Finding a shortest vector in a rotation of $\mathbb{Z}^n$

L3 Research Internship at IRISA
Supervised by Pierre-Alain Fouque and Alexandre Wallet

Jérôme Guyot

June 1^{st} - July 28^{th} 2023

Contents

1 Overview

General context

In 1994, Peter Shor described an algorithm solving the factorization and the discrete logarithm problems in polynomial time on a quantum computer. This would mean that the commonly used cryptosystems such as RSA would be broken. However for now, no quantum computer has enough power to execute Shor's algorithm and break those cryptosystems. Nevertheless, in a way to prepare for this to happen, there had been a focus on what is called "Post-Quantum Cryptography". In this way, in 2017 the NIST made a call for proposal aiming at developing new cryptographic standards.

Today, lattice-based cryptography seems to be the most mature solution. The structure of lattices allows efficient cryptosystems, and the problems on which they rely seem to be quantum resistant. Thus, 3 out of the 4 cryptosystems chosen by the NIST in 2022 are based on lattice problems.

Lattices have been widely studied, starting with ancient mathematicians such as Lagrange, Minkowski and Gauss up to more recent scientists after Shor's paper and the recent focus on it. The most studied problem in lattices is the problem of finding a short vector, it is known as SVP and has been proved to be NP-hard even up to approximation factors [Ajt98]. It is easy to see that the problem become easier when the approximation factor gets larger. The following illustration shows the time complexity of approximate SVP as the factor gets bigger.

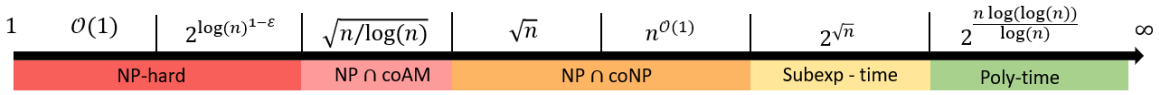


Figure 1: Time complexity of approximate SVP depending on the approximation factor

Problem studied

Even if the Shortest Vector Problem has been well studied, its reduction to rotations of $\mathbb{Z}^n$ known as $\mathbb{Z}$SVP is not as well known. It has been a long-standing open problem to determine if a polynomial-time algorithm for $\mathbb{Z}$SVP exists. At Eurocrypt 2023, an algorithm [BGPS23] was presented achieving a better time complexity than in the general case. Ducas reached the same complexity with a different approach [Duc23]. In fact, up until the two papers mentioned above, no speedup was known in the case of rotations of $\mathbb{Z}^n$. Thus, by studying these two papers, the goal is to see if a better time complexity can be achieved.

Proposed contribution

As the first paper relies on a chain of reduction that would be hard to enhance without changing the whole method, I mostly focused my work on the second one [Duc23]. I did experiments on the algorithm and studied its average running time. I proved a slightly better time complexity than what was announced in the original paper. Furthermore, I designed algorithms that are expected to have a better asymptotic running time but I could not prove it.

Field and limits of the contribution

This study illustrates the gap between the proven bounds in lattice reduction algorithm and the effective bounds. I refined the theoretical bounds on the volume reduction factor in order to prove a better running time. On its own, it is not a big result as I did not reduce the exponential factor in the running time. However having a better volume reduction factor allows for more blocks in the algorithms, thus reducing the exponent and leading to a significantly faster proven algorithm.

Results and prospects

Overall, I studied the problem of finding a shortest vector in a rotation of $\mathbb{Z}^n$ known as $\mathbb{Z}$SVP, and I proved a better running time on one of the existing algorithms. However, the major results concerns the volume reduction factor, one parameter of the algorithm, on which I proved a better theoretical bound. Further analysis could allow us to prove a better algorithm that actually works in practice and is expected to have a better time complexity.

Finally, if such an algorithm is proven, we might be able to apply it to attack the HAWK cryptosystem recently proposed at the NIST call for proposal 2023.

Meta-information

I first worked on the paper presented at Eurocrypt 2023 [BGPS23] but did not find an angle to work on. I then focused my work on [Duc23] and spent roughly 2 weeks on it but only got some minor results. I then tried to find new ways to solve $\mathbb{Z}$SVP and worked on SDKBZ and Mordell's inequalities, the orthogonality of the basis and the Seysen reduction, but none of these approaches were successful. Finally, I went back to the algorithm from [Duc23] and spent the rest of my internship on it, finally getting some interesting results.

During this internship I went to Rennes University crypto seminar 4 times as it stopped in July, and I went to the PEPR PQ-TLS project days in Paris about post-quantum cryptography.

I am still working with my supervisors on the problem of bounding the volume reduction factor as it looks quite promising.

2 Background

Notations

- Let $B = \{b_1, ..., b_n\}$ be a basis, we will note B^* the Gram-Schmidt basis of B, meaning that the b_i^* are orthogonal.
- We will denote by $\pi_B^\perp$ the orthogonal projection on the space orthogonal to $Span(B)$.
- Let B be a basis, we denote by $B_{i,j}$ the set $\{b_i, ..., b_{j-1}\}$.
- Let B be a basis, we denote by $B_{[i,j]}$ the set $\{\pi^\perp_{B_{0,i}}(b_i), ..., \pi^\perp_{B_{0,i}}(b_{j-1})\}$, with $B_{[0,j]} = B_{0,j}$.
- Let $\mathcal{S}$ be a set, we denote by $|\mathcal{S}|$ its cardinal.

2.1 Lattices

A lattice is a discrete subgroup of $\mathbb{R}^m$, $m \in \mathbb{N}^*$. Intuitively it can be seen as a vector space whose linear combinations are made with integral coefficients, it is a $\mathbb{Z} - module$. We will use $\mathcal{L}$ to refer to a lattice, and $B = \{b_1, ..., b_n\}$ for its basis where the b_i are linearly independent vectors of $\mathbb{R}^m$.

$$\mathcal{L} = \mathcal{L}(\{b_1, ..., b_n\}) = \sum_{i=1}^{n} z_i b_i \;\; z_i \in \mathbb{Z} \;,$$

The dimension/rank of a lattice is : $Dim(\mathcal{L}) = rg(\mathcal{L}) = Dim(Span(\mathcal{L}))$ where $Span(\mathcal{L})$ denotes the vector space generated by the lattice. And for $\mathcal{L} \subset \mathbb{R}^n$ if $rg(\mathcal{L}) = n$ then $\mathcal{L}$ is called full-rank. However, as opposed to free sets in a vector spaces, free sets in a lattice cannot always be completed into a basis. And as the coefficients have to be integers, in general there is no orthogonal basis of the lattice.

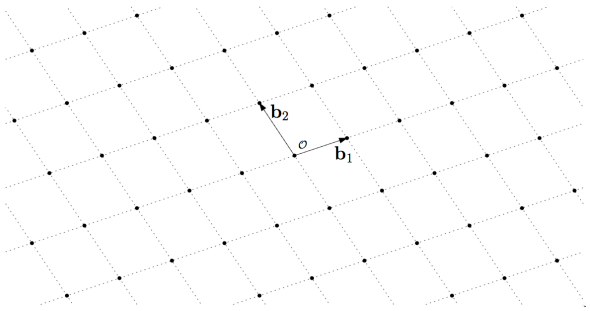


Figure 2: A two dimensional lattice

Lattice Rotations

Let $\mathcal{L}$ be a lattice and B a basis of $\mathcal{L}$. Then let us take $O \in \mathcal{O}_n(\mathbb{R})$, the lattice $\mathcal{L}(OB)$ is called a rotation of $\mathcal{L}$. More generally, the rotations of $\mathcal{L}$ are $\{\mathcal{L}(OB),\ O \in \mathcal{O}_n(\mathbb{R})\}$.

It is important to note that the rotation preserves the length and orthogonality of the vectors. This can be explained by the fact that the rotation does not change the Gram matrix of the basis : $(OB)^T OB = B^T B$. This means that the rotations of $\mathbb{Z}^n$ also have an orthonormal basis. Furthermore, note that all results and proofs will be written assuming that $\mathcal{L} = \mathbb{Z}^n$, it still holds if $\mathcal{L} = O\mathbb{Z}^n$, we just need to write O everywhere needed in the proofs.

2.2 Problems on lattices

The two fundamental problems on lattices are finding a shortest non-zero vector in the lattice, and given a vector in the space finding the closest lattice vector. Thus, short vectors are at the center of the theory of lattice based-cryptography. Moreover, those problems are shown to be NP-hard under randomized reductions [Ajt98]. Note that there is no uniqueness for a shortest vector as v and $-v$ have the same length.

SVP (Shortest Vector Problem) : Given a lattice $\mathcal{L}$, find v a non zero shortest vector of $\mathcal{L}$.

CVP (Closest Vector Problem) : Given a lattice $\mathcal{L}$ and a vector $w \in \mathbb{R}^m$, find $v \in \mathcal{L}$ minimizing $\|w - v\|$.

Let us denote by $\lambda_1(\mathcal{L})$ the length of a shortest vector of $\mathcal{L}$. As these problems are hard, relaxed variants have been defined and studied such as

γ-SVP (Approximate Shortest Vector Problem) : Given a lattice $\mathcal{L}$, find v a vector of $\mathcal{L}$ such that $\|v\| \leqslant \gamma\ \lambda_1(\mathcal{L})$.

Furthermore, there are also restriction of these problems to certain class of lattices, such as the restriction to rotation of $\mathbb{Z}^n$.

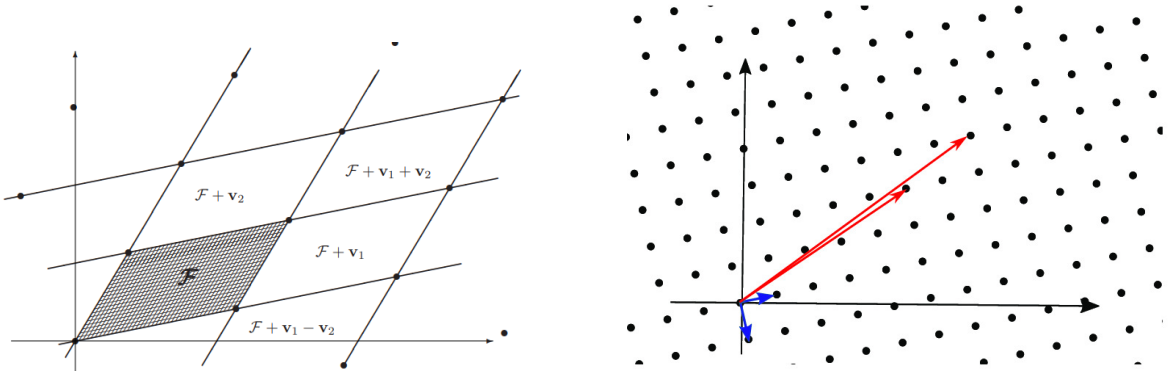
$\mathbb{Z}$SVP : Given a rotation of $\mathbb{Z}^n$ $\mathcal{L}$, find v a non zero shortest vector of $\mathcal{L}$.

2.3 Mathematical notions on lattices

Volume and fundamental domain

A lattice does not have a unique basis, however as the coefficients of the change-of-basis-matrix have to be in $\mathbb{Z}$ and invertible, we get that those matrices have determinant 1 or -1.

As the basis is not unique, one might like to find a parameter that only depend on the lattice and not on the basis.



(a) A fundamental domain

(b) A lattice with good and bad basis

Definition 2.1. *Let $\mathcal{L}$ be a lattice and B a basis of $\mathcal{L}$. Then the volume of $\mathcal{L}$ is given by*

$$Vol(\mathcal{L}) = Det(\mathcal{L}) = |Det(B)| \ .$$

The volume is a lattice invariant of the basis, and represents the volume of what is called the fundamental domain of the lattice, often denoted by $\mathcal{F}$.

Good and bad basis

The hardness of the SVP and CVP problems is highly dependent on the basis we have for the lattice. As can be seen in figure (b) above, having the blue basis makes it easier to find a short vector than doing the same with the red one. Thus the terms "good" and "bad" basis.

