The LLL Algorithm

Phong Nguyễn

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

May 2010, Luminy

1982

H. Lenstra

What is LLL or L^3 ?

The LLL Algorithm

- A popular algorithm presented in a legendary article published in 1982:

Math. Ann. 261, 515–534 (1982)

Mathematische Annalen

Factoring Polynomials with Rational Coefficients

A. K. Lenstra[1], H. W. Lenstra, Jr.[2], and L. Lovász[3]

1 Mathematisch Centrum, Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands

2 Mathematisch Instituut, Universiteit van Amsterdam, Roetersstraat 15, NL-1018 WB Amsterdam, The Netherlands

3 Bolyai Institute, A. József University, Aradi vértanúk tere 1, H-6720 Szeged, Hungary

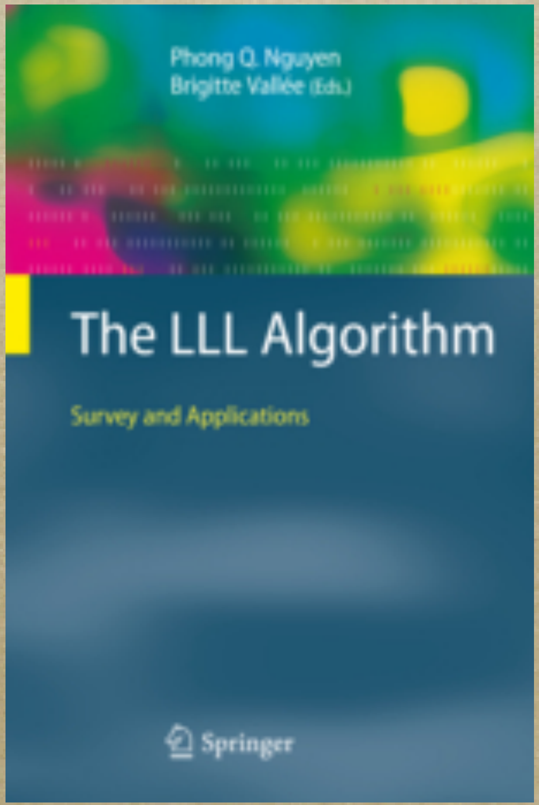
In this paper we present a polynomial-time algorithm to solve the following problem: given a non-zero polynomial $f \in \mathbb{Q}[X]$ in one variable with rational coefficients, find the decomposition of f into irreducible factors in $\mathbb{Q}[X]$. It is well

How Popular?

- The LLL article has been cited x1000 times.
- The LLL algorithm and/or variants are implemented in:
  - Maple
  - Mathematica
  - GP/Pari
  - Magma
  - NTL/SAGE, etc.

How Popular?

- A conference was organized in 2007 to celebrate the 25th anniversary of the LLL article.
- This gave rise to a book:



What is LLL about?

- It is an efficient algorithm.
- But it's not about:

Factoring Polynomials with Rational Coefficients

A. K. Lenstra[1], H. W. Lenstra, Jr.[2], and L. Lovász[3]

- It's about finding short lattice vectors.

coefficients of h_0 are relatively small. It follows that we must look for a "small" element in that lattice, and this is done by means of a basis reduction algorithm. It

Intuitively

- LLL is a vectorial analogue of Euclid's algorithm to compute gcds.
  - Instead of dealing with integers, it deals with vectors of integer coordinates.
  - It performs similar operations, and is essentially as efficient.

More Precisely

- We will present LLL as an algorithmic version of Hermite's inequality on Hermite's constant.
- It is essentially a variant of an implicit algorithm published by Hermite in 1850.

Extraits de lettres de M. Ch. Hermite à M. Jacobi sur différents objets de la théorie des nombres.

Première lettre.

Près de deux années se sont écoulées, sans que j'aie encore répondu à la lettre pleine de bonté, que Vous m'avez fait l'honneur de m'écrire *). Aujourd'hui, je viens Vous supplier de me pardonner ma longue négligence, et Vous exprimer toute la joie, que j'ai ressentie en me voyant une place dans le recueil de Vos oeuvres. Depuis long-temps éloigné du travail, j'ai été bien

Applications of LLL

- Linear algebra with "small" integers
- Cryptananalysis: breaking cryptosystems based on number theory
- Algorithmic number theory
- Complexity theory

Examples

- This formula for π was found in 1995 using a variant of LLL:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right),$$

- Elkies used LLL in the 2000s to find:
- $5853886516781223^3 - 447884928428402042307918^2 = 1641843$
- Odlyzko and te Riele used LLL in 1985 to disprove the Mertens conjecture.

Examples

- The two-square theorem: If p is a prime $\equiv$ 1 mod 4, then p is a sum of two squares $p=x^2+y^2$.
- To find such x and y, one may first compute a square root of -1 mod p, then use LLL.

Examples

- Breaking the Merkle-Hellman cryptosystem (early competitor to RSA):
  - Published in 1978, like RSA.
  - Broken by Shamir in 1982: key-recovery attack.

- Since 1982, dozens of public-key cryptosystems have been broken using LLL.

Examples

- The factorization record (Dec. 2009) for RSA numbers is a 768-bit number of the form N=pq: 232 digits.
- In the last stage, LLL was used hundreds of thousands of times, to compute square roots of huge algebraic numbers, yielding after 1500 core years...

RSA-768

- 123018668453011775513049495838496272077285356959 533479219732245215172640050726365751874520219978 646938995647494277406384592519255732630345373 15 482685079170261221429134616704292143116022212404 79 274737794080665351419597459856902143413
- =33478071698956898786044169848212690817704794983 7137685689124313889828837938780022876147165253174 3087737814467999489 x 367460436667995904282446337996279526322791581 64 343087642676032283815739666511279233373417143396 81027092798736308917

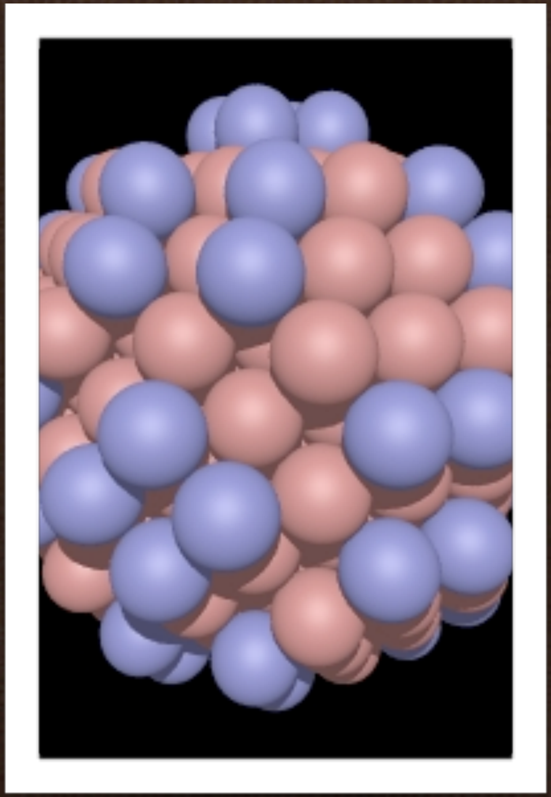
Summary

- History
- Background on Lattices
- The LLL approximation algorithm
- A few applications

Lattices in Cryptology

- Cryptanalysis
  - Lattice reduction algorithms are arguably the most popular tools in public-key cryptanalysis (RSA, DSA, knapsacks, etc.)
- Crypto design
  - Lattice-based cryptography is arguably the main alternative to RSA/ECC.
  - A unique property: worst-case assumptions.

A Historical Problem



Sphere Packings

The Hexagonal Packing

Kepler's "Conjecture" (1611)

What is the best packing in dim 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 motivated the study of lattices: geometry of numbers.

Significance

- Since the 18th century, mathematicians have been interested in proving the existence of short lattice vectors: bounds valid for any lattice in a given dimension.
- This is related to the best lattice packings.

Another motivation...
Euclid’s Algorithm

Euclid's Algorithm

- **Input**: two integers $a \geq b \geq 0$.
- **Output**: $\gcd(a,b)$.
- While $(b \neq 0)$
  - $a := a \bmod b$
  - Swap(a,b)
- Output(a)

Classical Results on Euclid's Algorithm

- What is the complexity of Euclid's algorithm using standard arithmetic?
- No more than multiplying large integers, using basic techniques.

