

Hermite's Constant and Lattice Algorithms

Phong Nguyễn

<http://www.di.ens.fr/~pnguyen>

and Ecole normale supérieure



April 27th, 2007

A Cryptographic Earthquake

- In 2004, Wang announced the first MD5 collision:

Sequence #1															
d1	31	dd	02	c5	e6	ee	c4	69	3d	9a	06	98	af	f9	5c
2f	ca	b5	87	12	46	7e	ab	40	04	58	3e	b8	fb	7f	89
55	ad	34	06	09	f4	b3	02	83	e4	88	83	25	71	41	5a
08	51	25	e8	f7	cd	c9	9f	d9	1d	bd	f2	80	37	3c	5b
d8	82	3e	31	56	34	8f	5b	ae	6d	ac	d4	36	c9	19	c6
dd	53	e2	b4	87	da	03	fd	02	39	63	06	d2	48	cd	a0
e9	9f	33	42	0f	57	7e	e8	ce	54	b6	70	80	a8	0d	1e
c6	98	21	bc	b6	a8	83	93	96	f9	65	2b	6f	f7	2a	70

Sequence #2															
d1	31	dd	02	c5	e6	ee	c4	69	3d	9a	06	98	af	f9	5c
2f	ca	b5	07	12	46	7e	ab	40	04	58	3e	b8	fb	7f	89
55	ad	34	06	09	f4	b3	02	83	e4	88	83	25	f1	41	5a
08	51	25	e8	f7	cd	c9	9f	d9	1d	bd	72	80	37	3c	5b
d8	82	3e	31	56	34	8f	5b	ae	6d	ac	d4	36	c9	19	c6
dd	53	e2	34	87	da	03	fd	02	39	63	06	d2	48	cd	a0
e9	9f	33	42	0f	57	7e	e8	ce	54	b6	70	80	28	0d	1e
c6	98	21	bc	b6	a8	83	93	96	f9	65	ab	6f	f7	2a	70

Both produce MD5 digest: **76dc611d6ebaafc66cc0879c71b5db5c**

- The NIST is planning a call in 2008 for a new hash function standard.

What happened for
hash functions...

Could that happen
to public-key cryptography?

Hard Problems for Public-Key Cryptography

- “Classical” problems
 - Integer Factorization (RSA)
 - Discrete Logarithm (DSA, ECC, Pairings)
- More “exotic” problems
 - Lattice reduction (NTRU)
 - Coding theory problems (McEliece)
 - Multivariate polynomials over finite fields

Are factoring and discrete logarithms really hard?

- No NP-hardness argument.
- Yet nobody knows how to solve them efficiently since their introduction, except with... quantum computers.
- But how deep is the connection with mathematics?

In this talk

- We will look at one of the “exotic” problems used in public-key cryptography: **lattice reduction**.
- We will argue that there is a deep connection between the main lattice algorithms and mathematics, which is highlighted by **Hermite’s constant**.

Lattice Reduction

- Surprisingly many applications
 - **algorithmic number theory** (factorization, Diophantine approximation, etc.)
 - **theoretical CS** (integer programming, complexity, etc.)
 - **cryptology** (cryptanalysis, crypto design)

Lattice-based Cryptography

- Can be much more efficient than RSA.
- Potentially resistant to quantum computers.
- Can have security properties based on worst-case assumptions.

Summary

- Hermite's Constant and Lattices
- Links between Hermite's Constant and lattice algorithms
 - The 2-dimensional case
 - Upper bounds on Hermite's Constant
 - Lower bound on Hermite's Constant

A historical problem

A horizontal line of red and black ink scribbles, appearing as if drawn with a marker or brush, extending across the width of the page below the text.

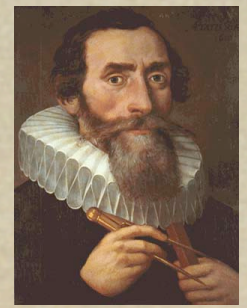
Sphere Packings



The Hexagonal Packing



Kepler's "Conjecture" (1611)



This is the best packing in dimension 3.
[Hales2005]

Beyond Kepler's Conjecture

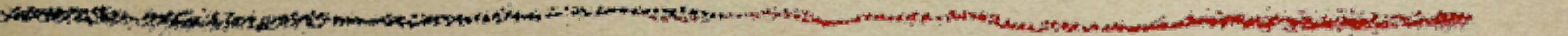
- What is the best sphere packing in higher dimension?



