Rankin's Constant and Blockwise Lattice Reduction

Nicolas Gama[1], Nick Howgrave-Graham[2], Henrik Koy[3], and Phong Q. Nguyen[4]

[1] École normale supérieure, DI, 45 rue d'Ulm, 75005 Paris, France
nicolas.gama@ens.fr
[2] NTRU Cryptosystems, Burlington, MA, USA
NHowgraveGraham@ntru.com
[3] Deutsche Bank AG, Postfach, 60262 Frankfurt am Main, Germany
henrik.koy@db.com
[4] CNRS/École normale supérieure, DI, 45 rue d'Ulm, 75005 Paris, France
http://www.di.ens.fr/~pnguyen

Abstract. Lattice reduction is a hard problem of interest to both public-key cryptography and cryptanalysis. Despite its importance, extremely few algorithms are known. The best algorithm known in high dimension is due to Schnorr, proposed in 1987 as a block generalization of the famous LLL algorithm. This paper deals with Schnorr's algorithm and potential improvements. We prove that Schnorr's algorithm outputs better bases than what was previously known: namely, we decrease all former bounds on Schnorr's approximation factors to their $(\ln 2)$-th power. On the other hand, we also show that the output quality may have intrinsic limitations, even if an improved reduction strategy was used for each block, thereby strengthening recent results by Ajtai. This is done by making a connection between Schnorr's algorithm and a mathematical constant introduced by Rankin more than 50 years ago as a generalization of Hermite's constant. Rankin's constant leads us to introduce the so-called smallest volume problem, a new lattice problem which generalizes the shortest vector problem, and which has applications to blockwise lattice reduction generalizing LLL and Schnorr's algorithm, possibly improving their output quality. Schnorr's algorithm is actually based on an approximation algorithm for the smallest volume problem in low dimension. We obtain a slight improvement over Schnorr's algorithm by presenting a cheaper approximation algorithm for the smallest volume problem, which we call transference reduction.

1 Introduction

Lattices are discrete subgroups of $\mathbb{R}^m$. A lattice L can be represented by a basis, that is, a set of $n \leq m$ linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^m$ such that L is equal to the set $L(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^n x_i \mathbf{b}_i, x_i \in \mathbb{Z}\}$ of all integer linear combinations of the $\mathbf{b}_i$'s. The integer n is the dimension of the lattice L. A lattice has infinitely many bases (except in trivial dimension ≤ 1), but some are more useful than others. The goal of *lattice reduction* is to find interesting lattice bases, such as bases consisting of reasonably short and almost orthogonal vectors: it can intuitively be viewed as a vectorial generalisation of gcd computations. Finding good reduced bases has proved invaluable in many fields of computer science and mathematics (see [10,7]), particularly in cryptology (see [17,21]).

Lattice reduction is one of the few potentially hard problems currently in use in public-key cryptography (see [21,17] for surveys on lattice-based cryptosystems). But the problem is perhaps more well-known in cryptology for its major applications in public-key cryptanalysis (see [21]): knapsack cryptosystems [22], RSA in special settings [9,5,4], DSA signatures in special settings [12,19], *etc.* Nevertheless, there are very few lattice reduction algorithms, and most of the (recent) theoretical results focus on complexity aspects (see [17]).

The first lattice reduction algorithm in arbitrary dimension is due to Hermite [11], and is based on Lagrange's two-dimensional algorithm [13] (often wrongly attributed to Gauss). It was introduced to show the existence of Hermite's constant (which guarantees the existence of short lattice vectors), as well as proving the existence of lattice bases with bounded orthogonality defect. The celebrated Lenstra-Lenstra-Lovász algorithm [14] (LLL) can be viewed as a relaxed variant of Hermite's algorithm, in order to guarantee a polynomial-time complexity. There are faster variants of LLL based on floating-point arithmetic (see [20,25]), but none improves the output quality of LLL, which is tightly connected to Hermite's historical (exponential) upper bound on his constant. The only (high-dimensional) polynomial-time reduction algorithm known with better output quality than LLL is due to Schnorr [24]. From a theoretical point of view, only one improvement to Schnorr's block-reduction algorithm has been found since [24]: by plugging the probabilistic AKS sieving algorithm [2], one may increase the blocksize $k = \log n / \log\log n$ to $k = \log n$ and keep polynomial-time complexity, which leads to (slightly) better output quality. Curiously, in practice, one does not use Schnorr's algorithm when LLL turns out to be insufficient: rather, one applies the so-called BKZ variants [26,27] of Schnorr's algorithm, whose complexity is unknown.