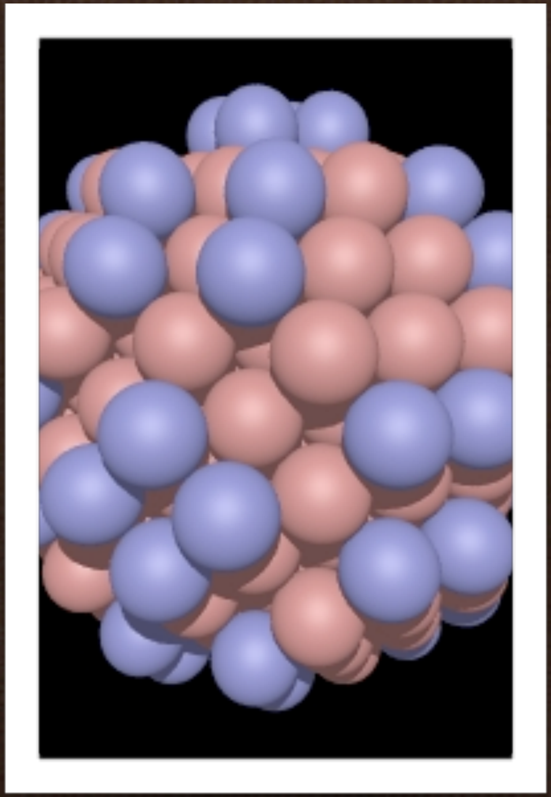
A generalization

- In 1773, Lagrange  notices that Euclid's algorithm answers the following question: given (n,a,b), is n of the form $ax+by$?
- He invents algorithms for this generalization: given (n,a,b,c), is n of the form $ax^2+bxy+cy^2$?

A Vectorial Euclid's Algorithm?

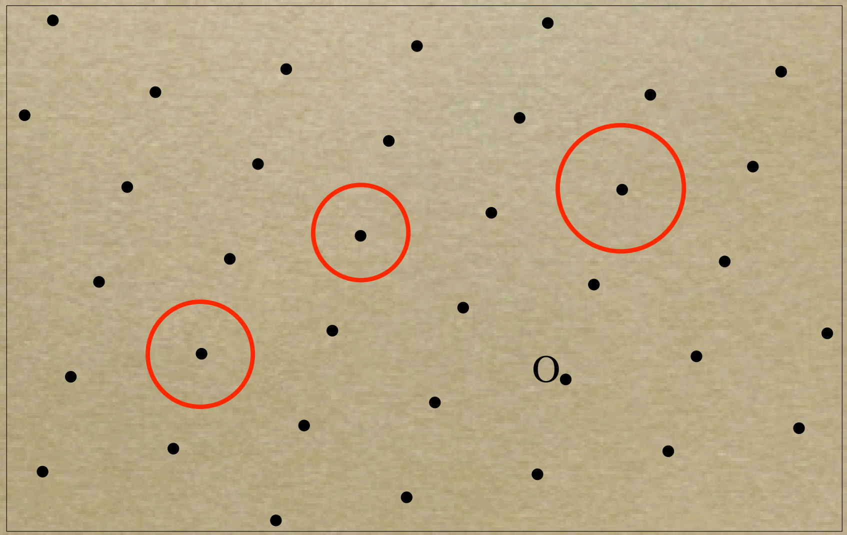
- Since $a\mathbf{Z}+b\mathbf{Z}=\gcd(a,b)\mathbf{Z}$, Euclid computes the shortest non-zero linear combination of a and b.
- Given a finite set B of vectors in $\mathbf{Z}^n$, can one compute the shortest non-zero vector in the set L(B) of all linear combinations?

Background on Lattices



Euclidean Lattices

- Consider $\mathbf{R}^n$ with the usual topology of a Euclidean space: let $\langle u,v \rangle$ be the dot product and $\|w\|$ the norm.
- A lattice is a discrete subgroup of $\mathbf{R}^n$.
- Ex: $\mathbf{Z}^n$ and its subgroups.



Exercises

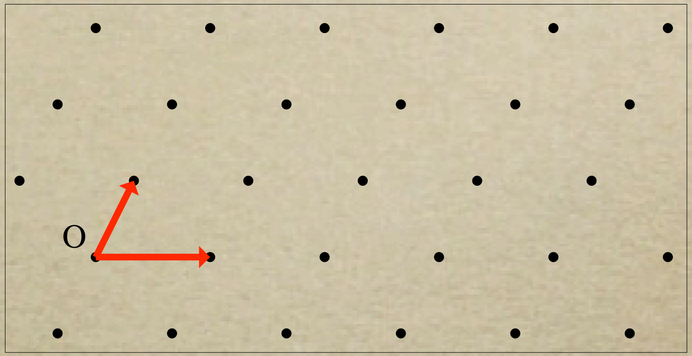
- Show that for any lattice L of $\mathbf{R}^n$:
  - $\exists r>0$ s.t. $\forall x \in L$, $L \cap B(x,r) = \{x\}$.
  - L is closed.
  - For any bounded subset S of $\mathbf{R}^n$, its intersection with L is finite.
  - L is countable.

Examples

- Let $b_1,b_2,\ldots b_d$ in $\mathbf{Q}^n$.
  - Then $L(b_1,\ldots,b_d)$ is a lattice.
- Let $b_1,b_2,\ldots b_d$ be linearly independent vectors in $\mathbf{R}^n$.
  - Then $L(b_1,\ldots,b_d)$ is a lattice.

Characterization of Lattices

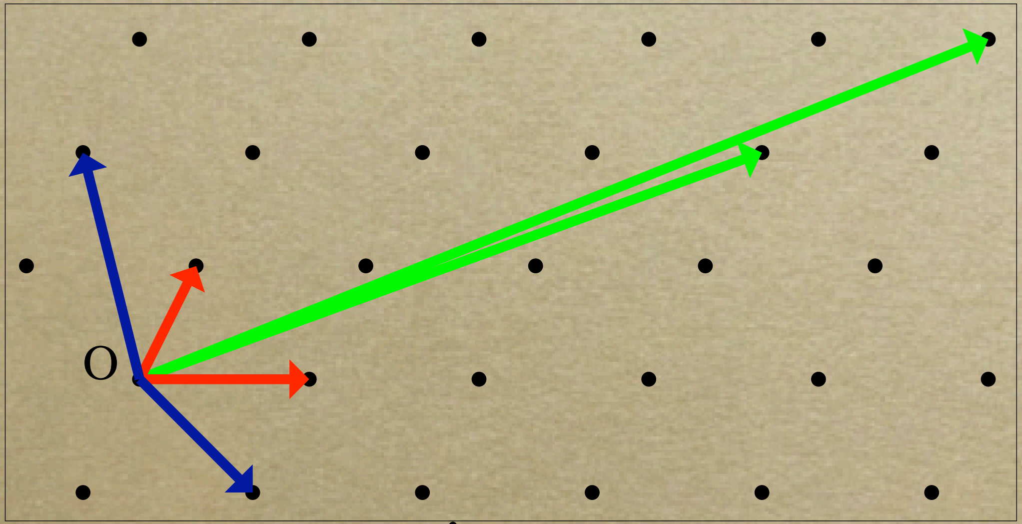
- Let L be a non-empty set of R^n. There is equivalence between:
  - L is a lattice.
  - There exists a set B of linearly independent vectors such that L=L(B).



- Such a B is a basis of a lattice L, and its cardinality is the dimension/rank of the lattice.

Volume of a Lattice

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



- By scaling, we can always ensure that the volume is 1 like $\mathbf{Z}^n$.

