Lecture 5

Lecturers: L. Ducas
T.A.: S. Etinski

LLL Reduction

1 Introduction

The issue with Hermite's algorithm is that it may be terribly slow: there can be as many as $(I/\lambda_1(L))^n$ many intermediate values for $\mathbf{b}_1$, where I is the initial length $\|\mathbf{b}_1\|$. It can therefore be as large as exponential in the input size. This is without taking into account that there may be many more steps happening where $\mathbf{b}_1$ does not vary.

The idea of the Lenstra-Lenstra-Lovász (LLL) algorithm is to slightly relax the termination condition to make the algorithm (much!) faster. First, we present a weak version of the notion of an LLL-reduced basis, which is sufficient to show that the number of iterations in the algorithm is polynomial. However, the size of the numbers occurring in this computation are not necessarily bounded; it might still be possible that the numbers involved in the computation are too big to efficiently compute with. In the next section we present the true LLL algorithm, which resolves this problem.

2 The LLL algorithm

Let us start by definining the targeted reduction notion.

Definition 1 (ε-Weakly LLL reduced) *Let γ_2 be the hermite constant over 2-dimensional full rank lattice.* $\mathbf{B}$ *is then said to be ε-WLLL reduced if*

$$\|\mathbf{b}_i^\star\|_2 \leq (\gamma_2 + \varepsilon)\|\mathbf{b}_{i+1}^\star\|_2 \quad \textit{for all } i.$$

For a fixed i, this is known as the Lovàsz condition *on* $(\mathbf{b}_i^\star, \mathbf{b}_{i+1}^\star)$.

Algorithm 1: ε-Weakly LLL-reduction algorithm

```
Input  : A positive real $\varepsilon > 0$. A basis $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n) \in \mathbb{Q}^{n\times n}$ of a lattice $L$.
Output: An $\varepsilon$-WLLL reduced basis $\mathbf{B}$.
while $\exists i$ such that $\|\mathbf{b}_i^\star\|_2 > (\gamma_2+\varepsilon)\|\mathbf{b}_{i+1}^\star\|_2$ do
    Find matrix $\mathbf{U} \in \mathbb{Z}^{2\times 2}$ such that $\mathbf{B}_{i:i+1}\mathbf{U}$ is Lagrange reduced
    $(\mathbf{b}_i', \mathbf{b}_{i+1}') \leftarrow (\mathbf{b}_i, \mathbf{b}_{i+1})\mathbf{U}$
    $\mathbf{B} \leftarrow (\mathbf{b}_1, \dots, \mathbf{b}_{i-1}, \mathbf{b}_i', \mathbf{b}_{i+1}', \mathbf{b}_{i+2}, \dots, \mathbf{b}_n)$
end
return $\mathbf{B}$
```

Note that the above algorithm only spends time on doing a Lagrange reduction whenever the norm of $\mathbf{b}_i^\star$ is significantly larger than what it could be. This 'significant amount' is decided by $\varepsilon > 0$. So, in fact, the above algorithm is exactly the Hermite reduction algorithm, but with the difference that it allows some 'slack' in the form of the parameter ε.

Theorem 2 *If $\mathbf{B} \in \mathbb{Z}^{n\times n}$, then Algorithm 1 terminates after poly$(n, \log\|\mathbf{B}\|_\infty, 1/\varepsilon)$ iterations.*

Proof: Define $P(\mathbf{B}) = \prod_{i=1}^{n} \det(L_{1:i}) = \prod_{i=1}^{n} \|\mathbf{b}_i^\star\|_2^{n-i+1}$, to be the *potential*[1] of the current basis $\mathbf{B}$ where $\pi_1(\cdot)$ is the projection given as in Gram-Schmidt Orthogonalization and $L_{1:i} = \mathcal{L}(\pi_1(\mathbf{b}_1), \dots, \pi_1(\mathbf{b}_i))$ (note that the notation $L_{1:i}$ hides a dependence in the choice of the basis). We will show that $P(\mathbf{B})$ decreases by a constant factor in each iteration and that $P(\mathbf{B})$ has a lower bound. This combined shows that the algorithm terminates within a polynomially bounded number of iterations.