One could see that the good basis has relatively short and orthogonal vectors. This will lead to the creation of lattice reduction algorithms, which are algorithms aiming at transforming a bad basis in a good one. The most famous one being the LLL algorithm [Len82] which runs in polynomial time but returns exponentially large vectors. Furthermore, there are other algorithms running in exponential time and achieving better bounds on the vectors of the basis. For example, the k-BKZ algorithm achieves the bound $k^{\frac{1}{2k}} Vol(\mathcal{L})^{\frac{1}{n}}$ in time $\mathcal{O}(k^{\frac{k}{2e}})$.

Hermite's constants and Gaussian heuristic

When looking at SVP, one might like to know how short is the shortest vector, meaning what bound can be achieved. Minkowski proved that

$$\lambda_1(\mathcal{L}) \leqslant \sqrt{n}\ Vol(\mathcal{L})^{\frac{1}{n}} \ .$$

This bound is useful when working with short vectors of a lattice, however reduction algorithms often work with another notion known as Hermite's constants. These constants are defined as

$$\gamma_n = \sup_{Dim(\mathcal{L})=n} \left(\frac{\lambda_1(\mathcal{L})}{Vol(\mathcal{L})^{\frac{1}{n}}} \right)^2 \ ,$$

which leads to the Hermite's bound :

$$\forall\ \mathcal{L} \text{ with } dim(\mathcal{L}) = n, \quad \lambda_1(\mathcal{L}) \leqslant \sqrt{\gamma_n}\ Vol(\mathcal{L})^{\frac{1}{n}} \ .$$

It has been showed that asymptotically, $\frac{n}{2\pi e} < \gamma_n \leqslant \frac{1.744n}{2\pi e} + o(n)$ which improves slightly on Minkowski.

Finally, algorithms on lattices often use the Gaussian approximation which states that the number of points of a full-rank lattice within S a measurable subset of $\mathbb{R}^n$, is roughly $\frac{Vol(S)}{Vol(\mathcal{L})}$.

Even though this approximation holds well for generic lattices, it does not work for $\mathbb{Z}^n$. This means that we will not be able to use it directly, but we can still use it on random sublattice of $\mathbb{Z}^n$. To show

why the heuristic does not hold on $\mathbb{Z}^n$, consider the intersection between this lattice and the closed ball or radius 1 centered in 0. This ball has an asymptotic volume of $(\frac{2\pi e}{n})^{\frac{n}{2}}$, and all the n unit vectors are in it, thus giving

$$|\mathbb{Z}^n \cap B_f(0,1)| = n \gg \left(\frac{2\pi e}{n}\right)^{\frac{n}{2}} .$$

In fact $\mathbb{Z}^n$ is a very unique lattice as is has an orthonormal basis. This rich structure is what gives the intuition that faster algorithms should exist when working exclusively with rotations of $\mathbb{Z}^n$.

For further explanations on those notions there is an analysis of those bounds and parameters and their link with reduction algorithms made in [Ngu10].

2.4 Dual Lattice

An important notion when working with lattices is the dual lattice. Let $\mathcal{L}$ be a lattice, its dual lattice is defined as

$$\mathcal{L}^\vee = \{v \in Span(\mathcal{L}) \mid \forall\ u \in \mathcal{L},\ \langle v, u\rangle \in \mathbb{Z}\} .$$

The dual lattice is indeed a lattice, and has some close link with the primal lattice $\mathcal{L}$.

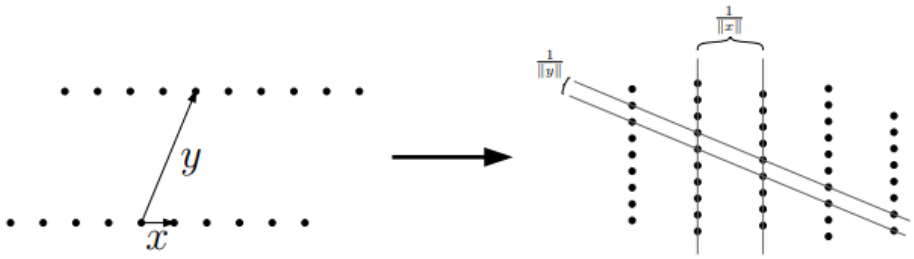


Figure 4: A lattice and its dual

Some properties of the duality

1. Let $\alpha \in \mathbb{R}^*$ then $(\alpha\mathbb{Z}^n)^\vee = \frac{1}{\alpha}\mathbb{Z}^n$.
2. Let $\mathcal{L}$ be a lattice, then $Vol(\mathcal{L}^\vee) = \frac{1}{Vol(\mathcal{L})}$.
3. Let B be a basis of $\mathcal{L}$ then $B(B^TB)^{-1}$ is a basis of $\mathcal{L}^\vee$.
4. $Span(\mathcal{L}) = Span(\mathcal{L}^\vee)$.
5. The dual of the dual is the primal : $(\mathcal{L}^\vee)^\vee = \mathcal{L}$.
6. $\mathcal{L}(B_{[i,j]})^\vee$ is a sublattice of $\mathcal{L}(B)^\vee$.

2.5 Lattice Algorithms

A more complete and detailed description of lattice algorithms can be found in 7.1.

The best exact SVP algorithm runs in $\mathcal{O}(2^n)$, where n is the dimension of the lattice. However, when we want to solve approximate SVP, we can acutally go faster. Algorithms such as BKZ or SD-BKZ use exact SVP algorithms as a subroutine. For example, the bound on SDBKZ [MW15] when applying the exact SVP in dimension k is $\|b_1\| \leqslant \sqrt{\gamma_k}^{\frac{n-1}{k-1}}\ \lambda_1(\mathcal{L})$, achieved in $\mathcal{O}(Poly(n)2^k)$.

Finally, in this work we will use the HKZ reduction algorithm. It works by taking a shortest vector of the lattice, and then search a lattice vector whose projection on the hyperplane orthogonal to the

previous vector is non zero and minimal among other lattice vectors. It then iterates this procedure, taking a vector whose projection on the space orthogonal to the current basis is a shortest vector. More formally, a basis is HKZ reduced if :

$$\|\pi^{\perp}_{B_{1,i}}(b_i)\| = \lambda_1(\pi^{\perp}_{B_{1,i}}(\mathcal{L}))\ ,$$

$$\forall\ j < i,\ \left|\frac{\langle b_i, \pi^{\perp}_{B_{1,j}}(b_j)\rangle}{\|\pi^{\perp}_{B_{1,j}}(b_j)\|^2}\right| < \frac{1}{2}\ .$$

2.6 What has been done on rotations of $\mathbb{Z}^n$?

What I described above is in the case of general lattices, that can be quite complex and have less structure than $\mathbb{Z}^n$. In fact, up until recently, the faster algorithm to solve the shortest vector problem in $\mathbb{Z}^n$ was the general algorithm. Even if $\mathbb{Z}^n$ seems way simpler to manipulate than general lattices, as it does not respect the Gaussian heuristic, it can be quite tricky.

In 2023, Benett and al. published a $\mathcal{O}(2^{\frac{n}{2}+o(n)})$ algorithm for $\mathbb{Z}$SVP [BGPS23]. This algorithm uses a chain of reduction on various lattice problems. However it can be seen as : they transformed via a clever sampling the rotation of $\mathbb{Z}^n$ into a γ-Unique-SVP lattice for a well chosen γ, which means that the lattice they obtained had $\lambda_2(\mathcal{L}) \geqslant \gamma\lambda_1(\mathcal{L})$. Then, they used an algorithm solving $(1.93 + o(1))$-Unique-SVP in $\mathcal{O}(2^{\frac{n}{2}+o(n)})$ via a reduction to another problem [LM09] [ADRS15]. This is a really significant improvement compared to the best SVP algorithm in $\mathcal{O}(2^n)$.

In 2023 also, Ducas in [Duc23] designed an algorithm for $\mathbb{Z}$SVP in the same time complexity but using a different approach. This algorithm will be described and studied in the next section.

3 Solving $\mathbb{Z}$SVP

Ducas in [Duc23] presented a two step algorithm solving $\mathbb{Z}$SVP in $\mathcal{O}(2^{\frac{n}{2}})$. The algorithm first takes an odd n or add a dimension, and slices the basis in two blocks of size $\frac{n}{2}$ and $\frac{n}{2}+1$. Once sliced, it transforms the blocks until obtaining two blocks isomorphic to $\mathbb{Z}^{\frac{n}{2}}$ and $\mathbb{Z}^{\frac{n}{2}+1}$. Finally applying a classic SVP algorithm on those blocks returns an orthonormal basis of the lattice. The following algorithms will also aim at finding such a basis as in the case of rotations of $\mathbb{Z}^n$, those problems are equivalent. In fact you can use an HKZ once the blocks are orthogonal to get an orthonormal basis, or an SVP to get only a shortest vector.

The reason for the choice of $\frac{n}{2}$ for the length of the blocks is that given k the length of the first block, it costs $\mathcal{O}(2^{n-k})$ to make it orthogonal to the rest of the basis.

Remark. *The algorithm uses an HKZ oracle once the blocks were obtained as the complexity of getting those blocks is the same as an HKZ in dimension $\frac{n}{2}$. However if one could get those blocks in a better complexity, then continuing by induction until dimension one would be better. Thus, keeping the complexity of the extraction of the blocks as the general complexity.*

Remark. *As the algorithm from [Duc23] returns a basis of the rotation, it answers the problem of finding the isomorphism between $\mathcal{L}$ and $\mathbb{Z}^n$ known as $\mathbb{Z}$LIP where LIP is the Lattice Isomorphism Problem. The general LIP is, given two lattices that are rotations of each other, find the rotation to go from one to another. Currently, the best known algorithm solving LIP is in $n^{\mathcal{O}(n)}$. Using HKZ, we can see that $\mathbb{Z}$LIP and $\mathbb{Z}$SVP are equivalent.*

3.1 Detecting the blocks

In [Duc23], the volume is used to detect when blocks are isomorphic to $\mathbb{Z}^{\frac{n}{2}}$. For this, it is proved that when the volume of the first block is 1 then it is isomorphic, which explains why the goal is to reduce the volume.

However this method could be applied to a much larger class of lattices. In fact, Ducas uses the fact that the volume cannot go lower than one. If we see the situation in a more global way, he is looking for the sublattice of dimension $\frac{n}{2}$ of minimal volume. Thus, this method should work on lattices that have a known dimension $m < n$ such that sublattices of minimal volume of dimension m have their orthogonal lattice within the lattice.

3.2 Reducing the volume

For now, we will consider the volume approach and this special case of a target sublattice with minimal volume. It means that we now have to reduce the volume of the blocks.

The algorithm in [Duc23] uses the middle vector $b_{\frac{n}{2}}$ as a pivot. A volume invariant transformations is applied to the $\frac{n}{2}+1$ last vectors which reduces the length of the middle vector : SVP reduce, meaning that the middle vector is the shortest vector of the lattice generated by the block. Then a similar transformation is used on the $\frac{n}{2}+1$ first vectors this time making the middle vector as large as possible : DSVP reduce meaning that the SVP reduces the dual lattice of the block, thus, making the middle vector quite large. At the end, one can show that the volume of the first block has decreased by at least a factor $\sqrt{1-\frac{1}{n}}$.

There is a more visual representation of this volume reduction phase in 6.2.

3.3 Algorithm for $\alpha\mathbb{Z}$SVP

Algorithm 1 is a slight modification of the algorithm [Duc23]. The only difference is the α in the condition of the while. This could also be done by scaling $\alpha\mathbb{Z}^n$ to $\mathbb{Z}^n$ and scaling it back after, but this shows that this manipulation is in fact useless as the algorithm works as well in the original lattice.