Our Results. We focus on the best high-dimensional lattice reduction algorithm known (Schnorr's semi block-$2k$ algorithm [24]) and potential improvements. Despite its importance, Schnorr's algorithm is not described in any survey or textbook, perhaps due to the technicality of the subject. We first revisit Schnorr's algorithm by rewriting it as a natural generalization of LLL. This enables to analyze both the running time and the output quality of Schnorr's algorithm in much the same way as with LLL. It also leads us to reconsider a certain constant β_k introduced by Schnorr [24], which is tightly related to the output quality of his semi block-$2k$ algorithm. Roughly speaking, β_k plays a role similar to Hermite's constant $\gamma_2 = \sqrt{4/3}$ in LLL.

We improve the best upper bound known for β_k: we show that essentially, $\beta_k \lessapprox 0.38 \times k^{2 \ln 2} \approx 0.38 \times k^{1.39}$, while the former upper bound [24] was $4k^2$. This leads to better bounds on the output quality of Schnorr's algorithm: for instance, the approximation factor $(6k)^{n/k}$ given in [24] can be decreased to its $(\ln 2)$-th power (note that $\ln 2 \approx 0.69$). On the other hand, Ajtai [1] recently proved that there exists $\varepsilon > 0$ such that $\beta_k \geq k^{\varepsilon}$, but no explicit value of ε was known. We establish the lower bound $\beta_k \geq k/12$, and our method is completely different from Ajtai's. Indeed, we use a connection between β_k and a mathematical constant introduced by Rankin [23] more than 50 years ago as a generalization of Hermite's constant.

Besides, Rankin's constant is naturally related to a potential improvement of Schnorr's algorithm, which we call block-Rankin reduction, and which may lead to better approximation factors. Roughly speaking, the new algorithm would still follow the LLL framework like Schnorr's algorithm, but instead of using Hermite-Korkine-Zolotarev (HKZ) reduction of $2k$-blocks, it would try to solve the so-called *smallest volume problem* in $2k$-blocks, which is a novel generalization of the shortest vector problem. Here, Rankin's constant plays a role similar to β_k in Schnorr's algorithm. But our lower bound on β_k actually follows from a lower bound on Rankin's constant, which suggests that there are intrinsic limitations to the quality of block-Rankin reduction. However, while Ajtai presented in [1] "worst cases" of Schnorr's algorithm which essentially matched the bounds on the output quality, this is an open question for block-Rankin reduction: perhaps the algorithm may perform significantly better than what is proved, even in the worst case. Finally, we make a preliminary study of the smallest volume problem. In particular, we show that HKZ-reduction does not necessarily solve the problem, which suggests that block-Rankin reduction might be stronger than Schnorr's semi block reduction. We also present an exact solution of the smallest volume problem in dimension 4, as well as an approximation algorithm for the smallest volume problem in dimension $2k$, which we call *transference reduction*. Because transference reduction is cheaper than the $2k$-dimensional HKZ-reduction used by Schnorr's algorithm, we obtain a slight improvement over Schnorr's algorithm: for a similar cost, we can increase the blocksize and therefore obtain better quality.

Road map. The paper is organized as follows. In Section 2, we provide necessary background on lattice reduction. In Section 3, we revisit Schnorr's algorithm and explain its main ideas. Section 4 deals with Rankin's constant and its connection with Schnorr's algorithm. In Section 5, we study the smallest volume problem, discuss its application to the so-called block-Rankin reduction, and present transference reduction.

Acknowledgements. Part of this work, as well as a visit of the second author to the ENS, were supported by the Commission of the European Communities through the IST program under contract IST-2002-507932 ECRYPT. We would like to thank Renaud Coulangeon for useful conversations.

2 Background

Let $\|.\|$ and $\langle ., . \rangle$ be the Euclidean norm and inner product of $\mathbb{R}^m$. Vectors will be written in bold, and we will use row-representation for matrices. For a matrix M whose name is a capital letter, we will usually denote its coefficients by $m_{i,j}$:

if the name is a Greek letter like μ, we will keep the same symbol for both the matrix and its coefficients. The notation $\lceil x \rfloor$ denotes a closest integer to x.

