Morel, Damien Stehlé, Gilles Villard

LIP-CNRS-ENSL-INRIA-UCBL-University of Sydney

ISSAC'09
July 29, 2009.

# Context

- Euclidean lattices.
- LLL-reduction.
- Using floating-point arithmetic to speed Gram-Schmidt computations up.
- Ensuring termination and correctness of the algorithm.

# Why ?

Lattice reduction is a powerful tool:

- Polynomial factorization.
- Combinatorial optimization.

# Why ?

Lattice reduction is a powerful tool:

- Polynomial factorization.
- Combinatorial optimization.

LLL remains expensive:

- Rational arithmetic is expensive within Gram-Schmidt.
- Using floating-point arithmetic instead.

# QR inside LLL

Advantages (wrt Nguyen-Stehlé ’05):

- Smaller condition number.
- Fewer operations.
- No need for the Gram matrix.

# QR inside LLL

Advantages (wrt Nguyen-Stehlé ’05):

- Smaller condition number.
- Fewer operations.
- No need for the Gram matrix.

Drawbacks:

- Subject to more cancellations.
- Satisfying the usual definition of reduction is impossible.
- Best results so far were heuristic (Koy-Schnorr ’01, Shoup’s NTL).

# Our results

- A more intuitive algorithm for floating-point LLL.
- Requires a smaller precision (40% less than $L^2$).
- Better overall complexity.

# Our results

- A more intuitive algorithm for floating-point LLL.
- Requires a smaller precision (40% less than $L^2$).
- Better overall complexity.
- Thanks to a new size-reduction algorithm.

# Comparison with previous rigorous LLL's

The overall complexity:

$$O\left[\left(d + \log \prod \frac{d_i^{initial}}{d_i^{final}} + \frac{1}{d} \log \prod \frac{\|\mathbf{b}_i^{initial}\|}{\|\mathbf{b}_i^{final}\|}\right) nd^2(d + \log \|B\|)\right],$$

where $d_i$ is the determinant of the $i \times i$ matrix $(\langle \mathbf{b}_j ; \mathbf{b}_k \rangle)_{j,k \leq i}$.

# Comparison with previous rigorous LLL's

The overall complexity:

$$O\left[\left(d + \log \prod \frac{d_i^{initial}}{d_i^{final}} + \frac{1}{d} \log \prod \frac{\|\mathbf{b}_i^{initial}\|}{\|\mathbf{b}_i^{final}\|}\right) nd^2(d + \log \|B\|)\right],$$

where $d_i$ is the determinant of the $i \times i$ matrix $(\langle \mathbf{b}_j ; \mathbf{b}_k \rangle)_{j,k \leq i}$.

| | Complexity | Precision |
|---|---|---|
| LLL | $O(d^5 \log^3 \|B\|)$ | $d \log \|B\|$ |
| Schnorr 88 | $O(d^4 \log^3 \|B\|)$ | $O(\log \|B\|)$ |
| $L^2$ | $O(d^5 \log^2 \|B\|)$ | $d \log_2 3$ |
| H-LLL | $O(d^5 \log^2 \|B\|)$ | $d$ |

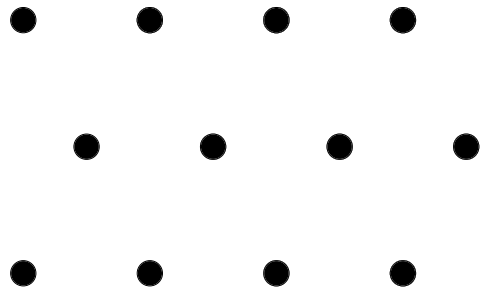
Hypotheses: $n = O(d), d = O(\log \|B\|)$

# Outline

# Outline

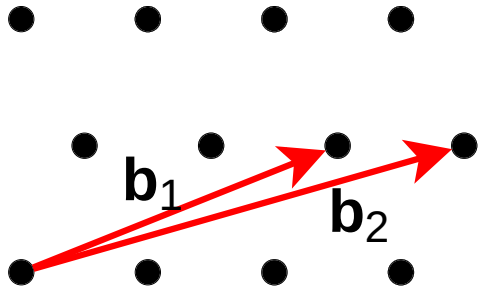
# Euclidean lattices

- Discrete subgroup of $\mathbb{Z}^n$.



# Euclidean lattices

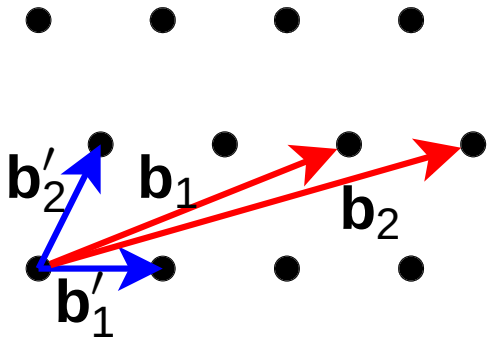
- Discrete subgroup of $\mathbb{Z}^n$.




- Described by a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, represented by a matrix $B$ (columns).