Algorithm 1 algorithm for $\alpha\mathbb{Z}$SVP

Require: B basis of $\mathcal{L}$, $\mathcal{L}$ rotation of $\alpha\mathbb{Z}^n$
Ensure: B basis of $\mathcal{L}$ with $\frac{1}{\alpha}B$ orthonormal
  $\alpha \leftarrow Vol(\mathcal{L})^{\frac{1}{2n}}$
  $B \leftarrow LLL(B)$
  $B_{k,2k+1} \leftarrow$ Primal-SVP Reduce $(B_{k,2k+1})$
  while $Vol(B_{0,k+1}) \geqslant \sqrt{2}\alpha^{2k}$ **do**
    $B_{0,k+1} \leftarrow$ Dual-SVP Reduce $(B_{0,k+1})$
    $B_{k,2k+1} \leftarrow$ Primal-SVP Reduce $(B_{k,2k+1})$
  end while
  $B_1 \leftarrow$ Primal-HKZ Reduce $(B_{0,k})$
  $B_2 \leftarrow$ Primal-HKZ Reduce $(B_{k,2k+1})$
  return B

4 A better time complexity

From now on and until the end, all results are my contribution, the only exceptions being 6.1 which is a detailed version of the proofs of [Duc23] and 7.1 which describes lattice algorithms.

4.1 Experimenting with more blocks

One could easily see that the complexity of the algorithm lies in the transformation from random sublattice to $\mathbb{Z}^k$. And in this phase we are using an exact SVP algorithm on blocks of size k. The first idea that comes to mind when we want to make it faster is to reduce k, which means cutting the original basis into more blocks.

However, when cutting into more than 2 blocks, it is hard to see how we can prove an actual volume reduction. In fact the strategy of the proofs in [Duc23] cannot be applied. Thus, we are not able to

prove that a similar algorithm will actually finish.

We experimented with some variants of the original algorithms working with 3 and 4 blocks in dimension up to 120. The algorithm finished, and returned the expected output. This suggests that more can be done on this problem. However, when increasing the number of blocks, the volume reduction factor increased too. It went from an average of 0.35 for 2 blocks, to 0.77 for 4 blocks, and it became relatively frequent for the reduction factor to be bigger than 1. There is an example of what the reduction factor can look like with 4 blocks in 6.2.

Furthermore, one can see that the effective volume reduction factor is extremely smaller than the predicted bound of $\sqrt{1-\frac{1}{n}}$. In fact, this proved bound is a worst case bound. Thus, by making a more refined analysis we could lower this bound.

If we are able to get better bounds on the volume reduction, then we might be able to prove that the algorithm finishes even with more blocks. Thus, allowing a significant proven speedup. This is why we will be working on a tight bound on the theoretical volume reduction.

4.2 Studying the random lattice case

It is known that the lattices generated by the first block are not random as they are converging towards $\mathbb{Z}^k$. However, for the first iterations of the algorithm, they can be considered quite random. For now, we will put those question on the side and focus on what we could achieve if we do have random lattices.

Ajtai proved in [Ajt02] that in the case of random lattices, the shortest vector will follow the Gaussian heuristic with high probability. Using the Gaussian heuristic on an n-dimensional lattice $\mathcal{L}$ with volume $Vol(\mathcal{L})$ we expect a shortest vector of length $\lambda_1(\mathcal{L}) = \left(\frac{Vol(\mathcal{L})}{V_n(1)}\right)^{\frac{1}{n}}$, which is asymptotically equal to $\sqrt{\frac{n}{2\pi e}}Vol(\mathcal{L})^{\frac{1}{n}}$. This allows to write the following lemma, proved in 6.7.

Lemma 4.1. *Let $\mathcal{L}$ be a random lattice, let us call $r = \frac{Vol(\mathcal{L}(B_{0,k}))_{new}}{Vol(\mathcal{L}(B_{0,k}))_{old}}$ the volume reduction factor during one loop iteration. Then we have*

$$\mathbb{E}(r) = \left(\frac{1}{V_{k+1}(1)Vol(\mathcal{L}(B_{0,k}))}\right)^{\frac{1}{k+1}(2-\frac{1}{k+1})},$$

where $V_n(1)$ is the volume of the n-dimensional ball of radius 1.

Thus, this reduction factor only has sense as long as $Vol(\mathcal{L}(B_{0,k})) > \frac{1}{V_{k+1}(1)} > 1$. This shows that such a bound cannot be used until the end, even if we consider perfectly random lattices. As said before, we will not be able to use this approach as we do not have random lattices, but we can have a relatively close result.

4.3 Studying the slide reduction

The slide algorithm is a lattice reduction algorithm described in [GN08]. It works quite similarly as what we just saw but with p blocks. The idea is to do multiple iterations of a round, each round being composed of a primal round and a dual round. During the primal round, all blocks are SVP-reduced. Whereas, during the dual round, the $p-1$ blocks obtained by shifting by 1 all the $p-1$ fist blocks are DSVP-reduced.

The while loop we are using for the volume reduction is essentially a round of the slide reduction process described in [GN08] with blocks of size $\frac{n}{2}$, this is what the following lemma is about, and is proven in 6.8.

Lemma 4.2. *The volume reduction loop is a round of the slide algorithm.*

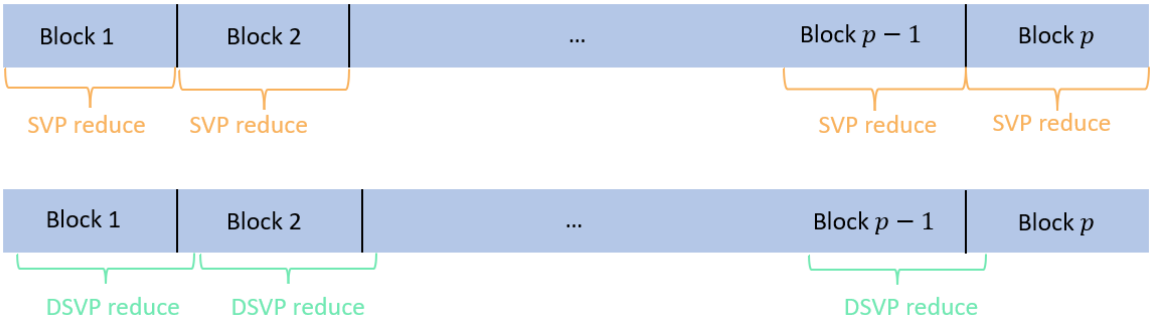


Figure 5: Visualisation of the slide algorithm

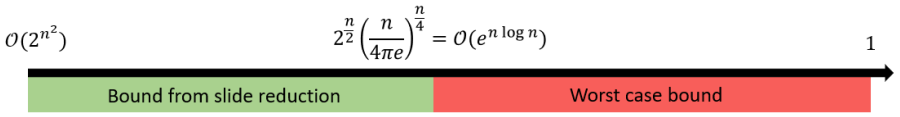


Figure 6: Volume reduction steps

The idea is to extend the ambient space to $\mathbb{Z}^{n+1}$ and add at the beginning of the basis a vector $\bar{b} = (0, ..., 0, \alpha)$ with $\alpha < 1$. We can then prove that a loop of the slide algorithm with blocksize $\frac{n+1}{2}$ is the same as our volume reduction loop.

Using the bounds proven in [Wal20] for the slide reduction we can show that once the basis is LLL reduced, it takes $\mathcal{O}(n\log(n))$ iterations of the loop to reach a volume of $2^{\frac{n}{2}}\sqrt{\gamma_{\frac{n}{2}}}^{\frac{n}{2}}$. Once this volume is reached, we can apply the classic bound proven before. Using the asymptotic formula of $\gamma_{\frac{n}{2}}$ gives us that this volume is in $\mathcal{O}(2^{n\log(n)})$. This means that going from this volume to 1 takes at most $\mathcal{O}(n^2\log(n))$ iterations when we use the proven worst case volume reduction factor. Thus, giving the theorem below, proved in 6.9.

Theorem 4.3. *The volume reduction phases finishes in at most $\mathcal{O}(n^2\log(n))$ iterations.*

In fact, the bounds from the slide reduction are close to the one we obtain using the Gaussian heuristic. Meaning that the general behavior until a volume of $\mathcal{O}(2^{n\log(n)})$ is roughly the same. This explain why we use the formula from this approach in the visualisation. Thus, the red curve is only here to give a rough idea of how the theoretical volume reduction factor evolves over the iterations.

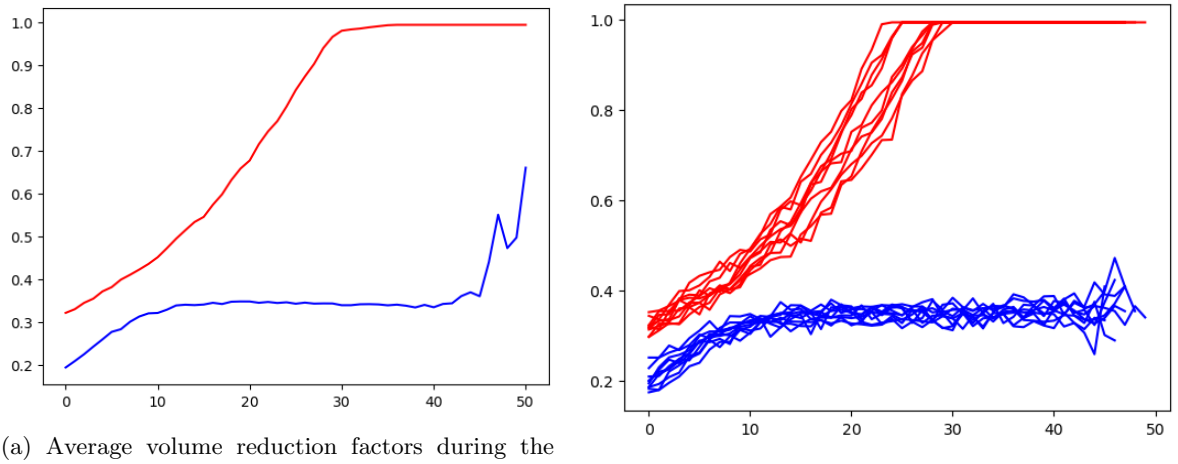
When applying the algorithm with $n = 80$ we get the graphs 7, where the abscissa is the number of iterations and the ordinate is the volume reduction factor. The blue curve corresponds to the effective volume reduction factor whereas the red is for the theoretical one.

Using the bounds from the slide reduction, we were able to greatly reduce the theoretical volume reduction factor in what we call the green phase : where the volume of the first block is bigger than $\mathcal{O}(2^{n\log(n)})$. The second part, called the red phase, is more complex and for now we only have the worst case bound.

4.4 Studying the red phase

The goal here is to reduce the theoretical volume reduction factor when the lattice is relatively close to $\mathbb{Z}^k$. Furthermore, the Gaussian heuristic or Minkowski's bounds cannot be used here.

We will use a new method inspired by Johnson-Lindenstrauss lemma : we will bound the length of the shortest vector in the projected block by the length of the shortest non zero projection of an



(a) Average volume reduction factors during the loop iterations.

(b) Distribution of the volume reduction factors.

Figure 7: Comparison between effective and theoretical volume reduction factor during the loop iterations

unit vector. Johnson-Lindenstrauss lemma states that the length of unit vector projected on a random k-dimensional vector space will be roughly $\sqrt{\frac{k}{n}}$. However we will not use it in this way. In fact, we will show that for every k-dimensional subset that is not generated by k canonical vectors e_i, there is a e_i with a non-zero projected norm smaller than $\sqrt{\frac{k-|S|}{n-|S|-|S^\perp|}}$, where $S = \{e_i \in W\}$ and $S^\perp = \{e_i \in W^\perp\}$ where $W^\perp$ is the space orthogonal to W.

Theorem 4.4. *Let W be a k-dimensional space with $W \subset \mathbb{R}^n$, $W \neq \mathbb{R}^k$. Let $S = \{e_i \in W\}$ and $S^\perp = \{e_i \in W^\perp\}$ where $W^\perp$ is the space orthogonal to W. Then $0 < \min_{e_i \notin W^\perp} \|\pi_W(e_i)\|^2 \leqslant \frac{k-|S|}{n-|S|-|S^\perp|}$.*

The proof of the theorem uses the fact that a k-dimensional subspace W can be represented by (Q, J) with $Q \in \mathcal{O}_n(\mathbb{R})$ and J a subset of k elements of $\{1, ..., n\}$. This way, W is generated by the columns of Q with indexes in J. And using the properties of columns of Q we get the results.

