MPRI 2-12-1: Phong Nguyen **Fall 2024**

Lecture 7: Hermite's Inequality and the LLL Algorithm

1 Hermite's inequality

Recall Hermite's constant:

$$\gamma_d = \max_{d-\mathrm{rank}L} \frac{\lambda_1(L)^2}{\mathrm{covol}(L)^{2/d}} \tag{1}$$

Lagrange proved that $\gamma_2 = \sqrt{4/3}$. Hermite proved the existence of γ_d by proving the following inequality:

Theorem 1.1 (Hermite's inequality [4]) *For all integer $d \geq 2$:*

$$\gamma_d \leq \gamma_2^{d-1}. \tag{2}$$

Proof. We give a proof by induction, slightly different from the historical proof of Hermite. Since the inequality is trivial for $d = 2$, assume that it holds for $d-1$. Consider a shortest nonzero vector $\mathbf{b}_1$ of a d-rank lattice L. Denote by $L' = \pi_2(L)$ the $(d-1)$-rank lattice obtained by projecting L over $\mathbf{b}_1^\perp$. Its volume is $\mathrm{vol}(L') = \mathrm{vol}(L)/\|\mathbf{b}_1\|$. Let $\mathbf{b}_2'$ be a shortest nonzero vector of L'. The induction assumption ensures that:

$$\|\mathbf{b}_2'\| \leq (4/3)^{(d-2)/4}\mathrm{vol}(L')^{1/(d-1)}.$$

We can lift $\mathbf{b}_2'$ (by size-reduction) into a nonzero vector $\mathbf{b}_2 \in L$ such that $\|\mathbf{b}_2\|^2 \leq \|\mathbf{b}_2'\|^2 + \|\mathbf{b}_1\|^2/4$. Since $\mathbf{b}_1$ cannot be longer than $\mathbf{b}_2$, we deduce:

$$\|\mathbf{b}_1\| \leq \sqrt{4/3}\|\mathbf{b}_2'\| \leq (4/3)^{d/4}\mathrm{vol}(L')^{1/(d-1)},$$

which can be rewritten as:

$$\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d},$$

which completes the proof. In retrospect, one notices that with the inequality $\|\mathbf{b}_1\| \leq \sqrt{4/3}\|\mathbf{b}_2'\|$, one has in fact proved the inequality:

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

By composing all these inequalities, one indeed obtains Hermite's inequality:

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

The historical proof given by Hermite in his first letter [4] to Jacobi also proceeded by induction, but in a slightly different way. Hermite considered an arbitrary primitive vector $\mathbf{b}_1$ of the lattice L. If $\mathbf{b}_1$ satisfies Hermite's inequality, that is, if $\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d}$, there is nothing to prove. Otherwise, one applies the induction assumption to the projected lattice $L' = \pi_2(L)$: one knows that there exists a primitive vector $\mathbf{b}_2^\star \in L'$ satisfying Hermite's inequality: $\|\mathbf{b}_2^\star\| \leq (4/3)^{(d-2)/4}\mathrm{vol}(L')^{1/(d-1)}$. One can lift this vector $\mathbf{b}_2^\star \in L'$ into a primitive vector $\mathbf{b}_2 \in L$ such that $\|\mathbf{b}_2\|^2 \leq \|\mathbf{b}_2^\star\|^2 + \|\mathbf{b}_1\|^2/4$. Since $\mathbf{b}_1$ does not satisfy Hermite's inequality, one notices that $\|\mathbf{b}_2\| < \|\mathbf{b}_1\|$: one can therefore replace $\mathbf{b}_1$ by $\mathbf{b}_2$, and start again. But this process cannot go on indefinitely: indeed, there are only finitely many vectors of L which have norm $\leq \|\mathbf{b}_1\|$. Hence, there must exist a nonzero vector $\mathbf{b}_1 \in L$ satisfying Hermite's inequality. □

The inequality (2) suggests to use two-dimensional reduction to find in any d-rank lattice a nonzero vector of norm less than:

$$\sqrt{\gamma_2^{d-1}}\mathrm{vol}(L)^{1/d} = (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d}.$$

This is somewhat the underlying idea behind all the algorithms of this section: Hermite's algorithms and the Lenstra-Lenstra-Lovász algorithm (LLL). In fact, the proof of (2) that we gave provides such an algorithm, implicitly.