2.1 Lattices

We refer to the survey [21] for a bibliography on lattices. In this paper, by the term lattice, we mean a discrete subgroup of some $\mathbb{R}^m$. The simplest lattice is $\mathbb{Z}^n$, and for any linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_n$, the set $L(\mathbf{b}_1, \dots, \mathbf{b}_n) = \{\sum_{i=1}^n m_i \mathbf{b}_i \mid m_i \in \mathbb{Z}\}$ is a lattice. It turns out that in any lattice L, not just $\mathbb{Z}^n$, there must exist linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_n \in L$ such that $L = L(\mathbf{b}_1, \dots, \mathbf{b}_n)$. Any such n-tuple of vectors $\mathbf{b}_1, \dots, \mathbf{b}_n$ is called a *basis* of L: a lattice can be represented by a basis, that is, a row matrix. Two lattice bases are related to one another by some matrix in $GL_n(\mathbb{Z})$. The *dimension* of a lattice L is the dimension n of the linear span of L. The lattice is full-rank if n is the dimension of the space. Let $[\mathbf{v}_1, \dots, \mathbf{v}_k]$ be vectors: we denote by $G(\mathbf{v}_1, \dots, \mathbf{v}_k)$ their *Gram matrix*, that is, the $k \times k$ symmetric positive definite matrix $(\langle \mathbf{v}_i, \mathbf{v}_j \rangle)_{1 \le i,j \le k}$ formed by all the inner products. The *volume* of $[\mathbf{v}_1, \dots, \mathbf{v}_k]$ is $(\det G(\mathbf{v}_1, \dots, \mathbf{v}_k))^{1/2}$, which is zero if the vectors are linearly dependent. The *volume* $\mathrm{vol}(L)$ (or *determinant*) of a lattice L is the volume of any basis of L.

Direct sum. Let L_1 and L_2 be two lattices such that $\mathrm{span}(L_1) \cap \mathrm{span}(L_2) = \{\mathbf{0}\}$. Then the set $L_1 \oplus L_2$ defined as $\{\mathbf{u} + \mathbf{v}, \mathbf{u} \in L_1, \mathbf{v} \in L_2\}$ is a lattice, whose dimension is $\dim L_1 + \dim L_2$. It is the smallest lattice containing L_1 and L_2.

Pure Sublattice. A sublattice U of a lattice L is *pure* if there exists a sublattice V of L such that $L = U \oplus V$. A set $[\mathbf{u}_1, \dots, \mathbf{u}_k]$ of independent lattice vectors of L is *primitive* if and only if $[\mathbf{u}_1, \dots, \mathbf{u}_k]$ can be extended to a basis of L, which is equivalent to $L(\mathbf{u}_1, \dots, \mathbf{u}_k)$ being a pure sublattice of L. For any sublattice U of a lattice L, there exists a pure sublattice S of L such that $\mathrm{span}(S) = \mathrm{span}(U)$, in which case $\mathrm{vol}(U)/\mathrm{vol}(S) = [S : U]$ is an integer.

Successive minima. The *successive minima* of an n-dimensional lattice L are the positive quantities $\lambda_1(L), \dots, \lambda_n(L)$ where $\lambda_r(L)$ is the smallest radius of a zero-centered ball containing r linearly independent vectors of L. The first minimum is the norm of a shortest non-zero vector of L. Note that: $\lambda_1(L) \le \dots \le \lambda_n(L)$.

Hermite's constant. The Hermite invariant of the lattice is defined by $\gamma(L) = \left(\lambda_1(L)/\mathrm{vol}(L)^{\frac{1}{n}}\right)^2$. Hermite's constant γ_n is the maximal value of $\gamma(L)$ over all n-dimensional lattices. Its exact value is known for $1 \le n \le 8$ and $n = 24$, and we have [16]: $\gamma_n \le 1 + \frac{n}{4}$. Asymptotically, the best bounds known are: $\frac{n}{2\pi e} + \frac{\log(\pi n)}{2\pi e} \le \gamma_n \le \frac{1.744n}{2\pi e}(1 + \circ(1))$ (see [8,18]). The lower bound follows from the so-called Minkowski-Hlawka theorem.