Remark. *The condition $W \neq \mathbb{R}^k$ is only here to ensure that the right column of the block matrix exists.*

This theorem, proved in 6.11, allows a better bounding on the shortest vector in the projection as we are bounding the projection of the unit vectors. However, it comes with the price of studying $S = \{e_i \in W\}$ and $S^\perp = \{e_i \in W^\perp\}$. In the case of 2 blocks, we might be able to use a little symmetry trick to bound $|S| - |S^\perp|$. The idea is to say that they are roughly equal, this conjecture is further developed in 6.12 and might lead to a proof of $\mathbb{E}(r) \leqslant \frac{1}{2}$ with high probability.

4.5 Algorithm on p blocks

As mentioned before, we can design variant of the algorithm working with more than 2 blocks. After reading 4.3 it is easy to see that the intuitive extension is the slide algorithm applied to p blocks. This gives algorithm 3 that is written in annex 6.4 in pseudo-code and illustrated by a schematic representation. This allows us to use the bounds of the slide algorithm in the green phase, making the analysis easier.

This algorithm actually works when used in dimension 120 with blocks of size 30 or 40. One thing that should be noted is that even if it finishes, there are some rare execution of the loop where the volume of the first block does not decrease and might increase. Thus, we need to evaluate the expected volume reduction factor and not the worst case. It is relatively easy to see that the more blocks there are, the bigger the expected volume reduction factor is, and eventually it will be bigger than one with too many blocks.

Let us call r_i^k the volume reduction factor between the blocks i and $i+1$ with block-size k, B_i^k the i^{th} block of B and $B_i^{k\perp} = \pi^{\perp}_{B_1^k,\dots,B_{i-1}^k}(B_i^k)$ the i^{th} block projected on the orthogonal of the $i-1$ first blocks. Let D be the reversed dual basis of B, and let $n = pk$. We get

$$r_i^k = \lambda_1(\mathcal{L}(B_{i+1}^{k\perp}))\ \lambda_1(\mathcal{L}(D_{p-i}^{k\perp})) \text{ for } 1 \leqslant i \leqslant p-1\ .$$

Thus, the goal of future studies, will be to find or bound

$$\kappa_n = \max_{1\leqslant k\leqslant n}\left(\frac{n}{k} \mid \mathbb{E}(r_1^k) < 1\right)\ .$$

Remark. *Intuitively, we are taking the maximum number of blocks $p = \frac{n}{k}$ for which we can prove the volume reduction.*

The real condition is in fact $1-\mathbb{E}(r_1^k)$ not negligible compared to n as we want a polynomial number of iteration of the loop.

The reason why we are only considering r_1^k is that one we get that $\mathcal{L}(B_1^k)$ is isomorphic to Z^k, we can apply HKZ on it, and return k shortest vectors. As mentioned before in the case of $\mathbb{Z}^n$ getting a shortest vector and an orthonormal basis is equivalent.

Finally, by taking $\kappa = \inf_{n\in\mathbb{N}*}(\kappa_n)$, we get that there is an algorithm finishing in time $\mathcal{O}(2^{\frac{n}{\kappa}})$ with high probability.

Remark. *The paper [Duc23] actually proves $\kappa \geqslant 2$.*

To sum up, by studying r_i^k the same way we studied r in the previous parts, and getting better theoretical bounds on it, we might be able to prove a bigger κ_n, which would lead to a bigger κ. Thus we would get a faster algorithm for $\mathbb{Z}$SVP.

5 Conclusion

The problem of finding a shortest vector in a rotation of $\mathbb{Z}^n$ is a relatively young problem. It could be useful to know if this problem is actually hard or not. If it is hard, we would be able to come up with efficient cryptographic schemes using the structure of $\mathbb{Z}^n$ to have faster computation and smaller keys.

In fact, it has been a long standing problem to determine if a polynomial algorithm could solve this problem. However before this year, little had been discovered. Then two papers were published [BGPS23][Duc23] and actually found a way to solve $\mathbb{Z}$SVP in $\mathcal{O}(2^{\frac{n}{2}})$ which is significantly faster than solving SVP (best algorithm in $\mathcal{O}(2^n)$). Thus, my work aimed at understanding the problem and finding a speedup based on these two papers.

Even if for now I did not came up with a significant proven speed up, the various experiments I did with the variants of the algorithm with more than two blocks suggests that those variants actually work with a probability depending on the number of blocks. The work I did on the bounding of the volume reduction factor, aims at proving that those algorithms finish and find the bigger number of blocks such that it finishes.

With further study, we could find this number of blocks, and prove a significantly faster algorithm that finishes with high probability. However, for this, we need to study the expected volume reduction factor when we have more blocks which seems harder than for 2 blocks.

This internship allowed me to discover the world of research, and lattice-based cryptography. I really enjoyed working on this subject and consolidated the idea that I am really interested in cryptography and in research. It also made me curious about quantum algorithmic as we are trying to build quantum secure schemes.

Finally I would like to thanks Pierre-Alain Fouque and Alexandre Wallet for their guidance and help during those 2 months, as well as the CAPSULE team and the personnel of IRISA. It was an extremely enjoyable experience and I am already looking forward to my next internship.

References

[ADRS15] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. "Solving the Shortest Vector Problem in 2n Time Using Discrete Gaussian Sampling: Extended Abstract". In: *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*. STOC '15. Portland, Oregon, USA: Association for Computing Machinery, 2015, pp. 733–742. ISBN: 9781450335362. DOI: 10.1145/2746539.2746606. URL: https://doi.org/10.1145/2746539.2746606.

[Ajt02] M. Ajtai. "Random lattices and a conjectured 0 - 1 law about their polynomial time computable properties". In: *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* 2002, pp. 733–742. DOI: 10.1109/SFCS.2002.1181998.

[Ajt98] Miklós Ajtai. "The Shortest Vector Problem in L2 is NP-Hard for Randomized Reductions (Extended Abstract)". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: Association for Computing Machinery, 1998, pp. 10–19. ISBN: 0897919629. DOI: 10.1145/276698.276705. URL: https://doi.org/10.1145/276698.276705.

[ALS20] Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. *A $2^{n/2}$-Time Algorithm for $\sqrt{n}$-SVP and $\sqrt{n}$-Hermite SVP, and an Improved Time-Approximation Tradeoff for (H)SVP*. 2020. arXiv: 2007.09556 [cs.DS].

[BGPS23] Huck Bennett, Atul Ganju, Pura Peetathawatchai, and Noah Stephens-Davidowitz. "Just How Hard Are Rotations of $\mathbb{Z}^n$? Algorithms and Cryptography with the Simplest Lattice". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 252–281. DOI: 10.1007/978-3-031-30589-4_9. URL: https://doi.org/10.1007/978-3-031-30589-4%5C_9.

[CFL28] Richard Courant, Kurt Friedrichs, and Hans Lewy. "Über die partiellen Differenzengleichungen der mathematischen Physik". In: *Mathematische annalen* 100.1 (1928), pp. 32–74.

[Duc23] Léo Ducas. *Provable Lattice Reduction of $\mathbb{Z}^n$ with Blocksize $n/2$*. Cryptology ePrint Archive, Paper 2023/447. https://eprint.iacr.org/2023/447. 2023. URL: https://eprint.iacr.org/2023/447.

[Fis05] Ernst Fischer. "Über quadratische Formen mit reellen Koeffizienten". In: *Monatshefte für Mathematik und Physik* 16 (1905), pp. 234–249.

[GN08] Nicolas Gama and Phong Q. Nguyen. "Finding Short Lattice Vectors within Mordell's Inequality". In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC '08. Victoria, British Columbia, Canada: Association for Computing Machinery, 2008, pp. 207–216. ISBN: 9781605580470. DOI: 10.1145/1374376.1374408. URL: https://doi.org/10.1145/1374376.1374408.

[HS07] Guillaume Hanrot and Damien Stehlé. "Improved Analysis of Kannan's Shortest Lattice Vector Algorithm". In: *Advances in Cryptology - CRYPTO 2007*. Ed. by Alfred Menezes. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 170–186. ISBN: 978-3-540-74143-5.

[Kan83] Ravi Kannan. "Improved Algorithms for Integer Programming and Related Lattice Problems". In: *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*. STOC '83. New York, NY, USA: Association for Computing Machinery, 1983, pp. 193–206. ISBN: 0897910990. DOI: 10.1145/800061.808749. URL: https://doi.org/10.1145/800061.808749.

[Len82] Lovász L. Lenstra H.W. jr. Lenstra A.K. "Factoring Polynomials with Rational Coefficients." In: *Mathematische Annalen* 261 (1982), pp. 515–534. URL: http://eudml.org/doc/182903.

[LM09] Vadim Lyubashevsky and Daniele Micciancio. "On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem". In: *Advances in Cryptology - CRYPTO 2009*. Ed. by Shai Halevi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 577–594. ISBN: 978-3-642-03356-8.

[MW15] Daniele Micciancio and Michael Walter. *Practical, Predictable Lattice Basis Reduction*. Cryptology ePrint Archive, Paper 2015/1123. https://eprint.iacr.org/2015/1123. 2015. URL: https://eprint.iacr.org/2015/1123.

[Ngu10] Phong Q. Nguyen. "Hermite's Constant and Lattice Algorithms". In: *The LLL Algorithm: Survey and Applications*. Ed. by Phong Q. Nguyen and Brigitte Vallée. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 19–69. ISBN: 978-3-642-02295-1. DOI: 10.1007/978-3-642-02295-1_2. URL: https://doi.org/10.1007/978-3-642-02295-1_2.

[Wal20] Michael Walter. *The Convergence of Slide-type Reductions*. Cryptology ePrint Archive, Paper 2020/1409. https://eprint.iacr.org/2020/1409. 2020. DOI: 10.1007/978-3-030-75245-3_3. URL: https://eprint.iacr.org/2020/1409.

6 Annex

6.1 Algorithm from DUC23 and proof

This subsection is not original work, it is only here to help with the understanding of the algorithm, all proofs are from [Duc23], with the only modification being the coefficient α, and some more details in the proofs. In this way the proof is made for algorithm 1, but replace $\sqrt{2}\alpha^{2k}$ by 1 and and α by 1 if you want the proof for 2.

The following algorithm was proposed in [Duc23] and actually solves $\mathbb{Z}SVP$ in $\mathcal{O}(2^{\frac{n}{2}})$.

Algorithm 2 Algorithm from Ducas

Require: B basis of $\mathcal{L}$, $\mathcal{L}$ rotation of $\mathbb{Z}^n$
Ensure: B orthonormal basis of $\mathcal{L}$
  $B \leftarrow LLL(B)$
  $B_{[k,2k+1]} \leftarrow$ Primal-SVP Reduce $(B_{[k,2k+1]})$
  while $Vol(B_{0,k+1}) > 1$ **do**
    $B_{0,k+1} \leftarrow$ Dual-SVP Reduce $(B_{0,k+1})$
    $B_{[k,2k+1]} \leftarrow$ Primal-SVP Reduce $(B_{[k,2k+1]})$
  end while
  $B_{0,k} \leftarrow$ Primal-HKZ Reduce $(B_{0,k})$
  $B_{[k,2k+1]} \leftarrow$ Primal-HKZ Reduce $(B_{[k,2k+1]})$
  return B

We now want to prove the correction and the termination of the algorithm. This is why we introduce those lemmas. The first one is for "detecting the block" to see if we finished the operation. While the other two are for the volume reduction.