To explain this algorithm, let us recall Gram-Schmidt orthogonalization. The *Gram-Schmidt orthogonalization* (GSO) of $\mathbf{b}_1, \ldots, \mathbf{b}_d$ is the orthogonal family $(\mathbf{b}_1^\star, \ldots, \mathbf{b}_d^\star)$ defined as follows: $\mathbf{b}_1^\star = \mathbf{b}_1$ and more generally $\mathbf{b}_i^\star = \pi_i(\mathbf{b}_i)$ for $1 \leq i \leq d$, where π_i denotes the orthogonal projection over the orthogonal supplement of the linear span of $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. Recall that:

$$\mathbf{b}_i^\star = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^\star, 1 \leq i \leq d \tag{3}$$

where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^\star \rangle}{\|\mathbf{b}_j^\star\|^2}$.

This algorithm makes sure that the basis is size-reduced and that all the local bases $(\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1})) = (\mathbf{b}_i^\star, \mathbf{b}_{i+1}^\star + \mu_{i+1,i}\mathbf{b}_i^\star)$ are L-reduced: these local bases correspond to the 2×2 matrices on the diagonal, when we represent the basis in triangular form. In other words, the reduced bases obtained

are size-reduced and such that for all $1 \leq i \leq d$:

$$\|\mathbf{b}_{i+1}^{\star}\|^2 \geq \frac{3}{4}\|\mathbf{b}_i^{\star}\|^2, \tag{4}$$

that is, the decrease of the norms of the Gram-Schmidt vectors (which are the diagonal coefficients in the triangular representation) is at most geometric, which is sometimes called Siegel's condition [10]. It is then easy to see that the first vector of such a basis satisfies:

$$\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d},$$

as announced. But it is unknown if this algorithm and those of Hermite are polynomial time: the LLL algorithm guarantees a polynomial running-time by relaxing inequalities (4).

2 Hermite's algorithms

We now describe the first reduction algorithms in arbitrary dimension, described by Hermite in his famous letters [4] to Jacobi, in the language of quadratic forms. They are very close to the algorithm underlying the proof of (2), but they do not explicitly rely on Lagrange's algorithm, although they try to generalize it. They were historically presented in a recursive way, but they can easily be made iterative, just like LLL. Hermite's first algorithm was described in the first letter [4] to Jacobi: Algorithm 1 is a simplified version of this algorithm; Hermite's historical algorithm actually uses duality, which we ignore for simplicity. It is easy to see that Algorithm 1 terminates, and that the output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ satisfies the following reduction notion (which we call H1):

- The basis is size-reduced.
- For all i, $\mathbf{b}_i^{\star}$ verifies Hermite's inequality in the projected lattice $\pi_i(L)$:

$$\|\mathbf{b}_i^{\star}\| \leq (4/3)^{(d-i)/4}\mathrm{vol}(\pi_i(L))^{1/(d-i+1)}$$

Surprisingly, Hermite notices himself that his first algorithm does not match with Lagrange's algorithm in dimension two. It seems to be one of the reasons why he presents a second algorithm (Algorithm 2) in his second letter [4] to Jacobi. It is easy to see that this algorithm terminates, and that the output basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ satisfies the following reduction notion (which we call H2):

Algorithm 1 A simplified version of Hermite's first reduction algorithm, described in the first letter to Jacobi [4].