Projected Lattice. Given a basis $[\mathbf{b}_1, ..., \mathbf{b}_n]$ of L, let π_i denote the orthogonal projection over $\mathrm{span}(\mathbf{b}_1, ..., \mathbf{b}_{i-1})^\perp$. Then $\pi_i(L)$ is an $(n + 1 - i)$-dimensional lattice. These projections are stable by composition: if $i > j$, then $\pi_i \circ \pi_j = \pi_j \circ \pi_i = \pi_i$. Note that:

$$\pi_i(L) = \pi_i\left(L(\mathbf{b}_i, \ldots, \mathbf{b}_n)\right) = L\left(\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_n)\right)$$

2.2 Lattice reduction

We will consider two quantities to measure the quality of a basis $[\mathbf{b}_1, \ldots, \mathbf{b}_n]$: the first one is the usual *approximation factor* $\|\mathbf{b}_1\| / \lambda_1(L)$, and the second one is $\|\mathbf{b}_1\| / \mathrm{vol}(L)^{1/n}$, which we call the *Hermite factor.* The smaller these quantities, the shorter the first basis vector. Lovász showed in [15] that any algorithm achieving a Hermite factor $\leq q$ can be used to efficiently find a basis with approximation factor $\leq q^2$ using n calls to the algorithm.

Orthogonalization. Given a basis $B = [\mathbf{b}_1, ..., \mathbf{b}_n]$, there exists a unique lower-triangular matrix μ with unit diagonal and an orthogonal family $B^* = [\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*]$ such that $B = \mu B^*$. They can be computed using Gram-Schmidt orthogonalization, and will be denoted the *GSO* of B. Note that $\mathrm{vol}(B) = \prod_{i=1}^n \|\mathbf{b}_i^*\|$, which will often be used. It is well-known [14,17] that:

$$\lambda_1(L(\mathbf{b}_1, ..., \mathbf{b}_n)) \geq \min_{1 \leq i \leq n} \|\mathbf{b}_i^*\| \tag{1}$$

Size-reduction. A basis $[\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is *size-reduced* with factor $\eta \geq 1/2$ if its GSO μ satisfies $|\mu_{i,j}| \leq \eta$ for all $1 \leq j < i$. An individual vector $\mathbf{b}_i$ is size-reduced if $|\mu_{i,j}| \leq \eta$ for all $1 \leq j < i$. Size reduction usually refers to $\eta = 1/2$, and is typically achieved by successively size-reducing individual vectors. Size reduction was introduced by Hermite.

LLL-reduction. A basis $[\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is LLL-reduced [14] with factor (δ, η) for $1/4 < \delta \leq 1$ and $1/2 \leq \eta < \sqrt{\delta}$ if the basis is size-reduced with factor η and if its GSO family satisfies the $(n-1)$ Lovász conditions $(\delta - \mu_{i+1,i}^2) \|\mathbf{b}_i^*\|^2 \leq \left\|\mathbf{b}_{i+1}^*\right\|^2$. LLL-reduction usually refers to the factor $(3/4, 1/2)$ because this was the choice considered in the original LLL paper [14]. But the closer δ and η are respectively to 1 and $1/2$, the more reduced the basis. Reduction with a factor (1,1/2) is closely related to a reduction notion introduced by Hermite [11].

When the reduction factor is close to (1,1/2), Lovász conditions and size-reduction imply the Siegel conditions [6]: $\|\mathbf{b}_i^*\|^2 \lessapprox \frac{4}{3} \left\|\mathbf{b}_{i+1}^*\right\|^2$ for all $1 \leq i \leq n-1$, which limit the drop of the $\|\mathbf{b}_i^*\|$. Here, the $\lessapprox$ symbol means that $\frac{4}{3}$ is actually $\frac{4}{3} + \varepsilon$ for some small $\varepsilon > 0$. In particular, the first vector satisfies $\|\mathbf{b}_1\|^2 \lessapprox \left(\frac{4}{3}\right)^{i-1} \|\mathbf{b}_i^*\|^2$. Hence, the Hermite factor of an LLL-reduced basis is bounded by:

$$\|\mathbf{b}_1\| / \mathrm{vol}(L)^{1/n} \lessapprox \left(\frac{4}{3}\right)^{(n-1)/4} = (\sqrt{\gamma_2})^{n-1}$$

and (1) implies that the approximation factor is bounded by:

$$\|\mathbf{b}_1\| / \lambda_1(L) \lessapprox \left(\frac{4}{3}\right)^{(n-1)/2} = (\gamma_2)^{n-1}$$

The LLL algorithm is an iterative algorithm. At the start of each loop iteration, the first i vectors are already LLL-reduced, then the $(i+1)$-th vector is size-reduced; if it does not satisfy Lovász condition, the consecutive vectors $\mathbf{b}_{i+1}$ and $\mathbf{b}_i$ are swapped and the counter i is decremented, otherwise i is incremented. The loop goes on until i eventually reaches the value n. If L is a full-rank integer lattice of dimension n and B is an upper bound on the $\|\mathbf{b}_i\|$'s, then the complexity of the LLL algorithm described (using integral Gram-Schmidt) without fast integer arithmetic is $O(n^6 \log^3 B)$. The main reason is that the integer $\prod_{k=1}^{n} \|\mathbf{b}_k^*\|^{2(n-k)}$ decreases by at least the geometric factor δ at every swap: thus, the number of swaps is $O(n^2 \log B)$. The recent L^2 algorithm [20] by Nguyen and Stehlé achieves a factor of (δ, ν) arbitrarily close to (1,1/2) in faster polynomial time: the complexity is $O(n^5(n + \log B)\log B)$ which is essentially $O(n^5 \log^2 B)$ for large entries. This is the fastest LLL-type reduction algorithm known for large entries.