Lemma 6.1. *Let $\mathcal{L} \subset \alpha\mathbb{Z}^n$, $rg(\mathcal{L}) = k$. Then $\frac{Vol(\mathcal{L})^2}{\alpha^{2k}} \in \mathbb{N}^*$, and if $Vol(\mathcal{L})^2 = \alpha^{2k}$ Then $\mathcal{L}$ is isomorphic to $\alpha\mathbb{Z}^k$*

Proof of 6.1. For the first point, let B be a basis of $\mathcal{L}$. This means that $B = \alpha B'$ with B' basis of $\mathbb{Z}^n$. Thus

$$Vol(\mathcal{L})^2 = Det(B^T B) = \alpha^{2k} Det(B'^T B') \ .$$

However, B' is a basis of $\mathbb{Z}^n$, which means that $Det(B'^T B') \in \mathbb{N}^*$. Hence $\frac{Vol(\mathcal{L})^2}{\alpha^{2k}} \in \mathbb{N}^*$.

For the second point, let us consider B' in its Hermite normal form : i.e. $B' = UB_H$, $B_H = \begin{bmatrix} X \\ Y \end{bmatrix}$, where U is a rotation matrix, and X is lower triangular such that all the lower coefficients are positive integers and smaller than the one on the diagonal : $\forall i \leqslant n,\ \forall j \leqslant i,\ 0 \leqslant X[i,j] < X[i,i]$.

Furthermore, writing λ_i for the eigenvalues of $B_H^T B_H$, and by using the Courant-Fischer [CFL28] [Fis05] theorem which can be used as $B_H^T B_H$ is symmetric and real, stating that in this case

$$\lambda_i = \max_{V \subset \mathbb{R}^k,\ dim(V)=i} \{ \min_{x \in V,\ \|x\|=1} \{ \langle B_H^T B x, x \rangle \} \} \ ,$$

we get

$$Det(B'^T B') = Det(B_H^T B_H) = \prod_{i \leqslant k} \lambda_i = \prod_{i \leqslant k} \max_{V \subset \mathbb{R}^k,\ dim(V)=i} \{ \min_{x \in V,\ \|x\|=1} \{ \langle B_H^T B x, x \rangle \} \} \ .$$

Then, using $B_H^T B_H = X^T X + Y^T Y$ and the fact that $Y^T Y$ is symmetric and positive gives us

$$\begin{aligned} Det(B'^T B') &= \prod_{i\leqslant k} \max_{V\subset\mathbb{R}^k,\ dim(V)=i} \{ \min_{x\in V,\ \|x\|=1} \{\langle X^T X x, x\rangle + \langle Y^T Y x, x\rangle\}\} \\ &\geqslant \prod_{i\leqslant k} \max_{V\subset\mathbb{R}^k,\ dim(V)=i} \{ \min_{x\in V,\ \|x\|=1} \{\langle X^T X x, x\rangle\}\} \\ &= Det(X^T X)\ . \end{aligned}$$

However, $0 < Det(X^T X)$ because of the Hermite normal form.
And as $Vol(\mathcal{L})^2 = \alpha^{2k}$, we get

$$Det(B'^T B') = \frac{Vol(\mathcal{L})^2}{\alpha^{2k}} = 1\ .$$

This then leads to

$$0 < Det(X^T X) \leqslant Det(B'^T B') = 1\ .$$

However, $Det(X) = \prod_{i\leqslant k} X[i,i]$ and $X[i,i] \in \mathbb{N}^*$.

Meaning that $Det(X) \in \mathbb{N}^*$, and $Det(X) = 1$. Thus,

$$\forall i \leqslant n,\ X[i,i] = 1$$

$$\forall j < i \leqslant n,\ X[i,j] = 0\ .$$

Concluding the fact that $X = I_k$.

A useful property is that I_k and Y are simultaneously diagonalizable.
Calling μ_i the eigenvalues of $Y^T Y$, one can deduct than $\lambda_i = 1 + \mu_i$. And as $Y^T Y$ is symmetric, $\forall i \leqslant k,\ \ \mu_i \geqslant 0$.

Finally,

$$det(B_H^T B_H) = \prod_{i\leqslant k} \lambda_i = \prod_{i\leqslant k} 1 + \mu_i = 1\ .$$

Hence, $\forall i \leqslant k,\ \mu_i = 0$ and so $Y = O_{M_{k,n-k}(R)}$.
Meaning that $\mathcal{L}(B')$ is isomorphic to $\mathbb{Z}^k$ and $\mathcal{L}$ is isomorphic to $\alpha\mathbb{Z}^k$.

□

Lemma 6.2. *Let $\mathcal{L} \subset \alpha\mathbb{Z}^n$, $rg(\mathcal{L}) = k$, such that $k < n$, and let $\mathcal{L}' = \pi_{\mathcal{L}}^{\perp}(\alpha\mathbb{Z}^n)$.*
Then $\lambda_1(\mathcal{L}') \leqslant \alpha$.

Proof of 6.2. Let $\mathcal{L} \subset \alpha\mathbb{Z}^n$, $rg(\mathcal{L}) = k$, such that $k < n$ and let $\mathcal{L}' = \pi_{\mathcal{L}}^{\perp}(\alpha\mathbb{Z}^n)$.
As $k < n,\ \exists j \leqslant n$ such that $e_j \notin \mathcal{L}$ where e_i are the vectors of the canonical basis of $\mathbb{Z}^n$.
Thus $\pi_{\mathcal{L}}^{\perp}(e_j) \in \mathcal{L}'$ and

$$0 < \left\|\pi_{\mathcal{L}}^{\perp}(\alpha e_j)\right\| \leqslant \|\alpha e_j\| = \alpha\ .$$

Finally, $\lambda_1(\mathcal{L}') \leqslant \alpha$.

□

Lemma 6.3. *Let $\mathcal{L} \subset \alpha\mathbb{Z}^n$, $rg(\mathcal{L}) = k$, such that $Vol(\mathcal{L})^2 > \alpha^{2k}$, and let $\mathcal{L}' = \pi_{\mathcal{L}}^{\perp}(\alpha\mathbb{Z}^n)$.*
Then $\lambda_1(\mathcal{L}') \leqslant \alpha\sqrt{1 - \frac{1}{n}}$.

Proof of 6.3. Let $\mathcal{L} \subset \alpha\mathbb{Z}^n$, $rg(\mathcal{L}) = k$, such that $Vol(\mathcal{L})^2 > \alpha^{2k}$, and let $\mathcal{L}' = \pi_{\mathcal{L}}^{\perp}(\alpha\mathbb{Z}^n)$.

Consider B an HKZ reduced basis of $\mathcal{L}$. This means that $\forall i \leqslant n,\ B_{[i,n]}$ is SVP-reduced.
As $Vol(\mathcal{L})^2 > \alpha^{2k}$, $\exists b_i$ in the basis of $\mathcal{L}$ such that $\forall j \leqslant n,\ b_i \neq \alpha e_j$.
Let i_0 be the minimal index in that case. As $\mathcal{L} \subset \alpha\mathbb{Z}^n,\ \|\alpha b_{i_0}\| \geqslant \alpha$.

Hence, as B is HKZ reduced, $\forall\ \alpha e_j \in \mathcal{L}, b_{i_0} \perp e_j$. Thus,

$$b_{i_0} = \sum_{\alpha e_j \notin \mathcal{L}} \alpha v_j e_j \ .$$

Let $v_{j_0} = \max_{\alpha e_j \notin \mathcal{L}} v_j$, This leads to

$$\|b_{i_0}\|_\infty = \alpha v_{j_0}$$
$$\langle b_{i_0}, \alpha e_{j_0} \rangle = \alpha \|b_{i_0}\|_\infty \ .$$

And as $e_{j_0} \notin \mathcal{L}$,

$$\begin{aligned} 0 &< \left\|\pi_{\mathcal{L}}^{\perp}(\alpha e_{j_0})\right\| \\ &\leqslant \left\|\pi_{b_{i_0}}^{\perp}(\alpha e_{j_0})\right\| \\ &= \left\|\alpha e_{j_0} - \frac{\langle b_{i_0}, e_{j_0}\rangle}{\|b_{i_0}\|^2} b_{i_0}\right\| \ . \end{aligned}$$

This means by applying the polar identity on the last line that,

$$\begin{aligned} \left\|\pi_{\mathcal{L}}^{\perp}(\alpha e_{j_0})\right\|^2 &\leqslant \|\alpha e_{j_0}\|^2 + \left\|\frac{\langle b_{i_0}, e_{j_0}\rangle}{\|b_{i_0}\|^2} b_{i_0}\right\|^2 - 2\langle \alpha e_{j_0}, \frac{\langle b_{i_0}, \alpha e_{j_0}\rangle}{\|b_{i_0}\|^2} b_{i_0}\rangle \\ &= \alpha^2 + \frac{\langle b_{i_0}, \alpha e_{j_0}\rangle^2}{\|b_{i_0}\|^2} - 2\frac{\langle b_{i_0}, \alpha e_{j_0}\rangle^2}{\|b_{i_0}\|^2} \\ &= \alpha^2 - \frac{\langle b_{i_0}, \alpha e_{j_0}\rangle^2}{\|b_{i_0}\|^2} \\ &= \alpha^2 (1 - \frac{\|b_{i_0}\|_\infty}{\|b_{i_0}\|}^2) \\ &\leqslant \alpha^2 (1 - \frac{1}{n}) \ . \end{aligned}$$

Thus $\lambda_1(\mathcal{L}') \leqslant \alpha\sqrt{1 - \frac{1}{n}}$.

□

Lemma 6.4. *Let $\mathcal{L} \subset \alpha\mathbb{Z}^n$ $rg(\mathcal{L}) = k$, then $Vol(LLL(\mathcal{L}_1)) \leqslant \alpha^k 2^{\frac{n^2}{2}}$.*

Proof of 6.4. First of all,

$$Vol(\mathcal{L}) = \prod_{i=0}^{2k} \|b_i^*\| \ ,$$

where b_i^* refers to the coefficients of the Gram-Schmidt orthogonalization.

One property of the LLL algorithm is that

$$\forall i,\ 0 \leqslant i < 2k, \|b_i^*\|^2 \leqslant \frac{1}{2}\left\|b_{i+1}^*\right\|^2 \ .$$

Hence,

$$Vol(\mathcal{L}_1) = \prod_{i=1}^{k-1} \|b_i^*\| \leqslant \prod_{i=0}^{k-1} 2^i \alpha \ .$$

Thus leading to

$$Vol(\mathcal{L}_1) \leqslant \alpha^k 2^{k(k+1)} \leqslant \alpha^k 2^{\frac{n^2}{2}} \ .$$

□

Theorem 6.5. *Algorithm 1 terminates with at most $\mathcal{O}(n^3)$ iterations of the loop*

Proof of 6.5. Let us take n an odd number and write it $n = 2k+1$, if n is even, we just add a dimension.

If before the while loop, $Vol(\mathcal{L}(B_{0,k})) < \sqrt{2}\alpha^{2k}$ then the algorithm finishes.

Else, let us consider $B_{[k,2k+1]} = \pi^{\perp}_{\mathcal{L}(B_{0,k})}(B_{k,2k+1})$. Then $B_{[k,2k+1]}$ is a basis of $\pi^{\perp}_{\mathcal{L}(B_{0,k})}(\alpha\mathbb{Z}^n)$. To see this, let us take y a vector in this space. Then there is a vector x in $\alpha\mathbb{Z}^n$ such that $\pi^{\perp}_{\mathcal{L}(B_{0,k})}(x) = y$. As B is a basis of $\alpha\mathbb{Z}^n$, we can write x as $x = \sum_{i=0}^{2k+1} x_i b_i$. This leads to

$$y = \pi_{\mathcal{L}(B_{0,k})}\left(\sum_{i=0}^{2k+1} x_i b_i\right) = \sum_{i=0}^{2k+1} x_i \pi_{\mathcal{L}(B_{0,k})}(b_i) = \sum_{i=k}^{2k+1} x_i \pi_{\mathcal{L}(B_{0,k})}(b_i) \ .$$

Finally, we get that $B_{[k,2k+1]}$ is a basis of $\pi^{\perp}_{\mathcal{L}(B_{0,k})}(\alpha\mathbb{Z}^n)$. This allows to apply 6.3 on $\mathcal{L}(B_{0,k})$, which shows that $\lambda_1(\mathcal{L}(B_{[k,2k+1]})) \leqslant \alpha\sqrt{1-\frac{1}{n}}$.