Input: A basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a d-rank lattice L.
Output:

if $d = 1$ **then**
  output $\mathbf{b}_1$
end if
Apply recursively the algorithm to the basis $(\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_d))$ of the projected lattice $\pi_2(L)$.
Lift the vectors $(\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_d))$ into $\mathbf{b}_2, \ldots, \mathbf{b}_d \in L$ in such a way that they are size-reduced with respect to $\mathbf{b}_1$.
if $\mathbf{b}_1$ satisfies Hermite's inequality, that is $\|\mathbf{b}_1\| \leq (4/3)^{(d-1)/4}\mathrm{vol}(L)^{1/d}$ **then**
  Output $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$
end if
Swap $\mathbf{b}_1$ and $\mathbf{b}_2$ since $\|\mathbf{b}_2\| < \|\mathbf{b}_1\|$, and restart from the beginning.

Algorithm 2 Hermite's second reduction algorithm, described in his second letter to Jacobi [4].

Input: a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L.
Output: a size-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ such that for all i, $\|\mathbf{b}_i^\star\|/\|\mathbf{b}_{i+1}^\star\| \leq \gamma_2 = \sqrt{4/3}$. In particular, each $\mathbf{b}_i^\star$ satisfies Hermite's inequality in the projected lattice $\pi_i(L)$.

if $d = 1$ **then**
  output $\mathbf{b}_1$
end if
By making swaps if necessary, ensure that $\|\mathbf{b}_1\| \leq \|\mathbf{b}_i\|$ for all $i \geq 2$.
Apply recursively the algorithm to the basis $(\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_d))$ of the projected lattice $\pi_2(L)$.
Lift the vectors $(\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_d))$ to $\mathbf{b}_2, \ldots, \mathbf{b}_d \in L$ in such a way that they are size-reduced with respect to $\mathbf{b}_1$.
if $\|\mathbf{b}_1\| \leq \|\mathbf{b}_i\|$ for all $i \geq 2$ **then**
  output $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$
else
  restart from the beginning.
end if

- The basis is size-reduced.
- For all i, $\mathbf{b}_i^\star$ has minimal norm among all the vectors of the basis $(\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}) \ldots, \pi_i(\mathbf{b}_d))$ of the projected lattice $\pi_i(L)$, that is $\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$ for all $1 \leq i \leq j \leq d$.

Notice that an H2-reduced basis necessarily satisfies (4), that is for all i :

$$\|\mathbf{b}_i^\star\| / \|\mathbf{b}_{i+1}^\star\| \leq \gamma_2 = \sqrt{4/3}.$$

This implies that its orthogonality defect is bounded:

$$\prod_{i=1}^{d} \|\mathbf{b}_i^\star\| \leq (4/3)^{d(d-1)/4} \mathrm{vol}(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_d)).$$

And this also shows that an H2-reduced basis is necessarily H1-reduced. Yes, this second algorithm is still not known to be efficient.

3 The LLL algorithm

Surprisingly, it is unknown if Hermite's algorithms are polynomial time for varying dimension. It is also the case for Lenstra's algorithm [7], which is a relaxed variant of Hermite's second algorithm, where the inequalities $\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$ are replaced by $c\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$ where c is a constant such that $1/4 < c < 1$. However, Lenstra proved that his algorithm was polynomial time for any fixed dimension, which was sufficient for his celebrated result on integer programming [7].

It is Lenstra, Lenstra and Lovász [6] who invented in 1982 the first polynomial-time reduction algorithm outputting bases nearly as reduced as Hermite's. This algorithm, known as LLL or L^3, is essentially a relaxed variant of Hermite's second algorithm: László Lovász discovered that a crucial modification guaranteed a polynomial running-time; more precisely, compared to the H2 reduction notion, one replaces for each i all the inequalities $\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_j)\|$ by a single inequality $c\|\mathbf{b}_i^\star\| \leq \|\pi_i(\mathbf{b}_{i+1})\|$ where c is a constant such that $1/4 < c < 1$. The final algorithm was published in [6].