Lattices and Quadratic Forms

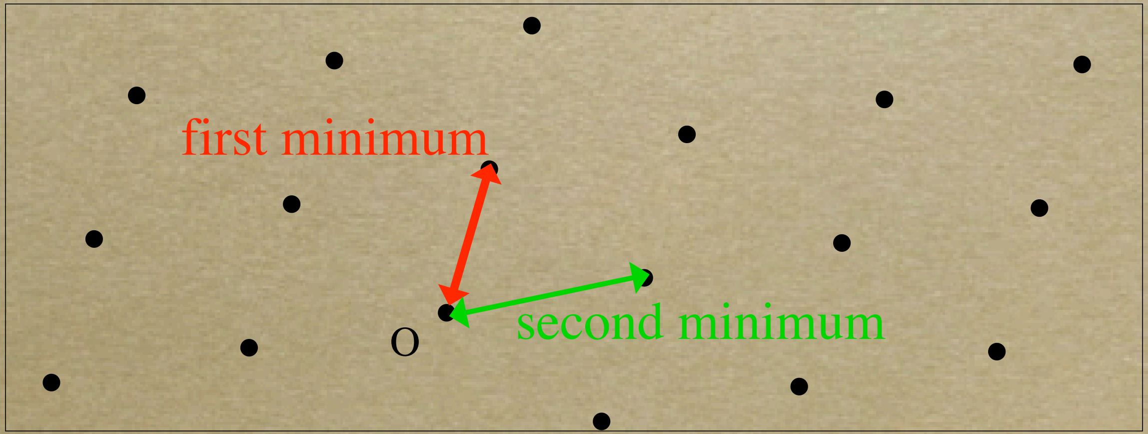
- Every lattice basis defines a positive definite quadratic form:

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

- Reciprocally: Cholesky factorization.
- The squared volume is the discriminant of the form.

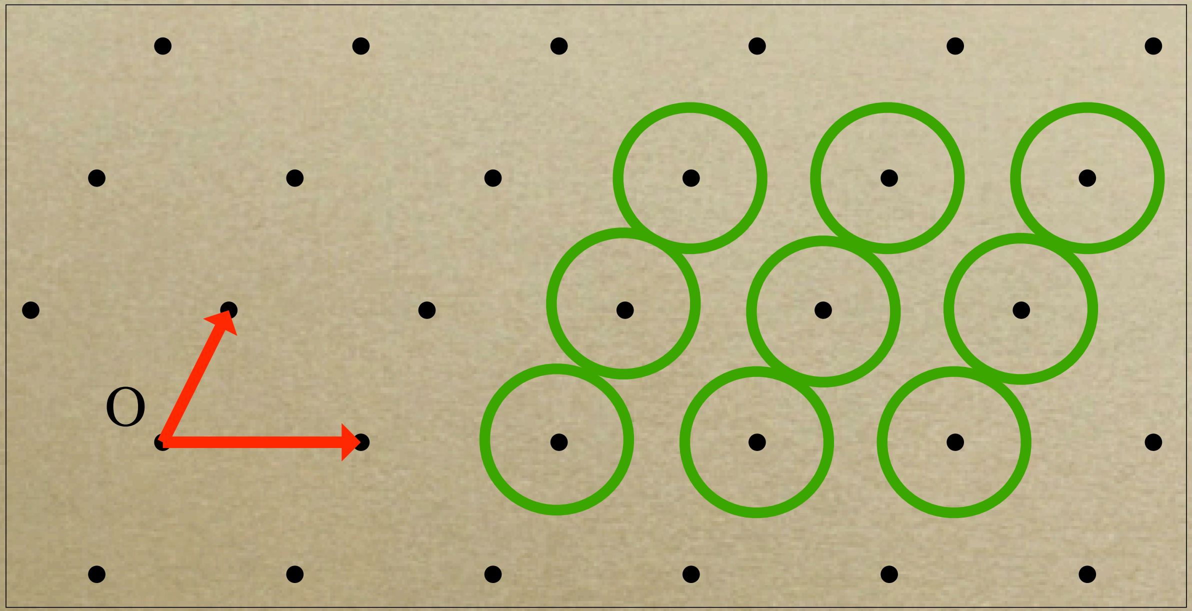
The First Minimum

- The intersection of a lattice with any bounded set is finite.
- In a lattice L, there are non-zero vectors of minimal norm: this is the first minimum $\lambda_1(L)$ or the minimum distance.



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.

Hermite's Constant (1850)

Hermite's Constant

- Let q be a positive definite quadratic form over $\mathbf{R}^n$: $$q(x_1,\ldots,x_n) = \sum_{1\le i,j\le n} q_{i,j}x_ix_j$$
  - Its discriminant is $\Delta(q) = \det(q_{i,j})_{1\le i,j\le n}$
  - It has a minimum $||q||$ over $\mathbf{Z}^n\backslash\{0\}$
- Hermite (1850) proved the existence of: $$\gamma_n = \max_{q \text{ over } \mathbb{R}^n} \frac{||q||}{\Delta(q)^{1/n}}$$

Hermite's Constant Again

- We have:

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

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

Facts on Hermite's Constant

- Hermite's constant is asymptotically linear:

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

- 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

Application:
the two-square theorem

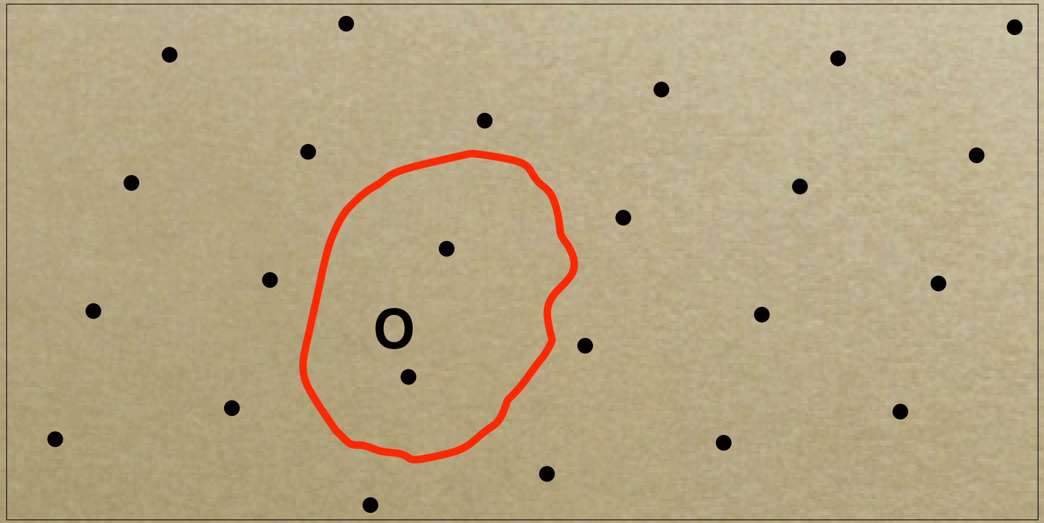
- Let p be a prime $\equiv 1 \bmod 4$.
- Then -1 is a square mod p: there exists r s.t. $r^2 \equiv 1 \bmod p$.
- Then $x^2+y^2 \equiv (x+ry)(x-ry) \bmod p$.
- Let $L=\{(x,y)\in\mathbf{Z}^2$ s.t. $x \equiv ry \bmod p\}$.

Application: the two-square theorem

- Let $L=\{(x,y)\in\mathbf{Z}^2$ s.t. $x \equiv ry \bmod p\}$. This is a lattice of dimension 2, with volume p.
- There must be a non-zero vector (x,y) in L of squared norm $\leq 2p/\sqrt{3}$. Then:
  - $x^2+y^2 \equiv 0 \bmod p$
  - $0 < x^2+y^2 \leq 2p/\sqrt{3}$
- Therefore $p=x^2+y^2$.