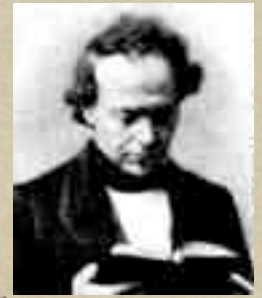
- What if we restrict to regular packings, e.g. **lattice packings**? Those are optimal in dim 2 and 3.

- This is related to Hermite's constant, and motivated the study of lattices: **geometry of numbers**.

Hermite's Constant



Hermite's Constant



- Let q be a positive definite quadratic form over \mathbb{R}^n :

$$q(x_1, \dots, x_n) = \sum_{1 \leq i, j \leq n} q_{i,j} x_i x_j$$

- Its discriminant is $\Delta(q) = \det(q_{i,j})_{1 \leq i, j \leq n}$

- It has a minimum $\|q\|$ over $\mathbb{Z}^n \setminus \{0\}$

- Hermite proved the existence of:

$$\gamma_n = \max_{q \text{ over } \mathbb{R}^n} \frac{\|q\|}{\Delta(q)^{1/n}}$$

Facts on Hermite's Constant



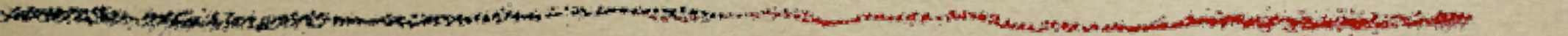
- Hermite's constant is asymptotically **linear**:

$$\Omega(n) \leq \gamma_n \leq O(n)$$

- We may restrict to **integral** quadratic forms.
- The exact value of the constant is only known up to dim 8, and in dim 24 [2004].

dim n	2	3	4	5	6	7	8	24
γ_n	$2/\sqrt{3}$	$2^{1/3}$	$\sqrt{2}$	$8^{1/5}$	$(64/3)^{1/6}$	$64^{1/7}$	2	4
approx	1.16	1.26	1.41	1.52	1.67	1.81	2	4

What does it have
to do with packings?



Lattices and Quadratic Forms

○ Let $\vec{b}_1, \dots, \vec{b}_d \in \mathbb{R}^n$ be linearly independent.

○ They define a positive definite quadratic form:

$$q(x_1, \dots, x_d) = \left\| \sum_{i=1}^d x_i \vec{b}_i \right\|^2$$

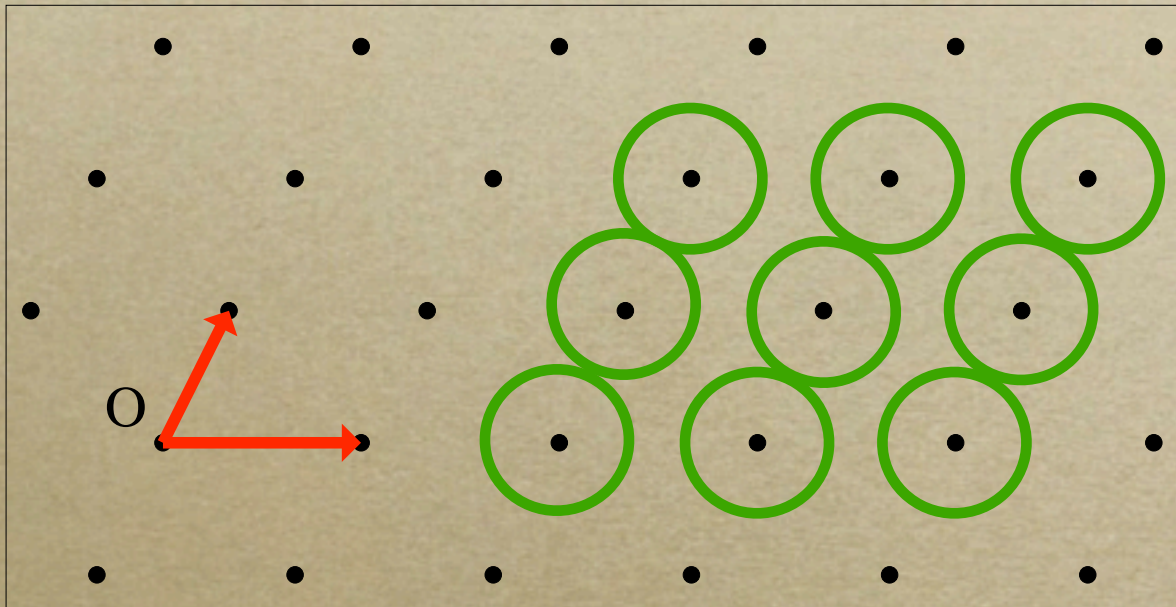
○ They also define a d-rank lattice:

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

○ Bilinear algebra shows a reciprocal: for every positive definite quadratic form...