Let δ be a real in $]\frac{1}{4}, 1]$. A numbered basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L is said to be *LLL-reduced* with factor δ if it is size-reduced, and if it satisfies *Lovász' condition*: for all $1 < i \leq d$,

$$\left\|\mathbf{b}_{i+1}^\star + \mu_{i+1,i}\mathbf{b}_i^\star\right\|^2 \geq \delta \|\mathbf{b}_i^\star\|^2.$$

Let us explain this mysterious condition. Since Gram-Schmidt orthogonalization depends on the order of the vectors, its vectors change if $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$ are swapped; in fact, only $\mathbf{b}_i^\star$ and $\mathbf{b}_{i+1}^\star$ can possibly change. And the new $\mathbf{b}_i^\star$ is simply $\mathbf{b}_{i+1}^\star + \mu_{i+1,i}\mathbf{b}_i^\star$, therefore Lovász' condition means that by swapping $\mathbf{b}_i$ and $\mathbf{b}_{i+1}$, the norm of $\mathbf{b}_i^\star$ does not decrease too much, where the loss is quantified by δ: one cannot gain much on $\|\mathbf{b}_i^\star\|$ by swap. In other words:

$$\delta\|\mathbf{b}_i^\star\|^2 \leq \|\pi_i(\mathbf{b}_{i+1})\|^2,$$

which illustrates the link with the H2 reduction notion. The most natural value for the constant δ is therefore $\delta = 1$ (in dimension 2, this matches with Lagrange's reduction), but then, it is unknown if such a reduced basis can be computed in polynomial time. The LLL-reduction was initially[1] presented in [6] with the factor $\delta = \frac{3}{4}$, so that in the literature, LLL-reduction usually means LLL-reduction with the factor $\delta = \frac{3}{4}$.

Lovász' condition can also be rewritten equivalently: for all i,

$$\|\mathbf{b}_{i+1}^\star\|^2 \geq \left(\delta - \mu_{i+1,i}^2\right)\|\mathbf{b}_i^\star\|^2,$$

which is a relaxation of (4). Thus, LLL reduction guarantees that each $\mathbf{b}_{i+1}^\star$ cannot be much shorter than $\mathbf{b}_i^\star$: the decrease is at most geometric. This proves the following result:

Theorem 3.1 *Assume that $\frac{1}{4} < \delta \leq 1$, and let $\alpha = 1/(\delta - \frac{1}{4})$. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ be an LLL-reduced basis with factor δ of a lattice L in $\mathbb{R}^n$. Then:*

1. $\|\mathbf{b}_1\| \leq \alpha^{(d-1)/4}(\mathrm{vol}L)^{1/d}$.
2. *For all $i \in \{1, \ldots, d\}$:* $\|\mathbf{b}_i\| \leq \alpha^{(d-1)/2}\lambda_i(L)$.
3. $\|\mathbf{b}_1\| \times \cdots \times \|\mathbf{b}_d\| \leq \alpha^{d(d-1)/4}\det L$.

Proof. We know that:

$$\|\mathbf{b}_{i+1}^\star\|^2 \geq \left(\delta - \mu_{i+1,i}^2\right)\|\mathbf{b}_i^\star\|^2 \geq \alpha\|\mathbf{b}_i^\star\|^2.$$

Thus by iteration:

$$\|\mathbf{b}_i^\star\|^2 \geq \alpha^{i-1}\ \|\mathbf{b}_1\|^2.$$

Therefore:

$$\mathrm{vol}L = \prod_{i=1}^{d}\|\mathbf{b}_i^\star\| \geq \|\mathbf{b}_1\|^d\alpha^{1+2+\cdots+(d-1)},$$

[1] This simplifies the exposition.

so

$$\|\mathbf{b}_1\| \leq \alpha^{(d-1)/4}(\mathrm{vol}L)^{1/d}.$$

For the second and third items, show that $\lambda_i(L) \geq \min_{i \leq j \leq d} \|\mathbf{b}_j^\star\|$ and that for all $j \geq i$: $\|\mathbf{b}_i\|^2 \leq \alpha^{j-1} \|\mathbf{b}_j^\star\|^2$. □

Thus, an LLL-reduced basis provides an approximation of the lattice reduction problem. By taking δ very close to 1, one falls back on Hermite's inequality in an approximate way, where the constant $4/3$ is replaced by $4/3 + \varepsilon$.