The existence of short lattice vectors

- Hermite proved in 1850: $\gamma_d \leq \left(\frac{4}{3}\right)^{(d-1)/2}$
- Minkowski's theorem implies: $\gamma_d \leq d$

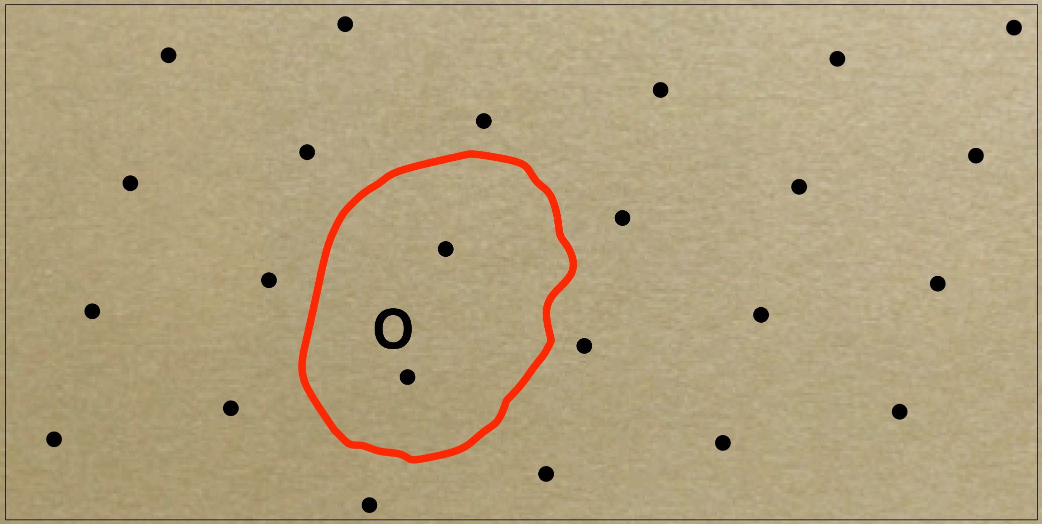


- Thus, any lattice contains a non-zero vector of norm $\leq \sqrt{d}\mathrm{vol}(L)^{1/d}$

Linear Bounds on Hermite's Constant

Minkowski's Theorem (1896)

- Let L be a full-rank lattice of R^n. Let C be a measurable subset of R^n, convex, symmetric, and of measure > 2^nvol(L).
- Then C contains at least a non-zero point of L.



Remarks

- The volume bound is optimal in the worst-case.
- If C is furthermore compact, the $>$ can be replaced by $\geq$.

Application to a ball

- Let C be the n-dim ball of radius r. Then its volume is r^n multiplied by:

$$v_n = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(1+\frac{n}{2}\right)} \sim \left(\frac{2e\pi}{n}\right)^{\frac{n}{2}} \frac{1}{\sqrt{\pi n}}$$

- To apply Minkowski's theorem, one can take:

$$r = \frac{2}{(v_n)^{\frac{1}{n}}} vol(L)^{\frac{1}{n}}$$

Application to a ball

- We obtain Minkowski's linear bound on Hermite's constant:

$$\sqrt{\gamma_n} \leq \frac{2}{(v_n)^{\frac{1}{n}}} = 2\,\frac{\Gamma\left(1+\frac{n}{2}\right)^{\frac{1}{n}}}{\sqrt{\pi}} \sim 2\sqrt{\frac{n}{2\pi e}}$$

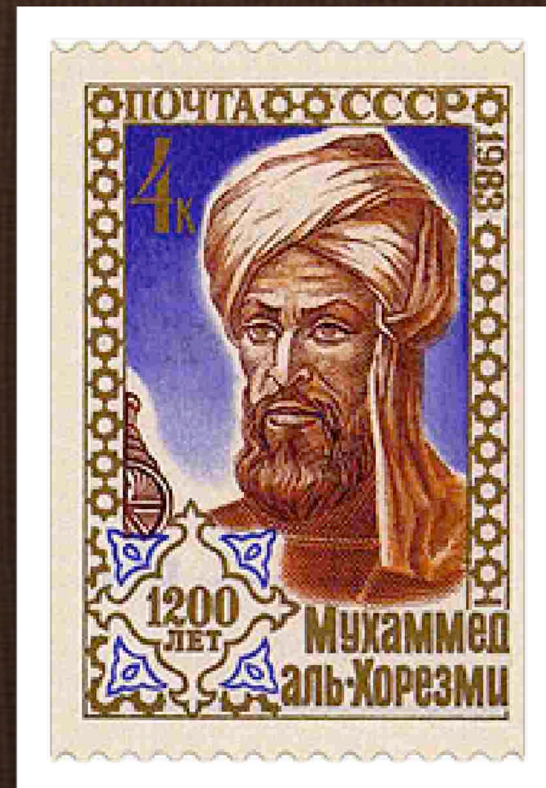
Proving Minkowski

- Blichfeldt's lemma:
  - Let L be a full-rank lattice of R^n.
  - Let F be a measurable subset of R^n, of measure > vol(L).
- Then F contains at least two distinct vectors whose difference is in L.

Other Proofs of Minkowski's Upper Bound

- Minkowski's original proof: using packings.
- Mordell's proof.

Lattice Algorithms



Algorithmic Problems

- There are two parameters:
  - The size of basis coefficients
  - The lattice dimension
- Two cases
  - Fixed dimension, the size of coeffs increases.
  - The dimension increases, and the size of coeffs is polynomial in the dimension.

Lattices and Complexity

- Since 1996, lattices are very trendy in complexity: classical and quantum.
- Depending on the approximation factor with respect to the dimension:
  - NP-hardness
  - non NP-hardness (NP∩co-NP)
  - worst-case/average-case reduction
  - polynomial-time algorithms

1

$O(1)$

$\sqrt{n}$

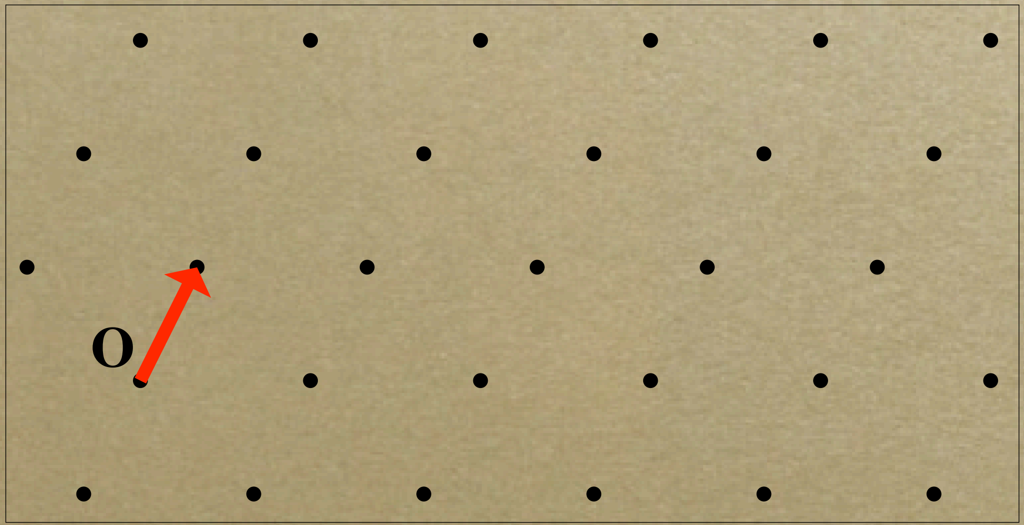
$O(n \log n)$

$2^{O(n \log \log n / \log n)}$

∞

The Shortest Vector Problem (SVP)

- Input: a basis of a d-dim lattice L
- Output: nonzero v∈L minimizing ||v||. The minimal norm is ||L||.



2	0	0	0	0
0	2	0	0	0
0	0	2	0	0
0	0	0	2	0
1	1	1	1	1

The Algorithm of [Lenstra-Lenstra-Lovász1982]: LLL or L^3

- Given an integer lattice L of dim d, LLL finds in polynomial time a basis whose first vector satisfies:

$$\|\vec{b}_1\| \leq 2^{(d-1)/4}\mathrm{vol}(L)^{1/d} \qquad \|\vec{b}_1\| \leq 2^{(d-1)/2}\|L\|$$

- The constant 2 can be replaced by 4/3+ε.fand the running time becomes polynomial in 1/ε. This is reminiscent of Hermite's inequality:

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

The Magic of LLL

- One of the main reasons behind the popularity of LLL is that it performs "much better" than what the worst-case bounds suggest, especially in low dimension.
- This is another example of worst-case vs. "average-case".