Lattice Packings

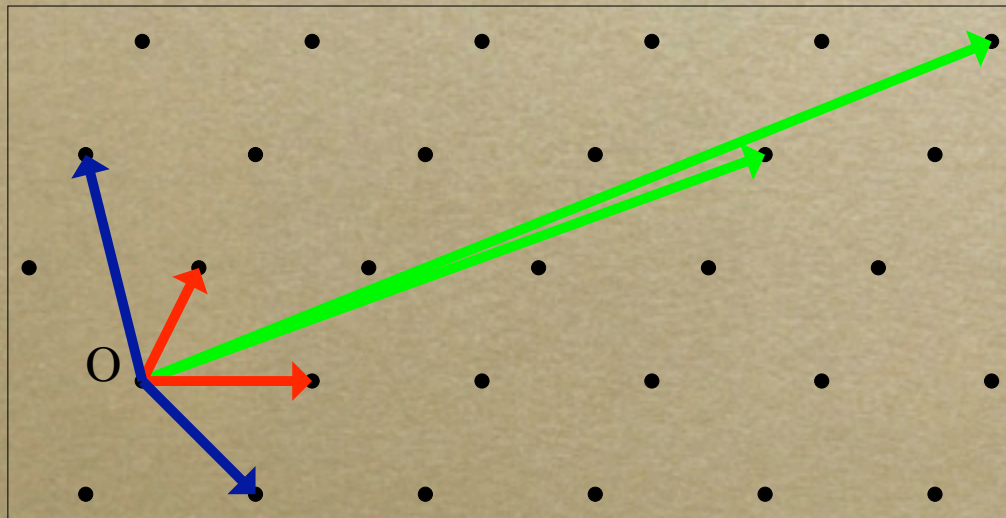
- Every lattice defines a sphere packing:



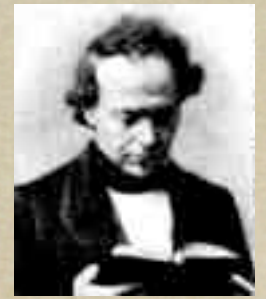
- The diameter of spheres is the **first minimum** of the lattice: the shortest norm of a non-zero lattice vector.

Volume of a Lattice

- Each basis spans a parallelepiped, whose volume only depends on the lattice. This is the **lattice volume**.



Hermite's Constant Again



○ We have:

$$\gamma_n = \max_q \frac{\|q\|}{\Delta(q)^{1/n}} = \max_L \frac{\|L\|^2}{\text{vol}(L)^{2/n}}$$

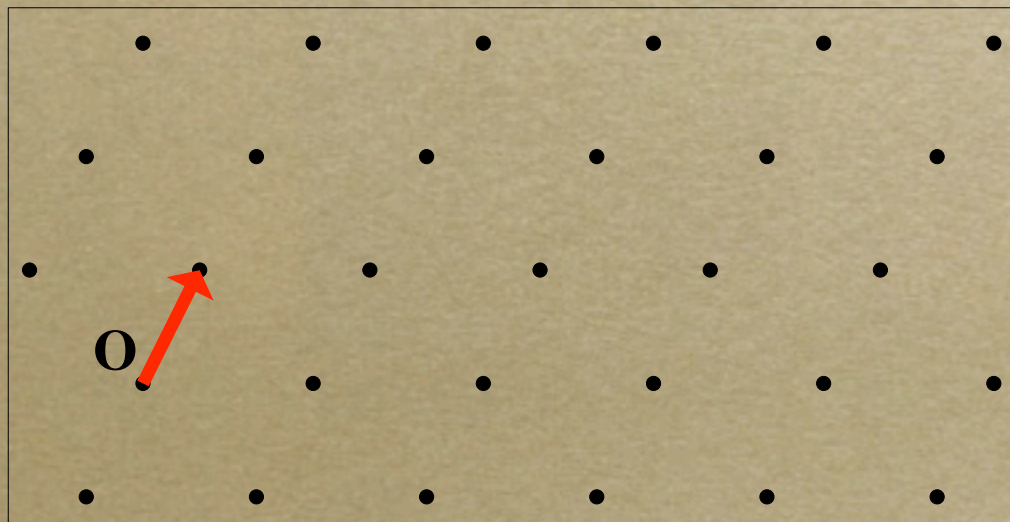
○ The optimal lattice packings correspond to the **critical lattices**, those reaching Hermite's constant.