HKZ REDUCTION. A basis $[\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice L is Hermite-Korkine-Zolotarev (HKZ) reduced if it is size-reduced and if $\mathbf{b}_i^*$ is a shortest vector of the projected lattice $\pi_i(L)$ for all $1 \leq i \leq n$. In particular, the first basis vector is a shortest vector of the lattice. Schnorr introduced in [24] a constant to globally measure the drop of the $\|\mathbf{b}_i^*\|$ of $2k$-dimensional HKZ bases:

$$\beta_k = \max_{\substack{L\ 2k\text{-dim. lattice} \\ H \text{ HKZ-basis of L}}} \left(\frac{\|\mathbf{h}_1^*\| \times \cdots \times \|\mathbf{h}_k^*\|}{\|\mathbf{h}_{k+1}^*\| \times \cdots \times \|\mathbf{h}_{2k}^*\|} \right)^{\frac{2}{k}}$$

which we rewrite more geometrically as,

$$\beta_k = \max_{\substack{L\ 2k\text{-dim. lattice} \\ H \text{ HKZ-basis of L}}} \left(\frac{\mathrm{vol}(\mathbf{h}_1, \ldots, \mathbf{h}_k)}{\mathrm{vol}(\pi_{k+1}(\mathbf{h}_{k+1}), \ldots, \pi_{k+1}(\mathbf{h}_{2k}))} \right)^{\frac{2}{k}}$$

Schnorr proved that $\beta_k \leq 4k^2$, and Ajtai recently proved in [1] that there exists $\varepsilon > 0$ such that $\beta_k \geq k^{\varepsilon}$, but this is an existential lower bound: no explicit value of ε is known. The value of β_k is very important to bound the output quality of Schnorr's algorithm. One can achieve an n-dimensional HKZ reduction in essentially the same time as finding the shortest vector of an n-dimensional lattice: the deterministic algorithm [24] needs $n^{O(n)}$ polynomial operations, and the probabilistic algorithm [2] needs $2^{O(n)}$ polynomial operations.

3 Revisiting Schnorr's Algorithm

In this section, we give an intuitive description of Schnorr's semi block-$2k$-reduction algorithm and show that it is very similar to LLL. The analogy between LLL and Schnorr's algorithm is summarized in Tables 2 and 1. We explain the relationship between the constant β_k and the quality of Schnorr reduced basis, and we give the main ideas for its complexity analysis. Here, we assume that the lattice dimension n is a multiple of k.

3.1 From LLL to Schnorr

In the LLL algorithm, vectors are considered two by two. At each loop iteration, the 2-dimensional lattice $L_i = [\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1})]$, is partially reduced (through a swap) in order to decrease $\|\mathbf{b}_i^*\|$ by at least some geometric factor. When all L_i are almost reduced, every ratio $\|\mathbf{b}_i^*\| / \left\|\mathbf{b}_{i+1}^*\right\|$ is roughly less than $\gamma_2 = \sqrt{\frac{4}{3}}$, which is Siegel's condition [6].

Schnorr's semi block-$2k$-reduction is a polynomial-time block generalization of the LLL algorithm, where the vectors $\mathbf{b}_i^*$ are "replaced" by k-dimensional blocks $S_i = [\pi_{ik-k+1}(\mathbf{b}_{ik-k+1}), \ldots, \pi_{ik-k+1}(\mathbf{b}_{ik})]$ where $1 \leq i \leq \frac{n}{k}$. The analogue of the 2-dimensional L_i in LLL are the $2k$-dimensional large blocks $L_i = [\pi_{ik-k+1}(\mathbf{b}_{ik-k+1}), \ldots, \pi_{ik-k+1}(\mathbf{b}_{ik+k})]$ where $1 \leq i \leq \frac{n}{k} - 1$. The link between the small blocks $S_1, \ldots, S_{n/k}$ and the large blocks $L_1, \ldots, L_{n/k-1}$ is that S_i consists of the first k vectors of L_i, while S_{i+1} is the projection of the last k vectors of L_i over $\mathrm{span}(S_i)^{\perp}$. As a result, $\mathrm{vol}(L_i) = \mathrm{vol}(S_i) \times \mathrm{vol}(S_{i+1})$.