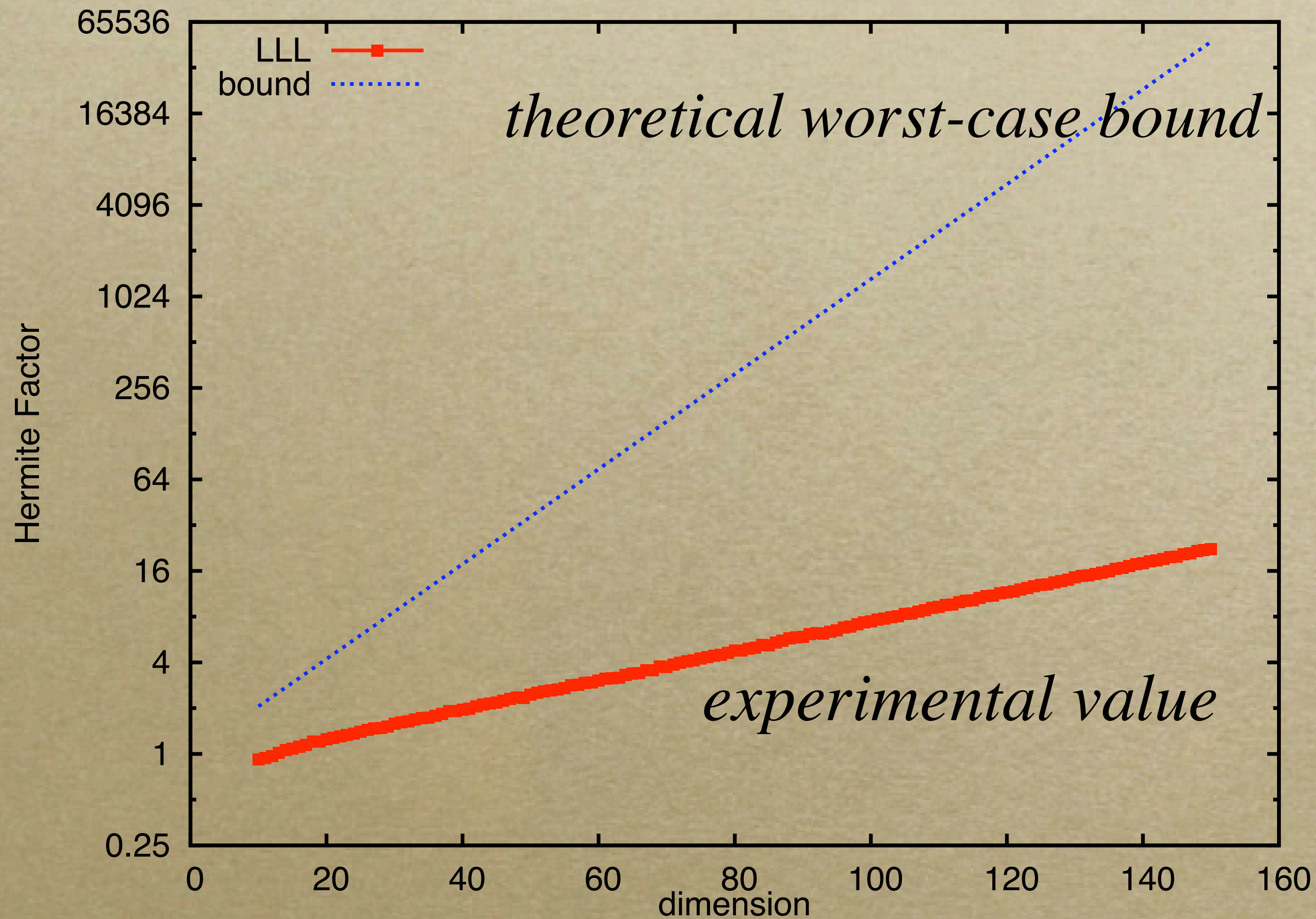
LLL: Theory vs Practice

- The approx factors $(4/3+\varepsilon)^{(d-1)/4}$ and $(4/3+\varepsilon)^{(d-1)/2}$ are tight in the worst case: but this is only for worst-case bases of certain lattices.
- Experimentally, $4/3+\varepsilon \approx 1.33$ can be replaced by a smaller constant ≈ 1.08, for any lattice, by randomizing the input basis.
- But there is no good explanation for this phenomenon, and no known formula for the experimental constant ≈ 1.08.

To summarize

- LLL performs better in practice than predicted by theory, but not that much better: the approximation factors remain exponential on the average and in the worst-case, except with smaller constants.
- Still no good explanation.

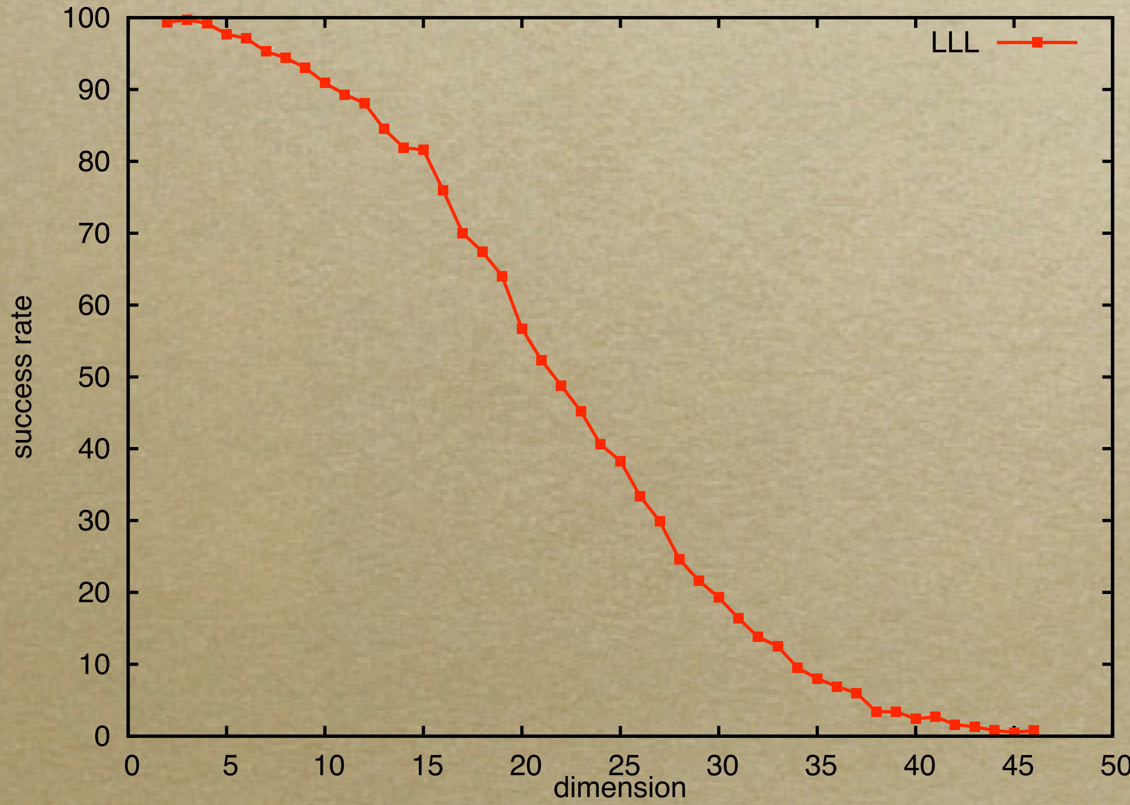
Illustration



Log(Hermite Factor)

Other unexplained phenomenon

- In small dimension, LLL behaves as a randomized exact SVP algorithm!



The Power of LLL

- LLL not only finds a “short” lattice vector, it finds a “short” lattice basis.