A single instance of Lagrange on $\mathbf{B}_{i:i+1}$ in an iteration only changes $\det(L_{1:i})$ and keeps all other determinants in the product of the potential fixed. It also maintains that $\mathbf{B} \in \mathbb{Z}^{n\times n}$. Before this iteration $\|\mathbf{b}_i^\star\|_2^2 \geq (\gamma_2 + \varepsilon)\det(L_{i:i+1})$. After the Lagrange reduction we must have $\|\mathbf{b}_i^\star\|_2^2 \leq \gamma_2 \det(L_{i:i+1})$. Therefore, after this Lagrange reduction, $\det(L_{1:i}) = \prod_{k=1}^{i} \|\mathbf{b}_k^\star\|_2$ must at least be diminished by a factor $\sqrt{\frac{\gamma_2}{\gamma_2+\varepsilon}}$. Thus, also the potential $P(\mathbf{B})$ must be reduced by (at least) this factor $f = \sqrt{\frac{\gamma_2}{\gamma_2+\varepsilon}}$.

As $L_{1:i} \subseteq \mathbb{Z}^n$, we must have that $\det(L_{1:i}) \in \mathbb{Z}\backslash\{0\}$. This has as a direct consequence that at any time in the algorithm, $P(\mathbf{B})$ must be larger than or equal to one.

The initial potential $P(\mathbf{B})_{\text{init}}$ (that is, the potential before running the algorithm) is bounded by $\prod_{i=1}^{n} \|\mathbf{b}_i\|_2^{n-i+1} \leq \|\mathbf{B}\|_\infty^{n(n+1)/2}$, where $\|\mathbf{B}\|_\infty := \max_i \|\mathbf{b}_i\|_2$. After N iterations, we know that

$$1 \leq P(\mathbf{B}) < f^N \cdot P(\mathbf{B})_{\text{init}} \leq f^N \cdot \|\mathbf{B}\|_\infty^{n(n+1)/2}$$

This means that the algorithm is surely terminated whenever $f^N \cdot \|\mathbf{B}\|_\infty^{n(n+1)/2} \leq 1$. Equivalently, it is ended when

$$N \geq \frac{n(n+1)\log(\|\mathbf{B}\|_\infty)}{-2 \cdot \log(f)}$$

So, the number of iterations is bounded by $\frac{n(n+1)\log(\|\mathbf{B}\|_\infty)}{-2\cdot\log(f)}$. Using the fact that $-\log(f) = 1/2 \cdot \log(1 + \varepsilon/\gamma_2) = \frac{\varepsilon}{2\gamma_2} + O(\varepsilon^2)$ yields the result. $\square$

Polynomial time yet? However, it is not enough to prove that the number of iterations is bounded, in order to show that Algorithm 1 is truly poly-time. Once again, the time spent computing with the (rational) numbers in the algorithm should also be taken into consideration; without more analysis we don't know whether the rational numbers occurring in $\mathbf{B}^\star$ have very large numerators and denominators (which might slow down the overall computation drastically). Also, we don't know whether the size of the integer coefficients of $\mathbf{B}$ can be bounded in any way. We will see that bounding the integer coefficients requires a modification of the algorithm, namely the 'true' LLL algorithm.

Lemma 3 below shows that the denominators of the entries in $\mathbf{B}^\star$ are bounded by $\det(\mathcal{L}(\mathbf{B}))^2$, so we are left only with the task of showing that the numerators are also not too large.

Lemma 3 *If* $\mathbf{B} \in \mathbb{Z}^{n\times m}$ *with* $n \geq m$ *has dimension* m*, then* $\mathbf{B}^\star \in \frac{1}{\det(\mathbf{B}^T\mathbf{B})}\mathbb{Z}^{n\times m}$.