# Euclidean lattices

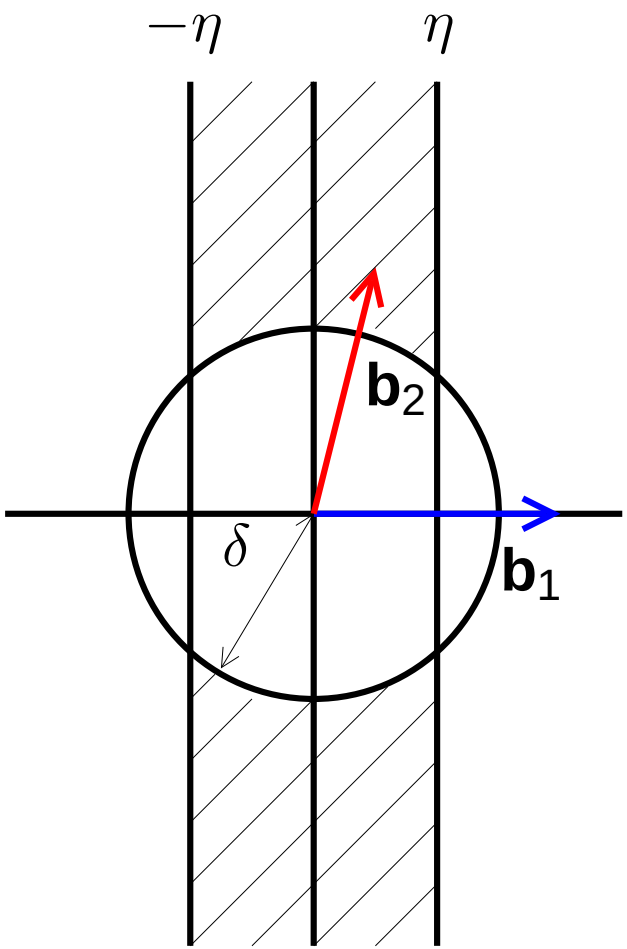
- Discrete subgroup of $\mathbb{Z}^n$.




- Described by a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$, represented by a matrix $B$ (columns).
- From a given basis to another via unimodular transforms.

$$\begin{pmatrix} 5 & 7 \\ 2 & 2 \end{pmatrix} \cdot \underbrace{\begin{pmatrix} -1 & 3 \\ 1 & -2 \end{pmatrix}}_{U} = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$$

# Quality of a 2-dimensional basis




$$B = \begin{pmatrix} 1 & a \\ 0 & b \end{pmatrix}$$

- Size-reduction: $|a| < \eta$.
- Lovász: $\delta \leq a^2 + b^2$.

# LLL-reduction

The 2-dimensional reduction is extended through $QR$ decomposition:

$$B = \begin{pmatrix} & & \\ Q & \text{orthog.} & \\ & & \end{pmatrix} \cdot \begin{pmatrix} * & * & R \\ & * & * \\ (0) & & * \end{pmatrix}$$

- Size-reduction: $|R_{i,j}| \leq \eta R_{i,i}$ $\quad \left(\eta \geq \frac{1}{2}\right)$.
- Lovász: $\delta R_{i,i}^2 \leq R_{i,i+1}^2 + R_{i+1,i+1}^2 \quad \left(1 - \eta^2 < \delta < 1\right)$.

# The LLL algorithm

- Start with $\kappa := 2$.
- Size-reduce vector $\kappa$ against the previous vectors.
- If the Lovász condition is satisfied: increment $\kappa$.
- Otherwise switch vectors $\kappa$ and $\kappa - 1$ and decrement $\kappa$.

# Outline

# Plugging fp arithmetic inside LLL

Computing Gram-Schmidt coefficients ($R$) and updating them takes most of LLL's execution time.

# Plugging fp arithmetic inside LLL

Computing Gram-Schmidt coefficients ($R$) and updating them takes most of LLL's execution time.
$\rightarrow$ Plugging fp numbers to compute $R$.

# Plugging fp arithmetic inside LLL

Computing Gram-Schmidt coefficients ($R$) and updating them takes most of LLL's execution time.
$\rightarrow$ Plugging fp numbers to compute $R$.

Challenge: ensuring the correctness of the computations made using approximate values of $R$.

# Approximating $R$ with the Householder algorithm

**Theorem [D. Stehlé, G. Villard & X.-W. Chang]**

Let $B = (b_1, \ldots, b_d)$ be a LLL-reduced basis. Let $p$ be the floating-point precision. Then noting $R$ the exact $R$-factor of $B$ and $\widetilde{R}$ the $R$-factor computed using Householder:

$$|R_{i,j} - \widetilde{R}_{i,j}| = 2^{O(d)} \cdot 2^{-p} \cdot R_{j,j}.$$

- Requires less precision than Cholesky's
- The type of error differs from Cholesky's.

# Stability and size-reduction are inconsistent

$$\begin{pmatrix} \ddots & & & \\ & R_{i,i} & \overset{\eta}{\leftarrow} & R_{i,j} \\ & & \ddots & \\ & (0) & & R_{j,j} \\ & & & \ddots \end{pmatrix}$$

Size-reduction linked to the row coefficient.