Table 1. Analogy between LLL and Schnorr's algorithm

LLL	**Schnorr's semi block-$2k$ reduction**
1: **while** $i \leq n$ **do**	1: **while** $i \leq n/k$ **do**
2: Size-reduce $\mathbf{b}_i$	2a: HKZ-reduce S_i, do the transformations on the basis vectors, not just on the projections 2b: Size-reduce $\mathbf{b}_{ik-k+1}, \ldots, \mathbf{b}_{ik}$.
3: $B' \leftarrow$copy of B 4: Try to decrease $\|\mathbf{b}_i^*\|$ in B': 4a: • by swap of $(\mathbf{b}_i, \mathbf{b}_{i+1})$	3: $B' \leftarrow$copy of B 4: Try to decrease $\mathrm{vol}(S_i)$ in B': 4a: • by swap of $(\mathbf{b}_{ik}, \mathbf{b}_{ik+1})$ 4b: • by HKZ reducing L_i
5: **if** $\|\mathbf{b}_i^*\|$ can lose a factor δ **then** 6: • perform the changes on B 7: • $i \leftarrow \max(i-1, 1)$ 8: **else** $i \leftarrow i+1$	5: **if** $\mathrm{vol}(S_i)$ can lose a factor $\frac{1}{(1+\varepsilon)}$ **then** 6: • perform the changes on B 7: • $i \leftarrow \max(i-1, 1)$ 8: **else** $i \leftarrow i+1$
9: **endwhile**	9: **endwhile**

Formally, a basis is semi-block-$2k$-reduced if the following three conditions hold for some small $\varepsilon > 0$:

$$B \text{ is LLL-reduced} \tag{2}$$

$$\text{For all } 1 \le i \le \frac{n}{k},\quad S_i \text{ is HKZ-reduced} \tag{3}$$

$$\text{For all } 1 \le i < \frac{n}{k},\quad \left(\frac{\mathrm{vol}(S_i)}{\mathrm{vol}(S_{i+1})}\right)^2 \le (1+\varepsilon)\beta_k^k \tag{4}$$

Like in LLL, the large block L_i is reduced at each loop iteration in order to decrease $\mathrm{vol}(S_i)$ by a geometric factor $1/(1+\varepsilon)$. Note that $\mathrm{vol}(S_i)/\mathrm{vol}(S_{i+1})$ decreases by $1/(1+\varepsilon)^2$. By definition of β_k, this ratio can be made smaller than $\beta_k^{k/2}$ if L_i is HKZ-reduced. For this reason, condition (4) is a block generalization of Siegel's condition which can be fulfilled by an HKZ-reduction of L_i.

Table 2. Comparison between LLL and Schnorr's semi block-$2k$ reduction.

Algorithm	LLL	Schnorr's Semi-$2k$ reduction
Upper bound on $\lVert\mathbf{b}_1\rVert/\mathrm{vol}(L)^{\frac{1}{n}}$	$\approx \left(\frac{4}{3}\right)^{\frac{n}{4}}$	$\approx \beta_k^{\frac{n}{4k}}$
Upper bound on $\lVert\mathbf{b}_1\rVert/\lambda_1(L)$	$\approx \left(\frac{4}{3}\right)^{\frac{n}{2}}$	$\approx \beta_k^{\frac{n}{2k}}$
time	Poly(size of basis)	Poly(size of basis)*HKZ($2k$)
small block S_i	$\pi_i(\mathbf{b}_i) = \mathbf{b}_i^*$	$[\pi_{ik-k+1}(\mathbf{b}_{ik-k+1}), \ldots, \pi_{ik-k+1}(\mathbf{b}_{ik})]$
large block L_i	$[\pi_{i-1}(\mathbf{b}_{i-1}), \pi_{i-1}(\mathbf{b}_i)]$	$[\pi_{ik-2k+1}(\mathbf{b}_{ik-2k+1}), \ldots, \pi_{ik-2k+1}(\mathbf{b}_{ik})]$
size of small block	1	k
size of large block	2	$2k$
Quantity to upper bound	$\lVert\mathbf{b}_i^*\rVert / \lVert\mathbf{b}_{i+1}^*\rVert$	$\mathrm{vol}(S_i)/\mathrm{vol}(S_{i+1})$
Method	Reduce L_i by size-reduction and swap	HKZ reduce L_i
Potential	$\prod_{i=1}^n \lVert\mathbf{b}_i^*\rVert^{2(n-i)}$	$\prod_{i=1}^{\frac{n}{k}} \mathrm{vol}(S_i)^{2(\frac{n}{k}-i)}$