Proof: We prove this by induction on the number of vectors in our basis (note that the length of the vectors remains the same). Let $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ be a basis. If $n = 1$, then $\mathbf{B}^\star$ has integer

[1]The word 'potential' is used in the literature, possibly to draw similarities with the 'potential' in classical mechanics, where an object has potential energy at the top of a hill which decreases as it rolls down the hill. In particular, it does not roll up the hill. The idea here is that as the algorithm progresses, there will be less and less *potential* in the basis, as if it were rolling down a hill towards an LLL reduced basis at the bottom.

coefficients and we are done. Write $\mathbf{C} = (\pi_2(\mathbf{b}_2), \ldots, \pi_2(\mathbf{b}_n))$, a projected basis. We make the inductive hypothesis that the statement holds for integer bases of dimension $n-1$. In order to apply this induction hypothesis on $\mathbf{C}$, we require that $\mathbf{C}$ is in $\mathbb{Z}^{n\times(n-1)}$. In general this is not true but we know that for any $\mathbf{v} \in \mathbb{Z}^n$, the projection $\pi_2(\mathbf{v}) = \mathbf{v} - \frac{\langle \mathbf{v}, \mathbf{b}_1\rangle}{\|\mathbf{b}_1\|_2^2} \cdot \mathbf{b}_1 \in \frac{1}{\|\mathbf{b}_1\|_2^2}\mathbb{Z}^n$. So we can scale up by the integer $d := \|\mathbf{b}_1\|^2$ so that $d\mathbf{C} \in \mathbb{Z}^{n\times(n-1)}$, and we can apply the induction hypothesis:

$$(d\mathbf{C})^\star = d\mathbf{C}^\star \in \frac{1}{\det(\mathbf{C}^T\mathbf{C})}\mathbb{Z}^{n\times(n-1)} \Rightarrow \mathbf{C}^\star \in \frac{1}{d\det(\mathbf{C}^T\mathbf{C})}\mathbb{Z}^{n\times(n-1)}$$

Now use that $\mathbf{C}$ has rank $n-1$, and therefore $\det(\mathbf{C}^T\mathbf{C}) = \prod_{i=2}^n \|\mathbf{b}_i^\star\|_2^2$. Every basis vector of $\mathbf{C}$ is orthogonal to $\mathbf{b}_1$, so $\mathbf{B}^\star = (\mathbf{b}_1|\mathbf{C}^\star) \in \frac{1}{d\det(C^TC)}\mathbb{Z}^{n\times m}$. Use $d\det(\mathbf{C}^T\mathbf{C}) = \|\mathbf{b}_1\|^2\det(\mathbf{C}^T\mathbf{C}) = \det(\mathbf{B}^T\mathbf{B})$ and we are done. □

2.1 Size-Reduction

Definition 4 *A basis of a lattice* $\mathbf{B}$ *is said to be* size reduced *when*

$$\forall i < j \quad \left|\langle \mathbf{b}_i^\star, \mathbf{b}_j\rangle\right| \le \frac{1}{2}\|\mathbf{b}_i^\star\|_2^2$$

The above definition is equivalent to saying that the off-diagonal of the upper triangular matrix of the Gram-Schmidt orthogonalization is bounded by 1/2. This notion of reduction allows us to estimate the size of the numerators of a basis. We will want to apply this reduction between every iteration of LLL to control the growth of coefficient, but at the same time, we do not wish to "disturb" the work of LLL on the potential: we want the size reduction to only affect $\mathbf{B}$, and not $\mathbf{B}^\star$.

Algorithm 2: SizeRed($\mathbf{B}$): Size-reduction algorithm

```
Input  : A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{Q}^{n\times m}$ for $m \le n$ of a lattice $L$.
Output: A size-reduced basis $\mathbf{C}$ of the same lattice $L$.
if $n = 1$ then
    return B
end
$\mathbf{B}' \leftarrow (\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$
$\mathbf{C}' \leftarrow$ SizeRed($\mathbf{B}'$)
$\mathbf{v} \leftarrow$ NearestPlane($\mathbf{C}', \pi_{\mathbf{C}'}(\mathbf{b}_n)$)
Return $(\mathbf{C}'|\mathbf{b}_n - \mathbf{v})$
```

Lemma 5 *Algorithm 2 is correct and terminates in polynomial time.*

Proof: Termination in polynomial time is left to the reader. For correctness, we proceed by induction on the rank n of the basis; the statement is vacuously true for $n = 1$.