The other interest of this reduction notion is that there exists a simple algorithm to compute such reduced bases, and which is rather close to Hermite's second algorithm (Algorithm 2). In its simplest form, the LLL algorithm corresponds to Algorithm 3. Compared to this simple version, the

Algorithm 3 The basic LLL algorithm.

Input: a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of a lattice L.
Output: the basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ is LLL-reduced with factor δ.

1: Size-reduce $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ (using Algorithm **??**).
2: **if** there exists an index j which does not satisfy Lovász' condition **then**
3: swap $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$, then return to Step 1.
4: **end if**

so-called iterative versions of the LLL algorithm consider instead the smallest index j not satisfying Lovász' condition: in contrast, Hermite's second algorithm considered the greatest index j refuting H2.

Theorem 3.2 *Assume that $\frac{1}{4} < \delta < 1$. If each $\mathbf{b}_i \in \mathbb{Q}^n$, Algorithm 3 computes an LLL-reduced basis in time polynomial in the maximal bit-length of the coefficients of the $\mathbf{b}_i$'s, the lattice rank d, and the space dimension n.*

Let us sketch a proof of this fundamental result, assuming to simplify that $\mathbf{b}_i \in \mathbb{Z}^n$. First of all, it is clear that if the algorithm terminates, then the output basis is LLL-reduced with factor δ. To see why the algorithm terminates, let us analyze each swap (Step 3). When $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$ are swapped, only $\mathbf{b}_j^\star$ and $\mathbf{b}_{j+1}^\star$ can be modified among all the Gram-Schmidt vectors. Let us therefore denote by $\mathbf{c}_j^\star$ and $\mathbf{c}_{j+1}^\star$ the new Gram-Schmidt vectors after swapping. Since the product of all the Gram-Schmidt vector norms is equal to $\mathrm{vol}(L)$, we have:

$$\|\mathbf{c}_j^\star\| \times \|\mathbf{c}_{j+1}^\star\| = \|\mathbf{b}_j^\star\| \times \|\mathbf{b}_{j+1}^\star\|.$$

Since Lovász' condition is not satisfied: $\|\mathbf{c}_j^\star\|^2 < \delta\|\mathbf{b}_j^\star\|^2$. Hence:

$$\|\mathbf{c}_j^\star\|^{2(d-j+1)}\|\mathbf{c}_{j+1}^\star\|^{2(d-j)} < \delta\|\mathbf{b}_j^\star\|^{2(d-j+1)}\|\mathbf{b}_{j+1}^\star\|^{2(d-j)}.$$

This suggests to consider the following quantity:

$$D = \|\mathbf{b}_1^\star\|^{2d}\|\mathbf{b}_2^\star\|^{2(d-1)} \times \cdots \times \|\mathbf{b}_d^\star\|^2.$$

At each swap, D decreases by a factor $\delta < 1$. Notice that D can be decomposed as a product of d Gram determinants $D_i = \Delta(\mathbf{b}_1, \ldots, \mathbf{b}_i)$ for i going through 1 to d. Therefore, D is in fact an integer, since $\mathbf{b}_i \in \mathbb{Z}^n$. It follows that the number of swaps is at most logarithmic in the initial value of D, which can be upper bounded by B^{2d} where B is the maximum of the initial norms $\|\mathbf{b}_i\|$. To bound the complexity of the algorithm, one also needs to upper bound the size of the rational coefficients $\mu_{i,j}$ and $\|\mathbf{b}_i^\star\|^2$ during the reduction. A careful analysis based on the D_i's shows that all the $\mu_{i,j}$'s always have polynomial size (see [6, 8, 1, 3]). Crucially, our analysis of swaps can actually show that during the execution of LLL, $\min_i \|\mathbf{b}_i^\star\|$ never decreases, and $\max_i \|\mathbf{b}_i^\star\|$ never increases.