# Stability and size-reduction are inconsistent

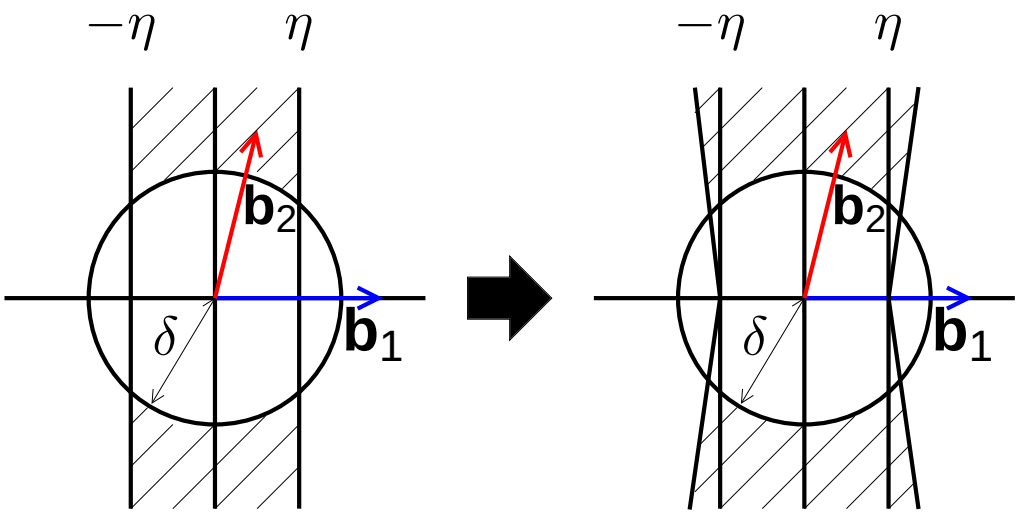
$$\begin{pmatrix} \ddots & & & \\ & R_{i,i} & & R_{i,j} \\ & & \ddots & \downarrow \\ & (0) & & R_{j,j} \\ & & & & \ddots \end{pmatrix}$$

Size-reduction not preserved under Householder's error.

# Stability and size-reduction are inconsistent

$$\begin{pmatrix} \ddots & & \eta & & \\ & R_{i,i} & \leftarrow & R_{i,j} & \\ & & \ddots & \downarrow & \theta \\ & (0) & & R_{j,j} & \\ & & & & \ddots \end{pmatrix}$$

New size-reduction: $|R_{i,j}| \leq \eta R_{i,i} \quad \rightarrow \quad |R_{i,j}| \leq \eta R_{i,i} + \theta R_{j,j}$.



# Outline

# H-LLL

Principle:

- Size-reducing using $\widetilde{R}$ computed by Householder.
- Checking the Lovász condition using $\widetilde{R}$.

# H-LLL

Principle:

- Size-reducing using $\widetilde{R}$ computed by Householder.
- Checking the Lovász condition using $\widetilde{R}$.

Challenges:

- The current vector is not (yet) reduced.
- The error made on the corresponding column of $R$ is proportional to the norm of that vector.

# Lazy size-reduction

- Only the most significant bits of $R_\kappa$ are known.

# Lazy size-reduction

- Only the most significant bits of $R_\kappa$ are known.
- Size-reduction is made in several steps.
- Each step decreases the norm of the vector.

# Lazy size-reduction

- Only the most significant bits of $R_\kappa$ are known.
- Size-reduction is made in several steps.
- Each step decreases the norm of the vector.
- Size-reduction stops when the norm no longer decreases.

# Lazy size-reduction

- Only the most significant bits of $R_\kappa$ are known.
- Size-reduction is made in several steps.
- Each step decreases the norm of the vector.
- Size-reduction stops when the norm no longer decreases.
- This ensures new bits are discovered at each step.

# Complexity analysis

- The number of steps involves the norm progress:

$$O\left(1 + \frac{1}{d} \log \frac{\|\mathbf{b}_\kappa^{initial}\|}{\|\mathbf{b}_\kappa^{final}\|}\right).$$

- Each step costs $O(nd^2(d + \log \|B\|))$ operations.

# Complexity analysis

- The number of steps involves the norm progress:

$$O\left(1+\frac{1}{d}\log\frac{\|\mathbf{b}_\kappa^{initial}\|}{\|\mathbf{b}_\kappa^{final}\|}\right).$$

- Each step costs $O(nd^2(d+\log\|B\|))$ operations.
- The overall complexity:

$$O\left[\left(d+\log\prod\frac{d_i^{initial}}{d_i^{final}}+\frac{1}{d}\log\prod\frac{\|\mathbf{b}_i^{initial}\|}{\|\mathbf{b}_i^{final}\|}\right)nd^2(d+\log\|B\|)\right],$$

where $d_i$ is the determinant of the $i\times i$ matrix $(\langle\mathbf{b}_j;\mathbf{b}_k\rangle)_{j,k\leq i}$.

# Conclusion and current work

- A new and more intuitive algorithm.
- A smaller required precision.
- C++ implementation inside fplll.
- Decreasing the required precision even more.