We start by proving that the output $\mathbf{C}$ is indeed a basis of the same lattice as $\mathbf{B}$. Let us assume by induction that $\mathbf{C}'$ is a basis of the same lattice as $\mathbf{B}'$; that is, $\mathbf{C}' = \mathbf{B}' \cdot \mathbf{U}'$ for some unimodular matrix $\mathbf{U}' \in \mathrm{GL}_n(\mathbb{Z})$. By construction, $\mathbf{v}$ is in the lattice generated by $\mathbf{C}'$, so $\mathbf{v} = \mathbf{C}' \cdot \mathbf{x}$ for some

$\mathbf{x} \in \mathbb{Z}^{n-1}$. One chan check that the returned basis writes as $\mathbf{C} = \left(\begin{smallmatrix} \mathbf{U} & 0 \\ \mathbf{0} & 1 \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} \mathbf{I} & 0 \\ -\mathbf{x} & 1 \end{smallmatrix}\right) \mathbf{B}$. Note that $\left(\begin{smallmatrix} \mathbf{U} & 0 \\ \mathbf{0} & 1 \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} \mathbf{I} & 0 \\ -\mathbf{x} & 1 \end{smallmatrix}\right)$ is integer and has determinant 1, it is therefore a unimodular transformation: $\mathbf{C}$ is indeed a basis of L.

Now, regarding the size-reduction property, the induction hypothesis gives $|\langle \mathbf{c}_i^\star, \mathbf{c}_j \rangle| \leq \frac{1}{2}\|\mathbf{c}_i^\star\|_2^2$ for all $i < j < n$, and it remains to prove it for $i < j = n$, where by construction we have $\mathbf{c}_n = \mathbf{b}_n - \mathbf{v}$. By the specification of the NearestPlane algorithm, we have that $|\langle \pi_{\mathbf{C}'}(\mathbf{b}_n) - \mathbf{v}, \mathbf{c}_i^\star \rangle| \leq \frac{1}{2}\|\mathbf{c}_i^\star\|_2^2$ for all i. Furthermore, $\mathbf{c}_i^\star$ belongs to the span of $\mathbf{C}'$, so $\langle \pi_{\mathbf{C}'}(\mathbf{b}_n), \mathbf{c}_i^\star \rangle = \langle \mathbf{b}_n, \mathbf{c}_i^\star \rangle$. □

As mentioned above, an important fact is that size-reducing a basis does not affect its Gram-Schmidt basis and therefore it does not affect the potential.

Lemma 6 *Let $\mathbf{B} \in \mathbb{Q}^{n\times m}$, where $m \leq n$, and let $\mathbf{C} = \mathit{SizeRed}(\mathbf{B})$. Then, $\mathbf{C}^\star = \mathbf{B}^\star$. Furthermore $P(\mathbf{C}) = P(\mathbf{B})$.*

Proof: We proceed again by induction on the dimension n of the basis. When $n = 1$, we have $\mathbf{c}_1^\star = \mathbf{b}_1^\star$. Now assume that the statement is true for bases with $n-1$ vectors. Let $\mathbf{B}'$, $\mathbf{C}'$, $\mathbf{v}$ be as in Algorithm 2. In particular $\mathbf{B}'^\star = \mathbf{C}'^\star$, and therefore $\pi_{\mathbf{B}'}^\perp = \pi_{\mathbf{C}'}^\perp$. We are finished when we show that

$$\mathbf{c}_n^\star := \pi_{\mathbf{C}'}^\perp(\mathbf{c}) = \mathbf{b}_n^\star.$$

By construction, $\mathbf{v}$ belongs to the lattice with basis $\mathbf{C}'$. In particular $\pi_{\mathbf{C}'}^\perp(\mathbf{v}) = \mathbf{0}$. Hence

$$\mathbf{c}_n^\star = \pi_{\mathbf{C}'}^\perp(\mathbf{c}_n) = \pi_{\mathbf{C}'}^\perp(\mathbf{b}_n - \mathbf{v}) = \pi_{\mathbf{C}'}^\perp(\mathbf{b}_n) = \pi_{\mathbf{B}'}^\perp(\mathbf{b}_n) = \mathbf{b}_n^\star.$$

□

Finally, we state a bound on the size of the vectors and Gram-Schmidt vector of a size-reduced basis as a function of its potential. The following Lemma is stated for integer lattices, but it can be applied to rational lattices by simply scaling the lattice up by the least common multiple of the numerators appearing in the input basis.