One Notion of Reduction: The Orthogonality Defect

- If $(b_1,\ldots,b_n)$ is a basis of L, then Hadamard's inequality says that:

$$\mathrm{vol}(L) \leq \prod_{i=1}^{d} \|\vec{b}_i\|$$

- Reciprocally, we may wish for a basis such that

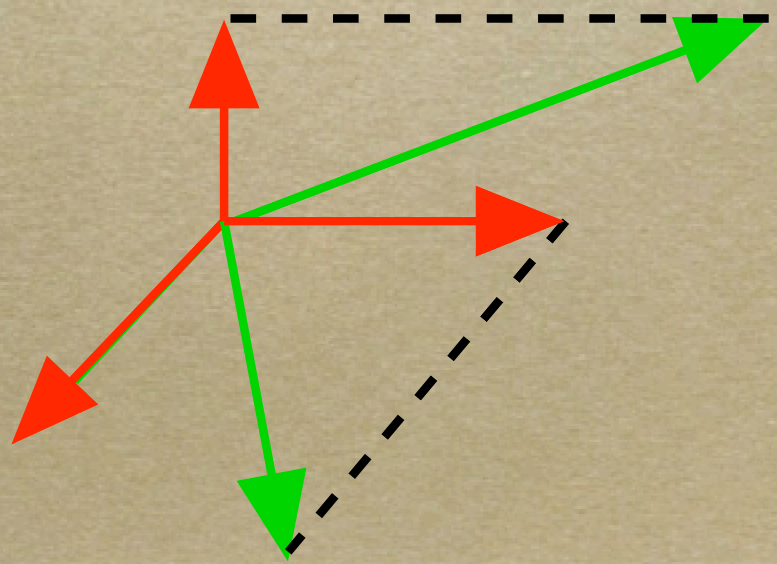
$$\prod_{i=1}^{d} \|\vec{b}_i\| \leq \mathrm{vol}(L) \cdot \mathrm{constant}$$

Triangularization from Gram-Schmidt

Gram-Schmidt

- From d linearly independent vectors, GS constructs d orthogonal vectors: the i-th vector is projected over the orthogonal complement of the first i-1 vectors.

$$\vec{b}_1^\star = \vec{b}_1$$
$$\vec{b}_i^\star = \vec{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \vec{b}_j^\star$$
$$\text{where } \mu_{i,j} = \frac{\langle \vec{b}_i, \vec{b}_j^\star \rangle}{\|\vec{b}_j^\star\|^2}$$



Gram-Schmidt and Volume

- For each k, $||b^*_k||$ is the distance of b_k to the subspace spanned by $b_1,\ldots,b_{(k-1)}$.
- If $b_1,\ldots,b_d$ is a basis of L, then:
  - $vol(L) = ||b^*_1|| \times ||b^*_2|| \times \ldots \times ||b^*_d||$

Computing Gram-Schmidt

- If $b_1,\ldots,b_d \in \mathbf{Z}^n$, then $b^*_1, b^*_2,\ldots,b^*_d \in \mathbf{Q}^n$.
- They can be computed in polynomial time from the recursive formula.
- Note:
  - The denominator of each b^*_i divides $(\|b^*_1\| \times \|b^*_2\| \times \ldots \times \|b^*_i\|)^2 = \mathrm{vol}(b_1,\ldots,b_i)^2$
  - The denominator of each $\mu_{i,j}$ divides $(\|b^*_1\| \times \|b^*_2\| \times \ldots \times \|b^*_j\|)^2 = \mathrm{vol}(b_1,\ldots,b_j)^2$

Gram-Schmidt = Triangularization

- If we take an appropriate orthonormal basis, the matrix of the lattice basis becomes triangular.

$$\begin{pmatrix} \|\vec{b}_1^*\| & 0 & 0 & \dots & 0 \\ \mu_{2,1}\|\vec{b}_1^*\| & \|\vec{b}_2^*\| & 0 & \dots & 0 \\ \mu_{3,1}\|\vec{b}_1^*\| & \mu_{3,2}\|\vec{b}_2^*\| & \|\vec{b}_3^*\| & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mu_{d,1}\|\vec{b}_1^*\| & \mu_{d,2}\|\vec{b}_2^*\| & \dots & \mu_{d,d-1}\|\vec{b}_{d-1}^*\| & \|\vec{b}_d^*\| \end{pmatrix}$$

Why Gram-Schmidt?

$$\mathrm{vol}(L) = \prod_{i=1}^{d} \|\vec{b}_i^\star\|$$

- If the Gram-Schmidt do not decrease too fast, then $\vec{b}_1 = \vec{b}_1^\star$ won't be too far from the d-th root of the volume.
- Neither from the first minimum because:

$$\lambda_1(L) \geq \min_i \|\vec{b}_i^\star\|$$

Two dimensions (1773)

Low Dimension

- If dim≤4, there exist bases reaching all the minima. Can we find them?
- Yes and as fast as Euclid!
  - Dim 2: Lagrange-Gauss, analysis by [Lagarias1980].
  - Dim 3: [Vallée1986-Semaev2001].
  - Dim 4: [N-Stehlé2004]

Reduction operations

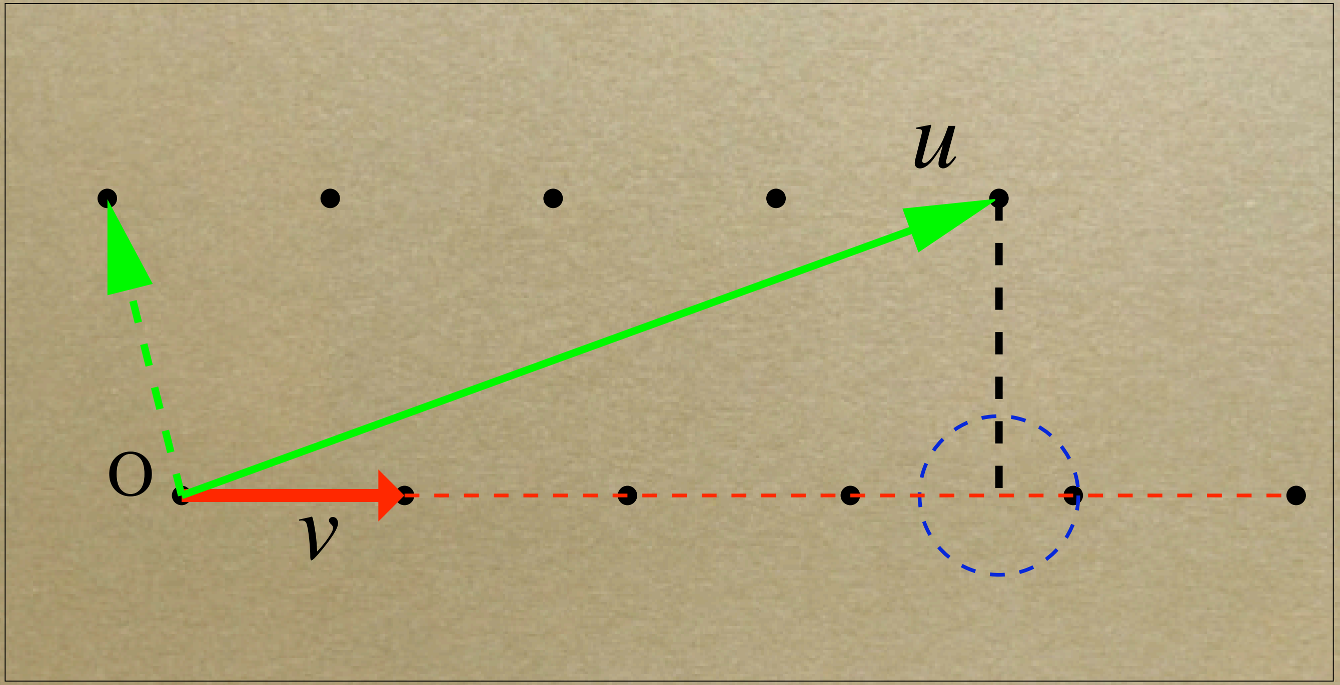
- To improve a basis, we may :
  - Swap two vectors.
  - Slide: subtract to a vector a linear combination of the others.
- That's exactly what Euclid's algorithm does.

Lagrange's Algorithm

- Input: a basis [u,v] of L
- Output: a basis of L whose first vector is a shortest vector.
- Assume that $||u|| \geq ||v||$
- Can we shorten u by subtracting a multiple of v?