Lattice Algorithms

- The input is an integer matrix. Parameters:
 - The size of basis coefficients
 - The lattice dimension
- Asymptotically, the dimension increases, and the size of coeffs is polynomial in the dimension.

The Shortest Vector Problem (SVP)

- Input: a basis of a lattice L of dim d .
- Output: $v \in L$ such that $\|v\| = \|L\|$.
- Approximate-SVP: $\|v\| \leq f(d) \|L\|$
- Hermite-SVP: $\|\vec{v}\| \leq f(d) \text{vol}(L)^{1/d}$



SVP Algorithms

- Polynomial-time approximation algorithms.
 - The LLL algorithm [1982].
 - Its block generalization by Schnorr [1987] and Gama et al [2006].
- Exponential exact algorithms.
 - Deterministic [Kannan1983].
 - Randomized [AKS2001].

SVP Algorithms

- Both categories are **complementary**
 - Approximation algorithms in high dimension use an exact algorithm in low dimension (around 20 in practice).
 - Exact algorithms rely on approximation algorithms as preprocessing. They are practical up to dim 50.

Approximation Algorithms for SVP

- All related to historical methods to upper bound Hermite's constant.

- [LLL82] corresponds to [Hermite1850].

$$\gamma_d \leq (4/3)^{(d-1)/2} = \gamma_2^{d-1}$$

- [Schnorr87] and [GHKN06] correspond roughly to Mordell's inequality.