Lemma 7 *A size-reduced basis $\mathbf{B}$ of an* integer *lattice $L \subseteq \mathbb{Z}^n$ has basis vectors $\mathbf{b}_i$ satisfying*

$$\log(\|\mathbf{b}_i\|), \log(\|\mathbf{b}_i^\star\|_2) \leq \mathit{poly}(n) \cdot \log(P(\mathbf{B})),$$

where $P(\mathbf{B}) = \prod_{i=1}^n \det(L_{1:i}) = \prod_{i=1}^n \|\mathbf{b}_i^\star\|_2^{n-i+1}$ denotes the potential of the basis $\mathbf{B}$ as in the proof of Theorem 2.

Proof: As $\mathbf{B} \in \mathbb{Z}^{n\times n}$, $\det(L_{1:i}) \geq 1$ for all $i \leq n$, and therefore $\det(L_{1:i}) \leq P(\mathbf{B})$. Writing $\|\mathbf{b}_i^\star\|_2 = \det(L_{1:i})/\det(L_{1:i-1})$ directly leads to $\|\mathbf{b}_i^\star\|_2 \leq P(\mathbf{B})$. As the basis is size-reduced, we can write any basis vector $\mathbf{b}_j = \mathbf{b}_j^\star + \sum_{i=1}^{j-1} c_i \mathbf{b}_i^\star$ with $|c_i| \leq 1/2$. Therefore

$$\|\mathbf{b}_j\| \leq \sum_i |c_i| \|\mathbf{b}_i^\star\|_2 \leq (1 + n/2) P(\mathbf{B}).$$

□

Algorithm 3: LLL-reduction algorithm

```
Input  : A basis $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ of a lattice $L$.
Output: An $\varepsilon$-LLL-reduced basis $\mathbf{B}$

Size-reduce the basis $\mathbf{B}$
while $\exists i$ such that $\|\mathbf{b}_i^\star\|_2 > (\gamma_2 + \varepsilon)\|\mathbf{b}_{i+1}^\star\|_2$ do
    Lagrange reduce $\mathbf{B}_{i:i+1}$ as in the WLLL algorithm.
    Size-reduce the new basis $\mathbf{B}$.
end
Return $\mathbf{B}$
```

2.2 LLL reduction and algorithm

Definition 8 *A basis $\mathbf{B}$ is ε-LLL reduced if it is both ε-WLLL reduced and size reduced.*

Usually we omit mentioning ε and instead we call such a basis LLL reduced. We also call the algorithm that achieves this reduction the LLL algorithm.

As for the WLLL algorithm, the correctness is trivial: the algorithm keeps running until the basis $\mathbf{B}$ is LLL reduced. The number of iterations of the main loop is the same as in WLLL, because the extra size-reduction step does not affect the Gram-Schmidt basis. The size of the manipulated rational is now controlled thanks to size-reduction.

But there is still a missing ingredient before we can conclude that LLL runs in polynomial time: we need to show that the Lagrange reduction algorithm is polynomial time. In the previous lecture, we only proved that the Lagrange reduction algorithm terminates after polynomially many interations in the size of its input 2 by 2 basis, but once again, one should raise the question of how large the coefficient gets during Lagrange reduction.

The lemmas that we have just demonstrated above are also applicable to Lagrange reduction. Namely, one can show that the Lagrange reduction algorithm maintains a size-reduced basis, and that the potential of the 2 by 2 basis on which it operates decreases. We leave the details to Exercise Sheet 5.

Given all the above, we reach our conclusion that LLL is polynomial time.

Theorem 9 *The LLL-basis reduction algorithm (as in Algorithm 3) with as input a basis $\mathbf{B} \subseteq \mathbb{Z}^{n\times n}$ runs in polynomial time.*

2.3 Properties of LLL reduced basis

Following the same argument as for Hermite's bound, one can prove that the first vector of an ε-LLL reduced basis satisfies $\|\mathbf{b}_1\| \leq (\gamma_2 + \varepsilon)^{\frac{n-1}{2}} \cdot \det(L)^{1/n}$. But other useful bounds can be stated as well. Before we get to them, let us state two more general lemmas that are not about LLL-reduced bases.