By coupling Th. 3.1 with Th. 3.2, we can summarize the LLL result as follows:

Corollary 3.3 *There exists an algorithm which, given as input a basis of a d-dimensional integer lattice $L \subseteq \mathbb{Z}^n$ and a reduction factor $\varepsilon > 0$, outputs a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ of L, in time polynomial in $1/\varepsilon$ and the size of the basis, such that:*

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \leq \left((1+\varepsilon)\sqrt{4/3}\right)^{(d-1)/2}$$
$$\|\mathbf{b}_i\|/\lambda_i(L) \leq \left((1+\varepsilon)\sqrt{4/3}\right)^{d-1}, \ \ 1 \leq i \leq d$$
$$\left(\prod_{i=1}^{d}\|\mathbf{b}_i\|\right)/\mathrm{vol}(L) \leq \left((1+\varepsilon)\sqrt{4/3}\right)^{d(d-1)/2}$$

4 LLL in practice

The LLL bound $\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \leq \left((1+\varepsilon)\sqrt{4/3}\right)^{(d-1)/2}$ is tight in the worst case: one notes that there is a lattice L and an LLL-reduced basis of L whose first vector $\mathbf{b}_1$ satisfies:

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} = \left((1+\varepsilon)\sqrt{4/3}\right)^{(d-1)/2}.$$

However, it has been observed since the discovery of LLL that LLL returns better vectors in practice, which has made LLL very useful: see [9] for a detailed description of this phenomenon. Roughly speaking, in practice, given any lattice L of sufficiently high dimension, by running LLL on a suitably random input basis, one can experimentally obtain a non-zero lattice vector $\mathbf{b}_1$ such that

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \leq (1.02\dots)^d.$$

Depending on the structure of L, $\mathbf{b}_1$ might be shorter, but never worse.

It is an longstanding open problem to prove such a phenomenon. For instance, one would like to prove the existence of a constant $\alpha < (4/3)^{1/4}$ such that given as input the Hermite normal form of a random integer lattice L (for a suitable distribution), the first vector returned by LLL satisfies with overwhelming probability

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \leq \alpha^{d-1}.$$

And finding the least possible value of α would be very interesting.

Surprisingly, Kim and Venkatesh [5] showed that in a random lattice, most LLL bases have a first vector close to the worst case:

$$\|\mathbf{b}_1\|/\mathrm{vol}(L)^{1/d} \approx \left((1+\varepsilon)\sqrt{4/3}\right)^{(d-1)/2}.$$

This means that the LLL algorithm does not output a random LLL-reduced basis: its output distribution is significantly biased.

However, a model explaining the average-case behaviour of LLL has been proposed recently by [2], based on sandpiles: it is not a proof that LLL behaves better than its worst-case bounds, but the model seems to match very well the typical shape of the bases output by LLL.

References

[1] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1995. Second edition.

[2] Jintai Ding, Seungki Kim, Tsuyoshi Takagi, Yuntao Wang, and Bo-yin Yang. A physical study of the LLL algorithm. *J. Number Theory*, 244:339–368, 2023.

[3] C. Dwork. *Lattices and Their Application to Cryptography*. Stanford University, 1998. Lecture Notes, Spring Quarter. Several chapters are

translations of Claus Schnorr's 1994 lecture notes *Gittertheorie und algorithmische Geometrie, Reduktion von Gitterbasen und Polynomidealen.*

[4] C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre. *J. Reine Angew. Math.*, 40:279–290, 1850. Also available in the first volume of Hermite's complete works, published by Gauthier-Villars.

[5] Seungki Kim and Akshay Venkatesh. The behavior of random reduced bases. *Int. Math. Res. Not. IMRN*, (20):6442–6480, 2018.

[6] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.

[7] H. W. Lenstra, Jr. Integer programming with a fixed number of variables. Technical report, Mathematisch Instituut, Universiteit van Amsterdam, April 1981. Report 81-03.

[8] L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50. SIAM Publications, 1986. CBMS-NSF Regional Conference Series in Applied Mathematics.

[9] P. Q. Nguyen and D. Stehlé. LLL on the average. In *Proc. of ANTS-VII*, volume 4076 of *LNCS*. Springer-Verlag, 2006.

[10] C. L. Siegel. *Lectures on the Geometry of Numbers.* Springer-Verlag, 1989.