$$\gamma_d \leq \gamma_k^{(d-1)/(k-1)}$$

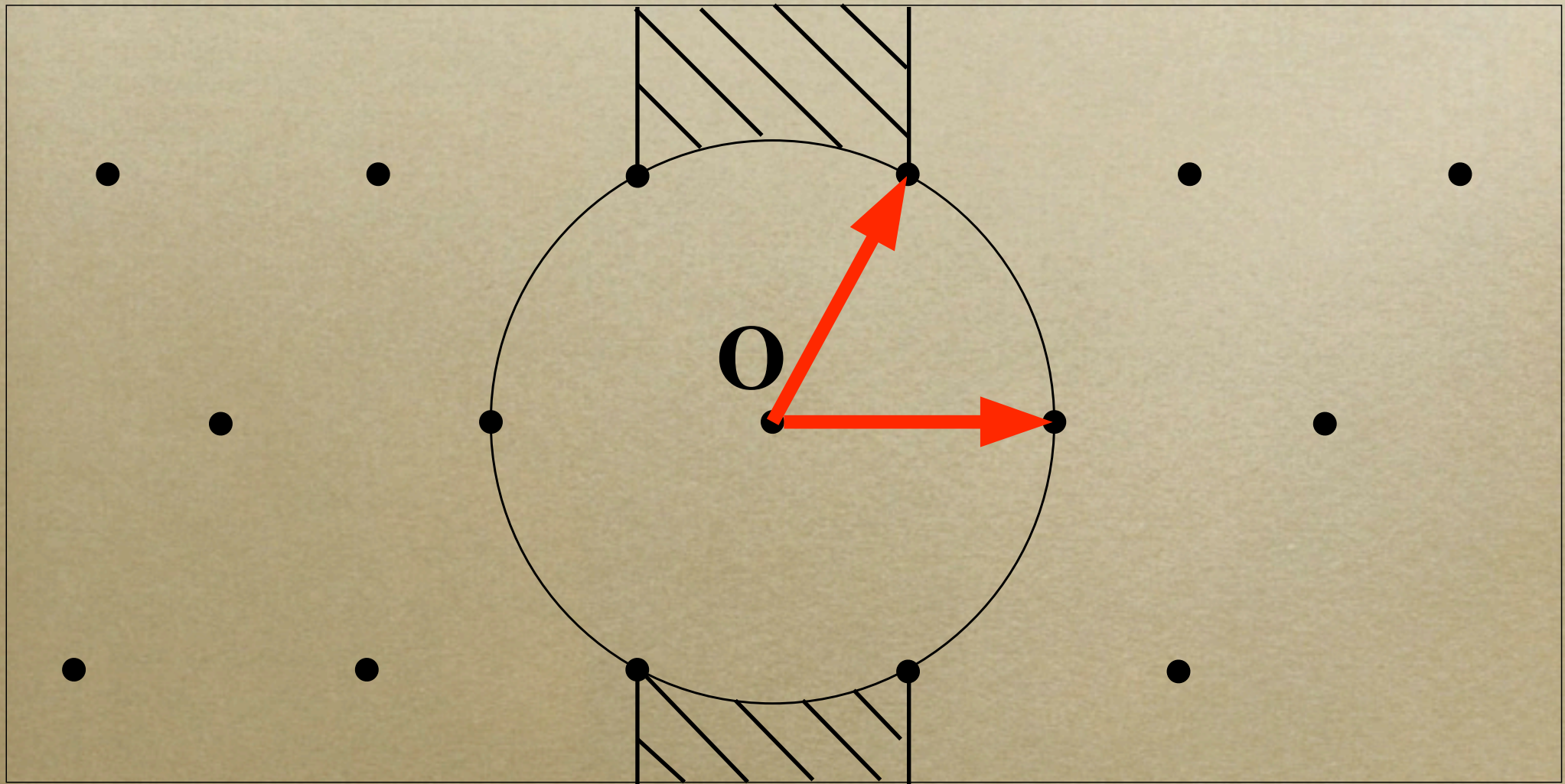
Rest of the Talk

- The 2-dimensional case
- Upper bounds on Hermite's Constant
- Lower bound on Hermite's Constant

The 2-Dimensional Case



The 2-dimensional Case



$$\gamma_2 = \sqrt{4/3}$$

The 2-dim Case

- By proving that $\gamma_2 \leq (4/3)^{1/2}$, we also described an algorithm (found by **Lagrange**, and also known as Gauss' algorithm) finding a basis such that:

$$\|\vec{b}_1\| \leq \|\vec{b}_2\| \quad |\langle \vec{b}_1, \vec{b}_2 \rangle| \leq \|\vec{b}_1\|^2/2$$

- Then the first vector is a shortest vector.

The n -Dimensional Case

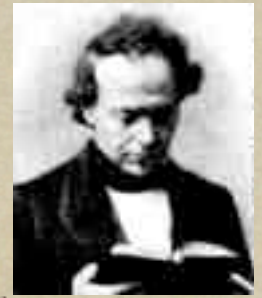


Hermite's Inequality



- Hermite proved $\gamma_d \leq (4/3)^{(d-1)/2}$ as a generalization of the 2-dim case by induction over d .
- Consider a shortest lattice vector, and project the lattice orthogonally to that vector...

Hermite's Inequality



- The proof actually suggests a **recursive algorithm** to find a lattice vector of norm:
$$\leq (4/3)^{(d-1)/4} \text{vol}(L)^{1/d}$$
- Project the lattice orthogonally to the first vector b_1 .
- Find a short vector in the projected lattice.
- Lift it and see if it is shorter than b_1 .

LLL and Hermite's Inequality

- The LLL algorithm [1982] is a relaxed variant of Hermite's algorithm: it is the first lattice algorithm to **provably run in polynomial time**. It approximates SVP within an exponential factor: the one corresponding to Hermite's inequality.
- [N-Stehlé2005] decreased the running time of LLL to an Euclid-like running time.

Algorithms based on Hermite's Inequality

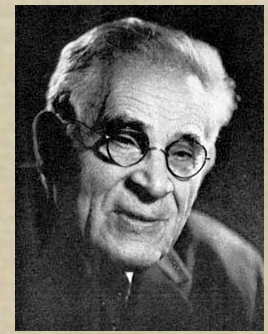
- They reduce all the 2×2 lattices.

$$\begin{pmatrix} a_{1,1} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & 0 & \dots & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 0 & \vdots \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & \dots \\ \vdots & & & & \\ a_{d,1} & a_{d,2} & \dots & a_{d,d-1} & a_{d,d} \end{pmatrix}$$

Using LLL to solve exact SVP

- LLL gives rise to a **super-exponential algorithm** for SVP [Ka1983], based on exhaustive search. If we use that SVP algorithm in low dimension as an oracle, can we improve the approximation factor of LLL, e.g. divide and conquer?