Lemma 10 *For any basis $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ of a lattice L, we have*

$$\lambda_1(L) \geq \min_i \|\mathbf{b}_i^\star\|_2$$

PROOF: As the Babai fundamental domain $\mathcal{P}(\mathbf{B}^\star) = \mathbf{B}^\star[-1/2, 1/2)^n$ is tiling, it contains only one lattice point, which is zero. Verify that therefore $\mathbf{B}^\star(-1,1)^n \cap L = \{0\}$. Let $r = \max\{t > 0 \mid t\mathcal{B}_2^n \subseteq \mathbf{B}^\star(-1,1)^n\}$, then clearly $\lambda_1(L) \geq r$. It is easy to see that $r = \min_i \|\mathbf{b}_i^\star\|_2$, and the proof is finished. □

LEMMA 11 *For any lattice L, one has* $\lambda_i(L_{k:n}) \leq \lambda_{i+k-1}(L)$.

PROOF: It is enough to prove that $\lambda_i(L_{k:n}) \leq \lambda_{i+1}(L_{k-1:n})$ for any $k \in \{2, \ldots, n\}$ and $i \in \{1, \ldots, n-1\}$. Here we will prove this inequality for $k = 2$, and let the reader verify that the exact same proof can easily be amended for general $k \in \{2, \ldots, n\}$. So, our aim is to prove that $\lambda_i(L_{2:n}) \leq \lambda_{i+1}(L)$, for all $i \in \{1, \ldots, n-1\}$.

Let $\mathbf{v}_1, \ldots, \mathbf{v}_n \in L$ attain the successive minima of L. Let j be the smallest index such that $\pi_2(\mathbf{v}_j) \in \operatorname{span}(\pi_2(\mathbf{v}_1), \ldots, \pi_2(\mathbf{v}_{j-1}))$, where $\pi_2(\cdot)$ is the projection onto the space orthogonal to vector $\mathbf{b}_1$ as given in Gram-Schmidt Orthogonalization for a basis $\mathbf{B}$. Set

$$\mathbf{w}_i = \begin{cases} \pi_2(\mathbf{v}_i) & \text{if} \quad i < j \\ \pi_2(\mathbf{v}_{i+1}) & \text{if} \quad i \geq j \end{cases}$$

Note that the $n-1$ vectors $\mathbf{w}_i$ are linearly independent elements of $L_{2:n}$. Furthermore, we have $\|\mathbf{w}_i\| \leq \max(\|\mathbf{v}_i\|, \|\mathbf{v}_{i+1}\|) \leq \max(\lambda_i, \lambda_{i+1}) \leq \lambda_{i+1}(L)$. This proves the claim. □

THEOREM 12 *Let* $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ *be an* ε*-LLL reduced basis. Set* $L = \mathcal{L}(\mathbf{B})$ *and* $\alpha = \gamma_2 + \varepsilon$. *Then, we have*

1. $\|\mathbf{b}_1\| \leq \alpha^{\frac{n-1}{2}} \cdot \det(L)^{1/n}$ *[Root Hermite-factor Bound]*
2. $\|\mathbf{b}_1\| \leq \alpha^{n-1} \cdot \lambda_1(L)$ *[Approximation factor Bound]*
3. $\|\mathbf{b}_i^\star\|_2 \leq \alpha^{n-i} \cdot \lambda_i(L)$.
4. $\|\mathbf{b}_i\| \leq \alpha^{i-1} \cdot \|\mathbf{b}_i^\star\|_2 \leq \alpha^{n-1} \cdot \lambda_i(L)$ *[Approximation factor alike Bound]*
5. $\prod_{i=1}^n \|\mathbf{b}_i\| \leq \alpha^{\frac{n(n-1)}{2}} \cdot \det(L)$.