The right slide

- Finding the best multiple amounts to finding a closest vector in the lattice spanned by v!
- The optimal choice is qv where q is the closest integer to $\langle u,v \rangle / ||v||^2$



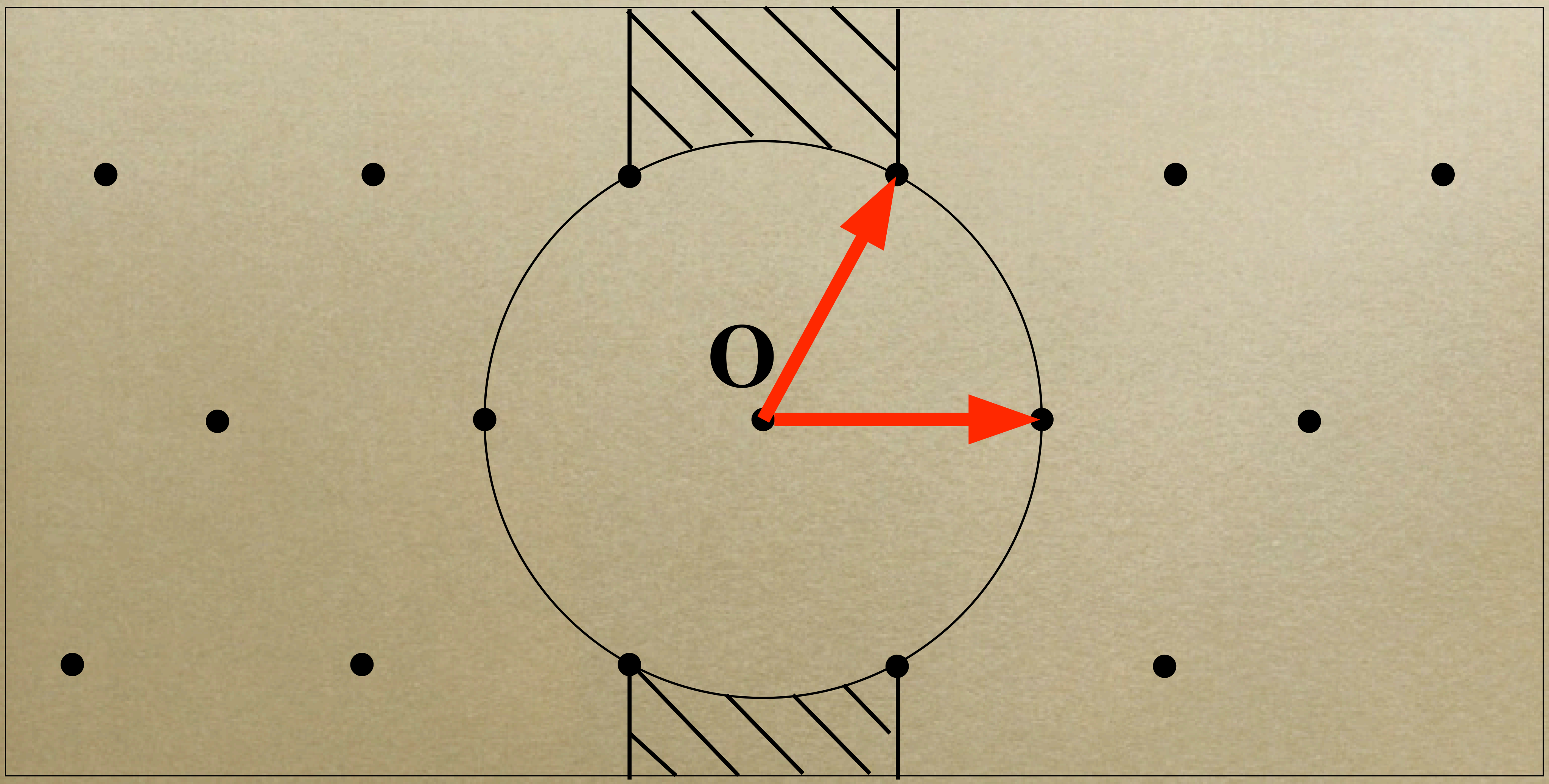
Lagrange’s Algorithm

- Repeat
  - Compute $r := qv$ where q is the closest integer to $\langle u,v \rangle / ||v||^2$.
  - $u := u-r$
  - Swap(u,v)
- Until $||u|| \leq ||v||$
- Output [u,v]

Lagrange's reduction

- A basis [u,v] is L-reduced iff
  - $\|u\| \leq \|v\|$
  - $|\langle u,v\rangle|/\|v\|^2 \leq 1/2$
- Such bases exist since Lagrange's algorithm clearly outputs L-reduced bases.

The 2-dimensional Case



$|\mu_{2,1}| \leq 1/2$ $\quad$ $\|\vec{b}_1^*\|^2/\|\vec{b}_2^*\|^2 \leq 4/3$ $\quad$ $\gamma_2 = (4/3)^{1/2}$

Exercises

- Show that if a basis [u,v] of L is Lagrange-reduced then:
  - $\|u\| = \lambda_1(L)$
- Show that Lagrange's algorithm is polynomial time, and even quadratic (in the maximal bit-length of the coefficients) like Euclid's algorithm. Hint: consider $\langle u,v \rangle$.

1773

1850

1982

The n-dimensional case:

From L to LLL

Bounding Hermite's Constant and Approximate SVP Algorithms

Bounding Hermite's Constant

- Early method to find Hermite's constant:
  - Find good upper bounds on Hermite's constant.
  - Show that the upper bound is also a lower bound, by exhibiting an appropriate lattice.
- This works up to dim 4.

Approximation Algorithms for SVP

- All related to historical methods to upper bound Hermite's constant.
- [LLL82] corresponds to [Hermite1850]'s inequality.

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

- [Schnorr87, GHKN06, GamaN08] correspond to [Mordell1944]'s inequality.

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

The Algorithm of [Lenstra-Lenstra-Lovász1982]: LLL or L^3

- Given an integer lattice L of dim d, LLL finds in polynomial time a basis whose first vector satisfies:

$$\|\vec{b}_1\| \leq 2^{(d-1)/4} \mathrm{vol}(L)^{1/d} \qquad \|\vec{b}_1\| \leq 2^{(d-1)/2} \|L\|$$

- It is often noted that the constant 2 can be replaced by 4/3+ε. This is reminiscent of Hermite's inequality:

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

The 2-dimensional Case

- By proving that $\gamma_2 \leq (4/3)^{1/2}$, we also described an algorithm to find the shortest vector in dimension 2. This algorithm is Lagrange's algorithm, also known as Gauss' algorithm.

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.
- Easy proof by induction: consider a shortest lattice vector, and project the lattice orthogonally...

Hermite's Reduction

- Hermite proved the existence of bases such that:

$$|\mu_{i,j}| \leq \frac{1}{2} \quad \text{and} \quad \frac{\|\vec{b}_i^\star\|^2}{\|\vec{b}_{i+1}^\star\|^2} \leq \frac{4}{3}$$

- Such bases approximate SVP to an exp factor:

$$\|\vec{b}_1\| \leq \left[(4/3)^{1/4}\right]^{d-1} \mathrm{vol}(L)^{1/d} \qquad \gamma_d \leq (4/3)^{(d-1)/2}$$

$$\|\vec{b}_i\| \leq \left[(4/3)^{1/2}\right]^{d-1} \lambda_i(L)$$

Computing Hermite reduction

- Hermite proved the existence of :

$$|\mu_{i,j}| \leq \frac{1}{2} \quad \text{and} \quad \frac{\|\vec{b}_i^\star\|^2}{\|\vec{b}_{i+1}^\star\|^2} \leq \frac{4}{3}$$

- By relaxing the 4/3, [LLL1982] obtained a provably polynomial-time algorithm.

The Algorithm of [Lenstra-Lenstra-Lovász1982] : LLL ou L^3

- Given an integer lattice of dim d, LLL finds a basis almost H-reduced in polynomial time $O(d^6B^3)$ where B is the maximal size of the norms of initial vectors.
- The running time is really cubic in B, because GS is computed exactly, which already costs $O(d^5B^2)$.

Note on the LLL bound

- In the worst case, we are limited by Hermite's constant in dimension 2, hence the 4/3 constant in the approximation factor.
- In practice however, the 4/3 seems to be replaced by a smaller constant, whose value can be observed empirically [N-St2006]. Roughly, $(4/3)^{1/4}$ is replaced by 1.02

LLL

- LLL tries to reduce all the 2x2 lattices.