3.2 Complexity analysis

Each time a large block L_i is reduced, $\mathrm{vol}(S_i)$ decreases by a geometric factor $1/(1+\varepsilon)$ and since $\mathrm{vol}(L_i) = \mathrm{vol}(S_i) \times \mathrm{vol}(S_{i+1})$ remains constant, $\mathrm{vol}(S_{i+1})$ increases by the same factor. So the integer quantity $\prod_{i=1}^{n/k} \mathrm{vol}(S_i)^{2\left(\frac{n}{k}-i\right)}$ decreases by $1/(1+\varepsilon)^2$. This can occur at most a polynomial number of times: hence the complexity of the reduction is Poly(size of basis)*HKZ($2k$) where HKZ($2k$) is the complexity of a $2k$-dimensional HKZ reduction as seen in Section 2.2. In order to ensure a polynomial complexity, it is necessary to keep $k \le \log n/\log\log n$ or $k \le \log n$ if we use the probabilistic AKS algorithm.

3.3 The Hermite factor of Schnorr's reduction

The Hermite factor of a semi block-$2k$-reduced basis depends mostly on Condition (4), which implies that $\mathrm{vol}(S_1) \lessapprox \beta_k^{\frac{n}{4}} \mathrm{vol}(L)^{k/n}$ because $\mathrm{vol}(L) = \prod_{i=1}^{n/k} \mathrm{vol}(S_i)$. If the first vector $\mathbf{b}_1$ is the shortest vector of S_1 (which is implied by (3)), then $\|\mathbf{b}_1\| \leq \sqrt{\gamma_k}\mathrm{vol}(S_1)^{\frac{1}{k}}$ by definition of Hermite's constant, and therefore:

$$\|\mathbf{b}_1\| / \mathrm{vol}(L)^{1/n} \lessapprox \sqrt{\gamma_k}\beta_k^{\frac{n}{4k}}$$

3.4 The approximation factor of Schnorr's reduction

If only condition (4) holds, even if $\mathbf{b}_1$ is the shortest vector of S_1, its norm can be arbitrarily far from the first minimum of L. Indeed, consider for instance the 6-dimensional lattice generated by $\mathrm{Diag}(1, 1, 1, 1, \varepsilon, \frac{1}{\varepsilon})$, and a blocksize $k = 3$. Then the first block S_1 is the identity and is therefore HKZ-reduced. The volume of the two blocks S_1 and S_2 is 1, thus condition (4) holds. But the norm of the first vector ($\|\mathbf{b}_1\| = 1$) is arbitrarily far from the shortest vector $\|\mathbf{b}_5\| = \varepsilon$.

Compared to Hermite's factor, we require additionally that every block S_k is reduced (which follows from condition (2)) to bound the approximation factor.

Using (1), there exists an index p such that $\|\mathbf{b}_p^*\| \leq \lambda_1(L)$. Let $a = \lfloor (p-1)/k \rfloor$, (so that the position p is inside the block S_{a+1}). Since B is LLL-reduced, $\mathrm{vol}(S_a) \lessapprox \frac{4}{3}^{\frac{(3k-1)k}{4}} \lambda_1(L)^k$, so the approximation factor is bounded by:

$$\|\mathbf{b}_1\| / \lambda_1(L) \lessapprox \sqrt{\gamma_k}\frac{4}{3}^{\frac{(3k-1)}{4}} \beta_k^{\frac{n/k-2}{2}}$$

Note however that Schnorr proved in [24] that Condition (3) allows to decrease the term $(4/3)^{(3k-1)/4}$ to $O\left(k^{2+\ln k}\right)$.