Thus, at the beginning of the loop $\|b_k^*\| \leqslant \alpha\sqrt{1-\frac{1}{n}}$, and b_k^* is orthogonal to $\mathcal{L}(B_{0,k})$. If we denote $Vol(\mathcal{L}(B_{0,k}))$ at the beginning of the loop by V_1, it leads to

$$Vol(\mathcal{L}(B_{0,k+1})) = \|b_k^*\| V_1 \ .$$

Furthermore, Dual-SVP reduced means that the basis of the dual is SVP reduced. To make things easier, we will consider the reversed dual basis, meaning that we will have d_k as the shortest vector. One thing to note is that the Gram-Schmidt orthogonalisation process gives

$$b_k^* = \frac{d_k}{\|d_k\|^2} \ .$$

We now want to apply 6.2 to get that $\lambda_1(\mathcal{L}(B_{0,k+1})^\vee) \leqslant \frac{1}{\alpha}$. However, for this we need to prove that $\mathcal{L}(B_{0,k+1})^\vee$ is the projection of a sublattice of $\frac{1}{\alpha}\mathbb{Z}^n$.

One can prove that a projection in the primal lattice is a section of the dual lattice. Formally, when B is a basis of the primal and D is a reversed dual basis $\{d_{2k}, ..., d_0\}$, it can be formulated as

$$\mathcal{L}(B_{[l,...,r]})^\vee = \mathcal{L}(\pi^{\perp}_{d_{2k},...,d_{r+1}}(\{d_r, ..., d_l\})) \ .$$

Thus $(\mathcal{L}(B_{0,k+1}))^\vee = \mathcal{L}(\pi^{\perp}_{d_{2k},...,d_{k+1}}(\{d_k, ..., d_0\}))$. Which means with $\bar{\mathcal{L}} = \mathcal{L}(\{d_{2k}, ..., d_{k+1}\})$ that

$$\mathcal{L}(B_{0,k+1})^\vee = \pi^{\perp}_{\bar{\mathcal{L}}}(\frac{1}{\alpha}\mathbb{Z}^n) \ ,$$

with $\bar{\mathcal{L}}$ a sublattice of $\frac{1}{\alpha}\mathbb{Z}^n$.

Thus, when we write $\widetilde{D}, \widetilde{B}$ the new basis, $\left\|\widetilde{d_k}\right\| \leqslant \frac{1}{\alpha}$ which leads to $\left\|\widetilde{b_k^*}\right\| \geqslant \alpha$.

The SVP reduction consists of taking a shortest vector of the lattice and then complete the basis. As the volume is preserved during an SVP reduction, $Vol(\mathcal{L}(B_{0,k+1}))$ has not changed. Hence,

$$Vol(\mathcal{L}(B_{0,k+1})) = \|b_k^*\| V_1 = \left\|\widetilde{b_k^*}\right\| \widetilde{V_1} \ ,$$

with $\widetilde{V_1}$ referring to the new volume of $\mathcal{L}(B_{0,k})$. Thus

$$\widetilde{V_1} = V_1 \frac{\|b_k^*\|}{\left\|\widetilde{b_k^*}\right\|} \leqslant V_1 \frac{\alpha\sqrt{1-\frac{1}{n}}}{\alpha} = V_1\sqrt{1-\frac{1}{n}} \ .$$

This shows that each iteration of the loop decreases the volume by a factor at least $\sqrt{1-\frac{1}{n}}$.

After the application of the LLL algorithm, as proved in annex, $Vol(\mathcal{L}(B_{0,k})) \leqslant \alpha^k 2^{\frac{n^2}{2}}$ and the while loop finishes when $Vol(\mathcal{L}_1) \leqslant \alpha^k$. Thus the number of iterations of the while loop is in $\mathcal{O}(\frac{(\log(2^{\frac{n^2}{2}})}{\log(\sqrt{1-\frac{1}{n}})}) = \mathcal{O}(\frac{n^2 \frac{\log(2)}{2}}{\frac{1}{2}\log(1-\frac{1}{n})}) = \mathcal{O}(n^3)$.

□

Theorem 6.6. *Algorithm 1 returns an orthonormal basis of a rotation of $\mathbb{Z}^n$.*

Proof of 6.6. Let $\mathcal{L}$ be a lattice, such that $\mathcal{L} \subset \alpha\mathbb{Z}^n$.

At the end of the While loop, $Vol(\mathcal{L}(B_{0,k})) < \sqrt{2}\alpha^{2k}$, thus

$$0 < \frac{Vol(\mathcal{L}(B_{0,k}))^2}{\alpha^{2k}} < 2 \ .$$

However, by 6.1, as

$$\mathcal{L}(B_{0,k}) \subset \alpha\mathbb{Z}^n \qquad \frac{Vol(\mathcal{L}(B_{0,k}))^2}{\alpha^{2k}} \in \mathbb{N}^* \ ,$$

this means that

$$Vol(\mathcal{L}(B_{0,k}))^2 = \alpha^{2k} \ .$$

Using 6.1 once again, $\mathcal{L}(B_{0,k})$ is isomorphic to $\alpha\mathbb{Z}^k$. Finally, applying the HKZ algorithm to $B_{0,k}$ gives us the canonical basis of $\alpha\mathbb{Z}^k$. This means that $\mathcal{L}(B_{[k,2k+1]}))$ is now isomorphic to $\alpha\mathbb{Z}^{k+1}$. And by applying HKZ on it too, we obtain a canonical basis of $\mathcal{L}$.

Thus Algorithm 1 returns B a basis of $\mathcal{L}$ such that $\frac{1}{\alpha}B$ is orthonormal.

□

6.2 Visualisation of volume reduction

The 2 blocks volume reduction

The volume reduction phase can be a bit tricky to fully understand. The following illustrations aim at making it more visual. The exact scale or value is not what we focus on, as well as the slope. We only aim at giving a rough intuition of how this volume reduction works.

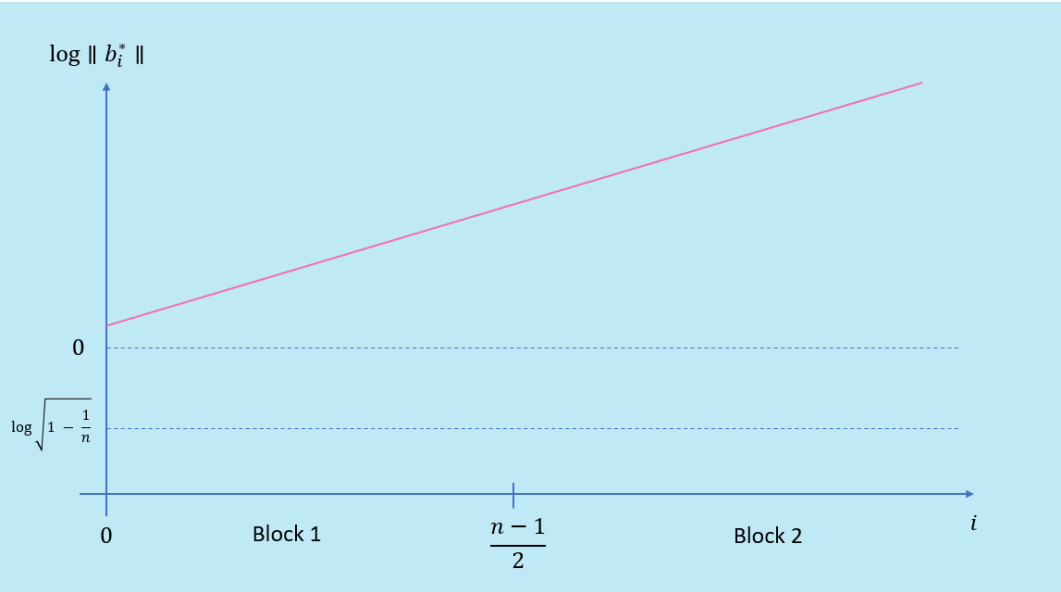


Figure 8: After LLL

After the LLL phase, the basis vectors have bounded length and it is the same for the Gram-Schmidt vectors. The reason for the positive slope comes from Lovasz condition in the LLL algorithm

$$\|b_{i+1}^*\|^2 \geqslant (\frac{3}{4} - \mu_{i,i+1}^2)\|b_i^*\|^2 \geqslant \frac{1}{2}\|b_i^*\|^2$$

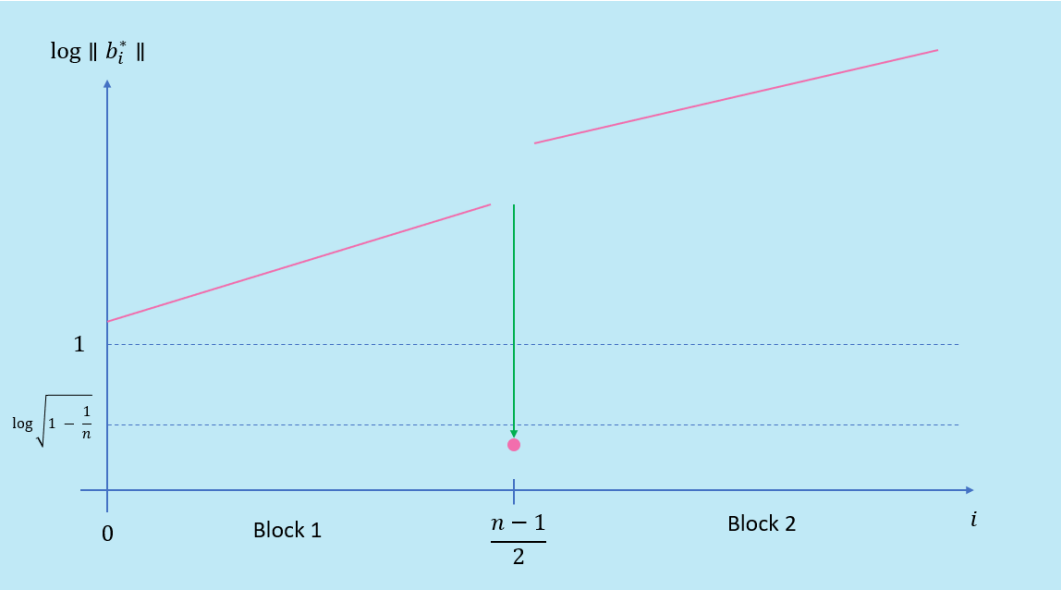


Figure 9: After SVP

The SVP reduction of the last $\frac{n+1}{2}$ vectors consists of getting b a shortest vector of the lattice generated by $B_{[\frac{n-1}{2},n]}$, lifting it in $B_{\frac{n-1}{2},n}$. Then we consider the basis we had : $B_{\frac{n-1}{2},n}$ and the new vector b. And we transform this generating set into a basis with first vector b. For this we can use LLL on $b \bigcup B_{\frac{n-1}{2},n}$.

We have that $\log(Vol(\mathcal{L}(B_{\frac{n-1}{2},n}))) = \sum_{\frac{n-1}{2} \leqslant i \leqslant n} \log(\|b_i^*\|)$. As this transformation does not change the volume of the block, the area under the curve for block 2 remains the same. Thus the $\frac{n-1}{2}$ last vectors become larger.

The DSVP reduction consists of a SVP reduction of the dual lattice. By applying it, we get a large $\|b_{\frac{n-1}{2}}^*\|$. Once again, the volume remains the same for block $B_{0,\frac{n+1}{2}}$, which means that the volume of the block $B_{0,\frac{n-1}{2}}$ decreases.

About the volume reduction factor with more than 4 blocks

It appears that until a certain threshold the volume reduction factor behaves as we would expect it : meaning that it gets bigger when the volume decreases. However, at around half of the iterations it starts decreasing. We do not know why it happens but this behavior appeared in every tests with 4 blocks. It should also be noted that it also appeared in tests with 3 blocks.

The second graph represent the volume of the blocks, the blue curve correspond to the block containing the k first vectors. It illustrates this surprising behavior.