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

Lenstra-Lenstra-Lovász

$$\vec{b}_i^\star = \vec{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \vec{b}_j^\star \qquad \text{where } \mu_{i,j} = \frac{\langle \vec{b}_i, \vec{b}_j^\star \rangle}{\|\vec{b}_j^\star\|^2}$$

- A basis is LLL-reduced if and only if
  - it is size-reduced $\qquad |\mu_{i,j}| \leq \frac{1}{2}$
  - Lovasz' conditions are satisfied

$$0.99 \|\vec{b}_{i-1}^\star\|^2 \leq \|\vec{b}_i^\star + \mu_{i,i-1} \vec{b}_{i-1}^\star\|^2$$

$$\text{Hence, roughly: } \|\vec{b}_{i-1}^\star\|^2 \leq \frac{4}{3} \|\vec{b}_i^\star\|^2$$

Description of the LLL Algorithm

- While the basis is not LLL-reduced
  - Size-reduce the basis
  - If Lovasz' condition does not hold for some pair (i-1,i): just swap b_{i-1} and b_i.

Size-reduction

- For $i = 2$ to d
  - For $j = i-1$ downto 1
    - Size-reduce b_i with respect to b_j: make $|\mu_{i,j}| \leq 1/2$ by $b_i := b_i - \mathrm{round}(\mu_{i,j}) b_j$
    - Update all $\mu_{i,j'}$ for $j' \leq j$.
- The translation does not affect the previous $\mu_{i',j'}$ where $i' < i$, or $i'=i$ and $j'>j$.

Why LLL is polynomial

- Consider the quantity $P = \prod_{i=1}^{d} \|\vec{b}_i^*\|^{2(d-i+1)}$
- If the b_i's have integral coordinates, then P is a positive integer.
  - Size-reduction does not modify P.
  - But each swap of LLL makes P decrease by a factor <= 1-ε
- This implies that the number of swaps is polynomially bounded.

Recap of LLL

- The LLL algorithm finds in polynomial time a basis such that:

$$|\mu_{i,j}| \leq \frac{1}{2} \qquad \text{and} \qquad \frac{\|\vec{b}_i^\star\|^2}{\|\vec{b}_{i+1}^\star\|^2} \leq \frac{4}{3} + \varepsilon$$

- Such bases approximate SVP to an exp factor:

$$\|\vec{b}_1\| \leq \left[(4/3+\varepsilon)^{1/4}\right]^{d-1} \mathrm{vol}(L)^{1/d}$$

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

$$\|\vec{b}_i\| \leq \left[(4/3+\varepsilon)^{1/2}\right]^{d-1} \lambda_i(L)$$

Implementing LLL

- We described a simple version of LLL, which is not optimized for implementation, for several reasons:
  - The use of rational arithmetic.
  - Size-reduction of a whole basis.

Simple Optimizations

- It is better to keep a counter k, which varies during the execution, and such that $b_1,\ldots,b_{(k-1)}$ are always LLL-reduced.
  - Initially, k=2.
  - At the end, k=d+1.
- We only need to size-reduce b_k and test Lovász' condition.

Other Optimizations

- We may rewrite LLL using only integer arithmetic, because we know good denominators for all the rational numbers.
- More tricky, but more efficient: we may replace rational arithmetic by floating-point arithmetic of suitable precision.

1982

Beyond LLL

Improving LLL

- Decreasing the running time: Faster LLLs.
- Improving the output quality: stronger LLLs.
  - Solving SVP exactly
  - Approximate SVP in polynomial time to within better factors

Faster LLL

- LLL runs in poly time $O(d^6 \log^3 B)$ without fast integer arithmetic.
- Improving "d": [Schönhage84,Schnorr88].
- But LLL generalizes Euclid's gcd algorithm, which is quadratic, not cubic. [N-Stehlé2005] found the first quadratic variant of LLL: $O(d^5 \log^2 B)$ without fast arithmetic.
- Is it possible to achieve quasi-linear time?

Applications of LLL: Exact SVP Algorithms

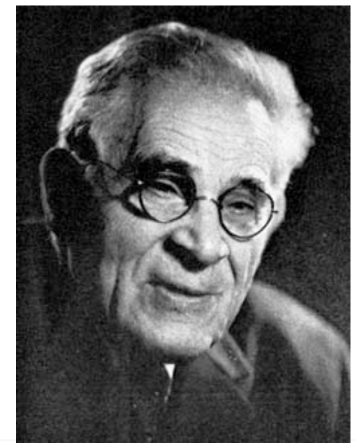
Exact SVP Algorithms

- Kannan (1983): deterministic super-exponential time $2^{O(d \ln d)}$ (and negligible space).
- Ajtai-Kumar-Sivakumar (2001): randomized exponential time $2^{O(d)}$ (but also exponential space). Not used in practice. Now also deterministic: [MV2010].

1850

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

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



1944

From Hermite to Mordell: Divide and Conquer

Applications of Exact Algorithms: Improving LLL in polynomial time

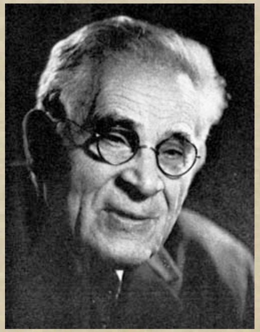
Divide and Conquer

- Consider a lattice L of dimension d.
- If we select a small k << d, we can find shortest vectors in lattices of dim k in time polynomial in d. For instance,
  $k = \log(d)/\log(\log(d))$ will do.
- Can we exploit such an oracle to improve the quality of LLL, provided that the number of calls is polynomial?

A Mathematical Analogue

- If we know Hermite's constant exactly in dim k, can we use that knowledge to upper bound Hermite's constant in dim d > k?

Mordell's Inequality

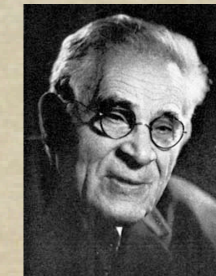


- Hermite's inequality is a particular case of Mordell's inequality:

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

- 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).

An Algorithmic Version of Mordell's Inequality



- Using a k-dim oracle, one "should" be able to solve Hermite-SVP with factor $\sqrt{\gamma_k}^{(d-1)/(k-1)}$
- This is achieved by the algorithm of [GamaN2008], which is to Mordell's inequality what LLL is to Hermite's inequality.
- By choosing an appropriate k=f(d), the whole algorithm is poly-time with a subexponential approx factor.

Schnorr's Algorithm (1987)

- Given an oracle which solves SVP up to dim 2k, Schnorr's algorithm finds a non-zero lattice vector of norm:

$$\leq O\left(\left(k^{\ln 2/(2k)}\right)^{d}\right) \mathrm{vol}(L)^{1/d}$$

See [Schnorr87,GHKN06]

From LLL to Block Reduction

- LLL tries to reduce all the 2x2 lattices.

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

Schnorr's Reduction (1987)

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

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

Gama-N's Algorithm

- Try to reduce all the disjoint k-dim lattices + all the "slided" dual k-dim lattices

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

Recap

- The best polynomial algorithms solve Hermite-SVP and Approx-SVP within a factor $(1+eps)^d$ which can be made slightly subexponential.
- Such algorithms might find the exact solution, depending on the properties of the lattice.
- The best exact algorithms are at least exponential, and are totally impractical if dim >= 130.

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. Ex: LLL and blockwise algorithms.

Speculation

- If all poly-time algorithms correspond to classical inequalities on Hermite's constant, do other methods for bounding Hermite's constant have algorithmic analogues?
  - Minkowski's Convex Body Theorem: it has a superexponential analogue based on Mordell's proof of Blichfeldt's lemma.
  - The method of [CohnElkies2003,CohnKumar2004].

CONCLUSION

Open problems

- Efficient algorithms to approximate SVP within a polynomial factor, possibly quantum.
- Other problems
  - Find a $2^{O(d)}$ SVP-algorithm not requiring exponential space.
  - Find an LLL with quasi-linear time.
  - Find a poly-time algorithm unrelated to Hermite's constant.

Bridging Theory and Practice

- The algorithms used in practice somewhat differ from the best theoretical algorithms.
- Assessing/understanding the "average-case" performances of lattice algorithms. What are the average-case constants?