Note that in the following proof, for the inequalities (1.) to (3.) no size-reduction is needed. Thus, those inequalities hold in weakly LLL-reduced bases as well. The last two inequalities, however, do depend on size-reduction.
PROOF:

1. The exact same reasoning as in proving the Hermite bound, replacing γ_2 by α, applies here.
2. We have $\|\mathbf{b}_1\| \leq \alpha^{i-1} \cdot \|\mathbf{b}_i^\star\|_2 \leq \alpha^{n-1} \cdot \min_i \|\mathbf{b}_i^\star\|_2 \leq \alpha^{n-1} \cdot \lambda_1(L)$, where the first two inequalities follow from the basis being weakly LLL-reduced and the last inequality uses Lemma 10.
3. Note that $\mathbf{B}$ being Weakly LLL-reduced means that $\mathbf{B}_{i:n}$ is Weakly LLL-reduced, too. Therefore, by part (2.) of this theorem, $\|\mathbf{b}_i^\star\|_2 \leq \alpha^{n-i}\lambda_1(L_{i:n}) \leq \alpha^{n-i}\lambda_i(L)$. The last inequality uses Lemma 11.

4. As the second bound follows from (3.), we only need to prove the first bound. Write $\mathbf{b}_i = \mathbf{b}_i^\star + \sum_{j<i} c_j \mathbf{b}_j^\star$. As the basis is both size-reduced and Weakly LLL-reduced, we know that $|c_j| \leq 1/2$ and $\|\mathbf{b}_j^\star\|_2 \leq \alpha^{i-j}\|\mathbf{b}_i^\star\|_2$ for $j < i$. Therefore

$$\|\mathbf{b}_i\|^2 \leq \|\mathbf{b}_i^\star\|_2^2 + \frac{1}{4}\sum_{j<i}\|\mathbf{b}_j^\star\|_2^2 = \|\mathbf{b}_i^\star\|_2^2\left(1 + \frac{1}{4}\sum_{j<i}\frac{\|\mathbf{b}_j^\star\|_2^2}{\|\mathbf{b}_i^\star\|_2^2}\right) \leq \|\mathbf{b}_i^\star\|_2^2\left(1 + \frac{1}{4}\sum_{j<i}\alpha^{2(i-j)}\right)$$

$$\leq \|\mathbf{b}_i^\star\|_2^2\left(1 + \frac{1}{4}\sum_{k=1}^{i-1}\alpha^{2k}\right) \leq \alpha^{2(i-1)} \cdot \|\mathbf{b}_i^\star\|_2^2.$$

The last inequality follows from the following tedious calculations. As $\alpha^2 \geq \gamma_2^2 \geq 4/3$, we have $3\alpha^2 \geq 4$, so $4(\alpha^2 - 1) \geq \alpha^2$ and therefore $1 \geq \frac{\alpha^2}{4(\alpha^2-1)}$.

$$1 + \frac{1}{4}\sum_{k=1}^{i-1}\alpha^{2k} = 1 + \frac{\alpha^2}{4}\sum_{k=0}^{i-2}\alpha^{2k} = 1 + \frac{\alpha^2}{4}\cdot\frac{\alpha^{2(i-1)} - 1}{\alpha^2 - 1} \leq 1 + (\alpha^{2(i-1)} - 1) = \alpha^{2(i-1)}$$

5. This is left as Exercise in sheet 5.

□

2.4 LLL as pre-processing

All the above bounds are upper bounds on $\|\mathbf{b}_i\|_2$ and $\|\mathbf{b}_i^\star\|_2$, but the control of the slope also allows to state lower bounds on $\|\mathbf{b}_i^\star\|_2$. Those are equally useful to prove the complexity of certain task, by using LLL to pre-process the basis before running another basis-sensitive algorithm.

Lemma 13 *If $\mathbf{B}$ is an ε-LLL reduced basis of L then $\|\mathbf{b}_i^\star\| \geq \alpha^{-i} \cdot \lambda_1(L)$.*

For example, pre-processing the basis with LLL before calling NearestPlane we obtain the following.

Theorem 14 *For any $\alpha > \gamma_2$, there exists a polynomial time algorithm solving $(\alpha^{-n}/2)$-BDD in lattices of dimensions n.*

Similarly, pre-processing the basis with LLL before calling FinckePohstEnum we obtain the following.

Theorem 15 *There exists an algorithm solving exact SVP in time $2^{O(n^2)}$ for lattices of dimension n.*

The above three statements are left as exercises. The third one requires re-using bounds in Exercise 2 of Sheet 3. **The old version of Exercise 2, Sheet 3 had mistakes. Be sure to use the up-to-date version**