Mordell's Inequality



- Hermite's inequality is a particular case of Mordell's inequality:

$$\gamma_d \leq \gamma_k^{(d-1)/(k-1)} \quad \text{if } 2 \leq k \leq d$$

- The standard proof of Mordell's inequality is based on primal/dual transfers.
- Mordell's inequality is tight for $(k,d)=(3,4)$ and $(7,8)$.

Mordell's Inequality Algorithmically

- Given an SVP-oracle in dim k , [Sc87,GKHN06] approximate SVP in dim d , with a similar dependence on the approx. factors as in Mordell's inequality

$$\gamma_d \leq \left(\gamma_k^{1/(k-1)} \right)^{d-1} = O \left(\left(k^{1/(k-1)} \right)^{d-1} \right)$$

- By choosing an appropriate $k=f(d)$, the whole algorithm is **poly-time with a subexponential approx factor.**

From LLL to Block Reduction

- LLL tries to reduce all the 2×2 lattices.

$$\begin{pmatrix} a_{1,1} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & 0 & \dots & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 0 & \vdots \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & \dots \\ \vdots & & & & \\ a_{d,1} & a_{d,2} & \dots & a_{d,d-1} & a_{d,d} \end{pmatrix}$$

Block-Reduction

- Try to reduce all the $2k$ -dim lattices.

$$\begin{pmatrix} a_{1,1} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & 0 & \dots & 0 \\ a_{3,1} & a_{3,2} & a_{3,3} & 0 & \dots & \vdots \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & \dots & \\ \vdots & & & & & \\ a_{d,1} & a_{d,2} & \dots & a_{d,d-1} & a_{d,d} \end{pmatrix}$$

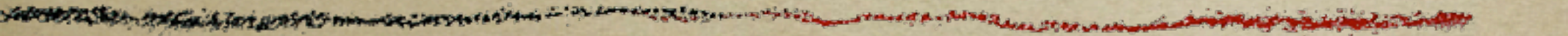
Limits of Approximation Algorithms

- Since Mordell's inequality can be tight, it seems difficult to improve the block strategy.
- If the algorithm also provides an absolute upper bound on the output, it implicitly gives an upper bound on Hermite's constant. This was the case for LLL and Schnorr's algorithm.

Speculation

- If LLL and Schnorr's algorithm correspond to classical inequalities on Hermite's constant, do other methods for bounding Hermite's constant have algorithmic analogues?
 - Minkowski's Convex Body Theorem or Blichfeldt's lemma.
 - The method of [CohnKumar2004].

Lower bound on Hermite's Constant





The Haar Measure

- Lebesgue's measure is the "unique" measure over \mathbb{R}^n which is invariant by translation.
- In 1933, Haar generalized Lebesgue's measure to **locally compact topological groups**: it is the "unique" measure which is invariant by the group action (left or right multiplication).

Random Lattices

- The set of **lattices modulo scale** can be identified with $G = \mathrm{SL}_n(\mathbb{R}) / \mathrm{SL}_n(\mathbb{Z})$.
- The Haar measure over $\mathrm{SL}_n(\mathbb{R})$ projects to a **finite** measure over G .
- We thus have a natural probability measure over G , which gives rise to **random lattices**.

Properties of Random Lattices

- The Minkowski–Hlawka theorem gives an asymptotic estimate of the first minimum of random lattices.
- This provides the best **lower bound** known on Hermite's constant.

Random Lattices and Algorithms

- It is possible to “generate” random lattices in polynomial time [GoMa03,Aj06].
- Is it possible to analyze the running-time and the output quality of lattice algorithms on random lattices?
- Are cryptographic lattices different from random lattices?

Open Problems



Speculation

- Extremely few algorithms and rather “elementary” techniques: have we looked hard enough, or is it really difficult?
- Could we (better) exploit **duality**?
- Could we exploit **randomness** and **distributions**?

The biggest open problem

- Efficient algorithms to approximate SVP within a polynomial factor.

Winning 25,100\$

- On ntru.com, one can download a 502-dim integer lattice.
- They offer 25,100\$ to whoever finds the shortest vector.
- More money in higher dimension.

More on LLL and lattices

- At the end of June 2007, there'll be a special conference celebrating the 25th anniversary of the LLL algorithm.
See <http://lll25.info.unicaen.fr/>
- This LLL+25 conference will present a dozen of surveys on the main applications of lattices in mathematics and computer science.