6.3 Proofs of better complexity

Lemma 6.7. *Let $\mathcal{L}$ be a random lattice, then we have*

$$\mathbb{E}(r) = \left(\frac{1}{V_{k+1}(1)Vol(\mathcal{L}(B_{0,k}))}\right)^{\frac{1}{k+1}(2-\frac{1}{k+1})},$$

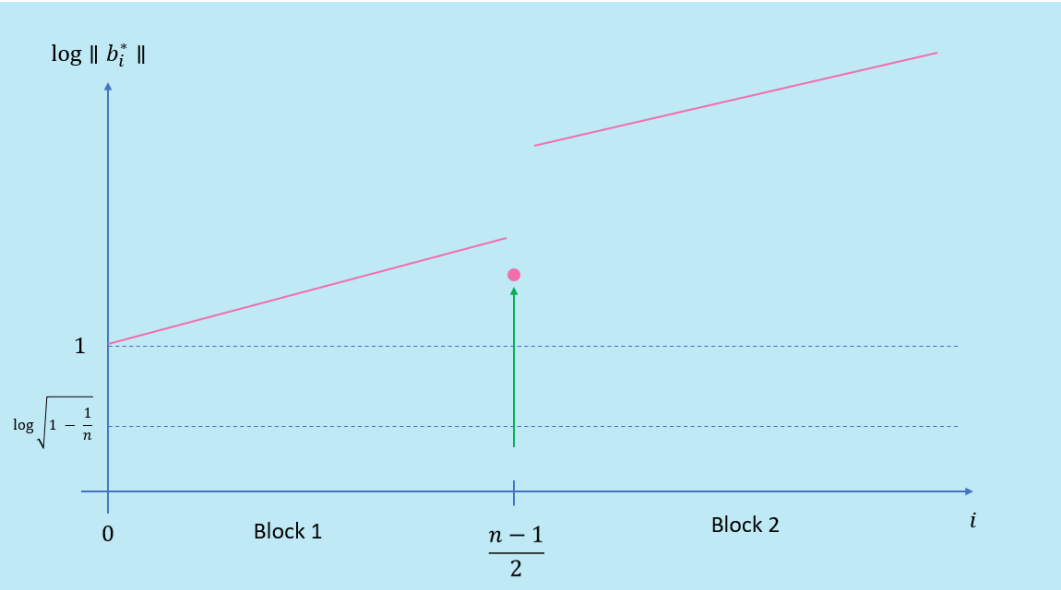


Figure 10: After DSVP

where $V_n(1)$ is the volume of the n-dimensional ball of radius 1.

Proof. During a single iteration of the while loop, we perform an SVP reduction and an DSVP reduction in dimension $k+1$.

Using the Gaussian heuristic on $B_{[k,2k+1]}$, we have

$$\mathbb{E}(\lambda_1(\mathcal{L}(B_{[k,2k+1]}))) = \sqrt{\frac{k+1}{2\pi e}} Vol(\mathcal{L}(B_{[k,2k+1]}))^{\frac{1}{k+1}} = \sqrt{\frac{k+1}{2\pi e}} \frac{1}{Vol(\mathcal{L}(B_{0,k}))^{\frac{1}{k+1}}} .$$

In the same way, using it for the DSVP gives asymptotically

$$\mathbb{E}(\lambda_1(\mathcal{L}(B_{[0,k+1]})^\vee)) = \sqrt{\frac{k+1}{2\pi e}} Vol(\mathcal{L}(B_{0,k+1})^\vee)^{\frac{1}{k+1}} = \sqrt{\frac{k+1}{2\pi e}} \left(\frac{1}{Vol(\mathcal{L}(B_{0,k}))\mathbb{E}(\lambda_1(\mathcal{L}(B_{[k,2k+1]})))} \right)^{\frac{1}{k+1}} .$$

We denote as V_{old} and V_{new} the volume $Vol(\mathcal{L}(B_{0,k}))$ before and after the loop iteration. The formula for the volume reduction can be formulated as :

$$\begin{aligned}
\mathbb{E}(r) &= \mathbb{E}\left(\frac{V_{new}}{V_{old}}\right) \\
&= \mathbb{E}(\lambda_1(\mathcal{L}(B_{[k,2k+1]})))\mathbb{E}(\lambda_1(\mathcal{L}(B_{[0,k+1]})^\vee)) \\
&= \mathbb{E}(\lambda_1(\mathcal{L}(B_{[k,2k+1]})))^{2-\frac{1}{k+1}}
\end{aligned}$$

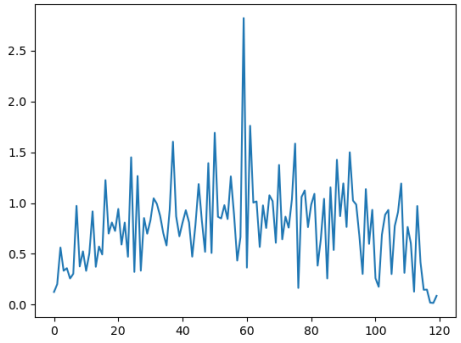
This ratio is still the same even when we consider the exact volume of the n-dimensional ball of radius 1 instead of its asymptotic volume.

This gives :

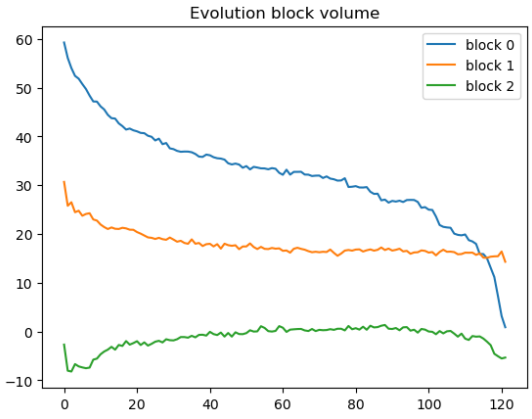
$$\mathbb{E}(r) = \left(\frac{1}{V_{k+1}(1) Vol(\mathcal{L}(B_{0,k}))} \right)^{\frac{1}{k+1}(2-\frac{1}{k+1})} .$$

□

Lemma 6.8. *The volume reduction loop is a round of the slide algorithm.*



(a) Evolution of the volume reduction factor during the iterations of the loop for 4 blocks in dimension 120



(b) Volume of the blocks

Figure 11: volume reduction factor of first block and volume evolution with 4 blocks in dimension 120

Proof. The while loop of the algorithm is in fact the loop from the slide reduction applied to a slightly modified basis. Let B be a basis of $\mathbb{Z}^n$, we will add an artificial dimension and extend those vectors with a 0. Then consider $\bar{b} = (0, ..., 0, \alpha)$, we get that $\bar{b}$ is orthogonal to $Span(B)$. Furthermore, if $\alpha < 1$, then we get that $\lambda_1(\mathcal{L}([\bar{b}|B_{0,k}])) = \alpha$. This means that $[\bar{b}|B_1]$ is SVP-reduced.

We now apply the slide reduction algorithm to the new basis, with blocks of size $\frac{n}{2} = k + 1$.
First we SVP-reduce $[\bar{b}|B_{0,k}]$, which does not change anything. Then we SVP-reduce $B_{[k,2k+1]}$, just as in the regular volume reduction loop. After this, we DSVP reduce $B_{0,k+1}$ which is again what we are doing in the algorithm. This means that our while loop is in fact a round of the slide reduction on $[\bar{b}|B]$.

Finally, as $\bar{b}$ is orthogonal to $B_{0,k}$ and does not change in the algorithm, we get that $Vol(\mathcal{L}(B_{0,k})) = \frac{1}{\alpha} Vol(\mathcal{L}([\bar{b}|B_{0,k}]))$. This means that the volume analysis from the slide algorithm holds on $B_{0,k}$. Thus, we can actually use the proven bounds from the slide reduction. □

Theorem 6.9. *The volume reduction phases finishes in at most $\mathcal{O}(n^2 \log(n))$ iterations.*

Proof. First we need to prove that we can actually reach the volume $2^{\frac{n}{2}} \sqrt{\gamma_{\frac{n}{2}}}^{\frac{n}{2}}$ in $\prime(n \log(n))$ iterations. To prove this we will use Corollary 1 from [Wal20] section 3.2,with (using the notations from the paper) $d = \frac{n+1}{2}$, $k = 1$ as we have regular SVP, which implies $\alpha = \sqrt{\gamma_{k+1}}$ and we chose $\epsilon = 1$.

Using this corollary, after

$$l \geqslant \frac{n^2}{4\frac{n-3}{2}} \log\left(\frac{\frac{n^2}{n-1} + \frac{n^3}{4(\frac{n-1}{2})^3}\log(\alpha))}{\epsilon}\right) = \mathcal{O}(n \log(n + \log(n))) = \mathcal{O}(n \log(n))$$

iterations, we have $Vol(B_{0,k+1}) \leqslant 2^{\frac{n}{2}} \sqrt{\gamma_{\frac{n}{2}}}^{\frac{n}{2}} = \mathcal{O}(2^{n \log(n)})$.

Now that we reached this volume, by applying the worst case bound, we can go from $\mathcal{O}(2^{n \log(n)})$ to 1. It will takes us at most $\frac{\log(2^{n \log(n)})}{\log(\frac{1}{\sqrt{1-1/n}})} = \mathcal{O}(n^2 \log(n))$ iterations.

Thus, the total number of iterations needed to go to a volume of 1 is at most $\mathcal{O}(n^2 \log(n))$. □

Lemma 6.10. *Let W be a k-dimensional space with $W \subset \mathbb{R}^n$. Then $\min_{i \leqslant n} \|\pi_W(e_i)\|^2 \leqslant \frac{k}{n}$.*

Proof. Let us prove this by contradiction. Let us assume that there is a subspace of $\mathbb{R}^n$ W such that for all $i \leqslant n$, $\|\pi_W(e_i)\|^2 > \frac{k}{n}$.

We can take an orthonormal basis for W using the Gram-Schmidt orthogonalisation process. And we can complete this set into an orthonormal basis of $\mathbb{R}^n$. This gives us a matrix $Q \in \mathcal{O}_n(\mathbb{R})$ and $J \subset \{1, ..., n\}$ of cardinal k, such that W is generated by the vectors $(b_j)_{j\in J}$.

Let $i \leqslant n$, using the properties of Q, we now have

$$\|\pi_W(e_i)\|^2 = \|\sum_{j\in J} \frac{\langle e_i, q_j\rangle}{\|q_j\|^2} q_j\|^2 = \sum_{j\in J} q_{i,j}^2 \ .$$

By hypothesis on W, we get $\sum_{1\leqslant i\leqslant n} \|\pi_W(e_i)\|^2 > \sum_{1\leqslant i\leqslant n} \frac{k}{n} > k$.
However, using the fact that Q forms an orthonormal basis, we can deduce

$$\sum_{1\leqslant i\leqslant n} \|\pi_W(e_i)\|^2 = \sum_{1\leqslant i\leqslant n} \sum_{j\in J} q_{i,j}^2 = \sum_{j\in J} 1 = k \ .$$

This is absurd, which means that such a W cannot exist. Thus, for every k-dimensional subspace W, there is a unit vector e_i such that $\|\pi_W(e_i)\|^2 \leqslant \frac{k}{n}$.

□

Theorem 6.11. *Let W be a k-dimensional space with $W \subset \mathbb{R}^n$, $W \neq \mathbb{R}^k$. Let $S = \{e_i \in W\}$ and $S^\perp = \{e_i \in W^\perp\}$ where $W^\perp$ is the space orthogonal to W. Then $0 < \min_{e_i \notin W^\perp} \|\pi_W(e_i)\|^2 \leqslant \frac{k-|S|}{n-|S|-|S^\perp|}$.*

Proof. Let us write W the same way as in the proof of 6.10. We can thus write W as

$$W = \begin{bmatrix} I_{|S|} & 0_{|S|,k-|S|} \\ 0_{n-|S|-|S^\perp|,|S|} & * \\ 0_{|S^\perp|,|S|} & 0_{|S^\perp|,k-|S|} \end{bmatrix} \ ,$$

which helps us show that

$$\sum_{i=|S|+1}^{n-|S^\perp|} \sum_{j=|S|+1}^{k} \|\pi_W(e_i)\|^2 = k - |S| \ .$$

Thus, we have $0 < \min_{e_i \notin W^\perp} \|\pi_W(e_i)\|^2 \leqslant \frac{k-|S|}{n-|S|-|S^\perp|}$.

□

Theorem 6.12. *conjecture ! not formally proven*
Let r be the volume reduction factor, then $\mathbb{E}(r) \leqslant \frac{1}{2}$.

Proof. Idea behind the conjecture
Let us consider a basis B of $\mathbb{Z}^n$ on which we will apply the algorithm.

From the previous analysis, we get that

$$\mathbb{E}(r) = \mathbb{E}(\lambda_1(\mathcal{L}(B_{[k,2k+1]})))\mathbb{E}(\lambda_1(\mathcal{L}(B_{0,k+1})^\vee)) = \mathbb{E}(\lambda_1(\pi^\perp_{B_{0,k}}(\mathbb{Z}^n)))\mathbb{E}(\lambda_1(\pi^\perp_{B^\vee_{[k,2k+1]}}(\mathbb{Z}^n))) \ .$$

We now want to bound $\mathbb{E}(\lambda_1(\pi^\perp_{B_{0,k}}(\mathbb{Z}^n)))$.

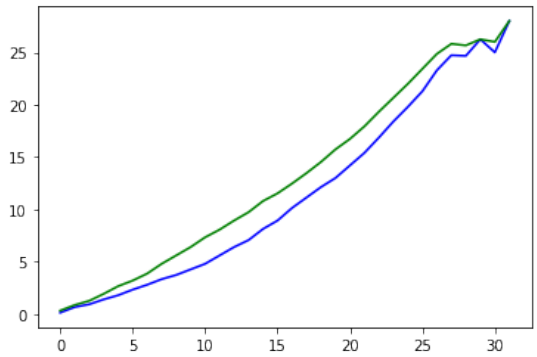
Let us denote by $S_{i,j} = \{e_i \in Span(B_{i,j})\}$, and by $S_{[i,j]} = \{e_i \in Span(B_{[i,j]})\}$.

Using 6.11, we get that when $\mathcal{L}(B_{0,k})$ is not isomorphic to $\mathbb{Z}^k$,

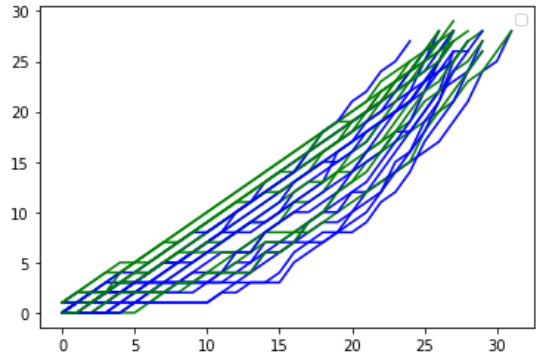
$$\min_{e_i \notin S_{0,k}} \|\pi^\perp_{B_{0,k}}(e_i)\|^2 \leqslant \frac{k+1-|S_{[k,2k+1]}|}{n-|S_{[k,2k+1]}|-|S_{0,k}|} \ .$$

However, by property of the dual, $S_{[k,2k+1]} = \{e_i \in Span(B_{[k,2k+1]})\} = \{e_i \in Span(B_{[k,2k+1]}\vee)\}$.

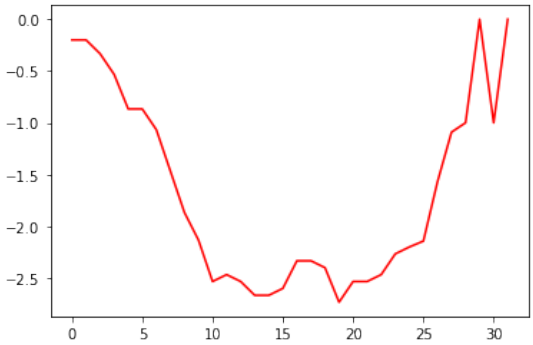
And $\mathcal{L}(B_{[k,2k+1]})\vee$ is a sublattice of $\mathbb{Z}^n$ because a projection in a primal is a section in the dual. And it has the same volume as $\mathcal{L}(B_{0,k})$. Furthermore, the transformations are the same on both



(a) Mean of $|S_{0,k}|$ and $|S_{[k,2k+1]}|$ during the loop iterations



(b) Distribution of $|S_{0,k}|$ and $|S_{[k,2k+1]}|$ during the loop iterations



(c) Evolution of $|S_{0,k}| - |S_{[k,2k+1]}|$ during the loop iterations

Figure 12: Evolution of $|S_{0,k}|$ and $|S_{[k,2k+1]}|$ during the loop iterations

lattices. In fact the DSVP step in the algorithm is the SVP step when we consider the reversed dual basis and the SVP step is the DSVP step. This shows that those two lattices are going through the exact same transformations, the only difference being the dimension : one is of dimension k and the other $k+1$.

Hence, $|S_{0,k}|$ and $|S_{[k,2k+1]}|$ roughly follow the same probability law.

This give that $\mathbb{E}(|S_{0,k}|) \leqslant \mathbb{E}(|S_{[k,2k+1]}|) \leqslant \mathbb{E}(|S_{0,k}|) + 1$.

Thus

$$\mathbb{E}(\min_{e_i \notin S_{0,k}} \|\pi^{\perp}_{B_{0,k}}(e_i)\|^2) \leqslant \frac{\frac{n+1}{2} - \mathbb{E}(|S_{0,k}|)}{n+1-2\mathbb{E}(|S_{0,k}|)} = \frac{1}{2} .$$

Finally, we use that $\mathbb{E}(\lambda_1(\pi^{\perp}_{B_{0,k}}(\mathbb{Z}^n))^2) \leqslant \mathbb{E}(\min_{e_i \notin S_{0,k}} \|\pi^{\perp}_{B_{0,k}}(e_i)\|^2)$.
And by using the same analysis to bound $\mathbb{E}(\lambda_1(\pi^{\perp}_{B^{\vee}_{[k,2k+1]}}(\mathbb{Z}^n)))$, we obtain $\mathbb{E}(r) \leqslant \frac{1}{2}$.

□

Finally, I did experiments to study the evolution of $|S_{0,k}|$ and $|S_{[k,2k+1]}|$ during the iterations of the algorithm. As it can be quite long I did not launched it on big dimension. Figure 12 represents the evolution of those parameters, the blue curve corresponds to $|S_{0,k}|$ and the green one to $|S_{[k,2k+1]}|$, they have been generated in dimension 61, this being relatively small. It seems that $|S_{0,k}| \leqslant |S_{[k,2k+1]}|$, and the curvature is not the same, thus it might get further from one another in higher dimension. However, further experiments in higher dimension are needed to see if the conjecture holds.

6.4 Code algorithm on p blocks

Algorithm 3 is the variant of the original algorithm for p blocks. Just as before, by adding one vector at the beginning, it is a slide reduction with blocks of size $k+1$.

Algorithm 3 Algorithm on p blocks

Require: B basis of $\mathcal{L}$, $\mathcal{L}$ rotation of $\mathbb{Z}^n$, $n+1 = p(k+1)$.
Ensure: B orthonormal basis of $\mathcal{L}$
  $B \leftarrow LLL(B)$
  while $Vol(B_{0,k+1}) > 1$ **do**
    for $0 \leqslant i \leqslant p-2$ **do**
      $B_{[k+i(k+1),k+(i+1)(k+1)]} \leftarrow$ Primal-SVP Reduce $(B_{[k+i(k+1),k+(i+1)(k+1)]})$
    end for
    for $0 \leqslant i \leqslant p-2$ **do**
      $B_{[i(k+1),(i+1)(k+1)]} \leftarrow$ Dual-SVP Reduce $(B_{[i(k+1),(i+1)(k+1)]})$
    end for
  end while
  $B_{0,k} \leftarrow$ Primal-HKZ Reduce $(B_{0,k})$
  for $1 \leqslant i \leqslant p-1$ **do**
    $B_{[k+(i-1)(k+1),k+i(k+1)]} \leftarrow$ Primal-HKZ Reduce $(B_{[k+(i-1)(k+1),k+i(k+1)]})$
  end for
  return B

We can represent this algorithms in the same way as for the slide reduction. The $\bar{b}$ illustrates the vector we would have to add in order to get a real slide reduction, by following the method from 6.8.

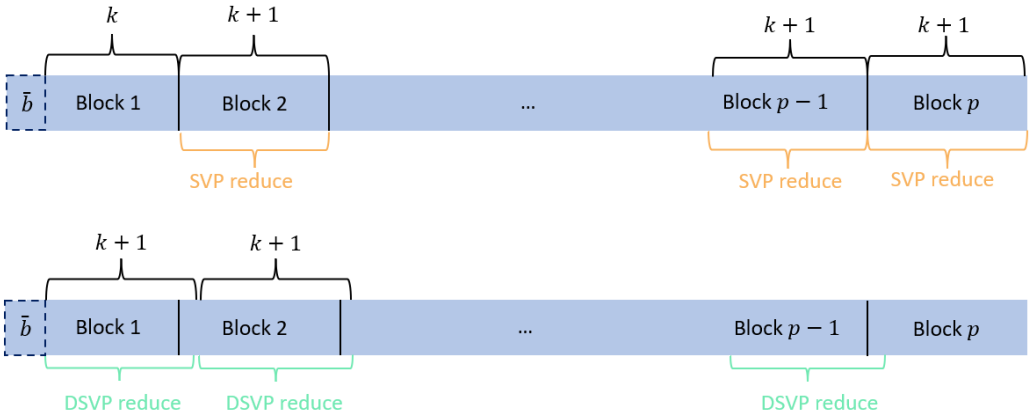


Figure 13: Schematic representation of algorithm 3

7 Background annex

7.1 Lattice algorithms

As most lattice algorithms focus on SVP, the description of lattice algorithms will be focused on SVP.

Exact algorithms

These algorithms return a shortest vector of the lattice but at an exponential time cost. There are two commonly used techniques to achieve this task : enumeration and sieving. Intuitively, enumeration algorithms list all the extremely short vectors of the lattice, hence the exponential time, and return one shortest [Kan83]. Whereas sieving algorithms generate an exponential number of vectors and find

clever way to pair them, take their means to reduce the size of the vectors, and return a shortest vector at the end with high probability. Currently, in order to solve SVP in dimension n, the best proved running time is in $\mathcal{O}(2^n)$ and uses sieving [ADRS15]. Whereas, the best running time for enumeration is in $\mathcal{O}(n^{\frac{n}{2e}})$ [HS07].

Approximation algorithms / Lattice reduction algorithms

The most commonly used approximation algorithms return a basis whose first vector is relatively short. These algorithms are extremely faster than the ones above but at the cost of a huge loss in the length on the shortest vector. As said before the LLL algorithm is polynomial but the bound on the first vector is $\|b_1\| \leqslant 2^n \lambda_1(\mathcal{L})$, which is far from Minkowski's bound. We can note that the fact that polynomial approximation algorithms can only have an exponential approximation factor is really rare.

More complex algorithms such as BKZ or SDBKZ use exact SVP algorithms as a subroutine. In fact they call the exact algorithms in a lower dimension in order to reduce the cost but lose in the quality of the returned vectors. For example, the bound on SDBKZ [MW15] when applying the exact SVP in dimension k is $\|b_1\| \leqslant \sqrt{\gamma_k}^{\frac{n-1}{k-1}} \lambda_1(\mathcal{L})$, achieved in $\mathcal{O}(Poly(n)2^k)$. There are also algorithms aiming at better approximation factor without returning a good basis. For example, there is an algorithm which solves $\sqrt{n}$-SVP in $\mathcal{O}(2^{\frac{n}{2}})$ [ALS20].