4 Rankin's constant and Schnorr's algorithm

4.1 Rankin's constant

If L is a n-dimensional lattice and $1 \leq m \leq n$, the Rankin invariant $\gamma_{n,m}(L)$ is defined as (cf. [23]):

$$\gamma_{n,m}(L) = \min_{\substack{\mathbf{x}_1, \ldots, \mathbf{x}_m \in L \\ \mathrm{vol}(\mathbf{x}_1, \ldots, \mathbf{x}_m) \neq 0}} \left(\frac{\mathrm{vol}(\mathbf{x}_1, \ldots, \mathbf{x}_m)}{\mathrm{vol}(L)^{m/n}}\right)^2$$

which can be rewritten as:

$$\gamma_{n,m}(L) = \min_{\substack{S \text{ sublattice of } L \\ \dim S = m}} \left(\frac{\mathrm{vol}(S)}{\mathrm{vol}(L)^{m/n}}\right)^2$$

Rankin's constant is the maximum $\gamma_{n,m} = \max \gamma_{n,m}(L)$ over all n-dimensional lattices. Clearly, $\gamma_{n,n}(L) = 1$ and $\gamma_{n,1}(L) = \gamma_n(L)$, so $\gamma_{n,n} = 1$ and $\gamma_{n,1} = \gamma_n$. Rankin's constants satisfy the following three relations, which are proved in [16,23]:

$$\forall n \in \mathbb{N},\ \gamma_{n,1} = \gamma_n \tag{5}$$

$$\forall n, m \text{ with } m < n\ \gamma_{n,m} = \gamma_{n,n-m} \tag{6}$$

$$\forall r \in [m+1, n-1],\ \gamma_{n,m} \leq \gamma_{r,m}(\gamma_{n,r})^{m/r} \tag{7}$$

The only known values of Rankin's constants are $\gamma_{4,2} = \frac{3}{2}$, which is reached for the $\mathbb{D}_4$ lattice, and those corresponding to the nine Hermite constants known. In the definition of $\gamma_{n,m}(L)$, the minimum is taken over sets of m linearly independent vectors of L, but we may restrict the definition to primitive sets of L or pure sublattices of L, since for any sublattice S of L, there exists a pure sublattice S_1 of L with $\mathrm{span}(S) = \mathrm{span}(S_1)$ and $\mathrm{vol}(S)/\mathrm{vol}(S_1) = [S : S_1]$. If $\mathrm{vol}(S)$ is minimal, then $[S : S_1] = 1$ so $S = S_1$ is pure.

4.2 Relation between Rankin's constant and Schnorr's constant

Theorem 1. *For all $k \geq 1$, $(\gamma_{2k,k})^{2/k} \leq \beta_k$.*

Proof. Let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_{2k}]$ be a HKZ-reduced basis of a lattice L, and let $h(B) = \left(\|\mathbf{b}_1^*\|^2 \times \cdots \times \|\mathbf{b}_k^*\|^2\right) / \left(\left\|\mathbf{b}_{k+1}^*\right\|^2 \times \cdots \times \|\mathbf{b}_{2k}^*\|^2\right)$. By definition of β_k, $h(B) \leq \beta_k^k$. On the other hand, $h(B)$ is also equal to $\left(\mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_k)/\mathrm{vol}(L)^{1/2}\right)^4$ and therefore: $\gamma_{2k,k}^2(L) \leq h(B)$. Thus $(\gamma_{2k,k}(L))^{2/k} \leq \beta_k$, which completes the proof. □

4.3 Improving the upper bound on Schnorr's Constant

The key result of this section is:

Theorem 2. *For all $k \geq 2$, Schnorr's constant β_k satisfies: $\beta_k \leq \left(1 + \frac{k}{2}\right)^{2\ln 2 + \frac{1}{k}}$. Asymptotically it satisfies $\beta_k \leq \frac{1}{10} k^{2\ln 2}$.*

Without any change to Schnorr's algorithm, we deduce a much better quality for the output basis than with the former bound $\beta_k \leq 4k^2$, because both the exponent $2\ln 2 \approx 1.386$ is much lower than 2, and the coefficient $1/2^{2\ln 2}$ is about 10 times lower than 4. The bounds on the approximation factor and Hermite's factor of Schnorr's algorithm can be raised to the power $\ln 2 \approx 0.69$. The proof uses an easy bound mentioned by Schnorr in [24]:

$$\beta_k \leq \prod_{j=0}^{k-1} \gamma_{k+j+1}^{2/(k+j)} \tag{8}$$

Numerically, it can be verified that the product (8) is $\leq k^{1.1}$ for all $k \leq 100$ (see Figure 2). The bound $\gamma_j \leq 1 + \frac{j}{4}$ combined with an upper bound of the total exponents prove Theorem 2 for all k (see Section A in the appendix).

Surprisingly, we do not know a better upper bound on $(\gamma_{2k,k})^{2/k}$ than that of Theorem 2. The inequality (7) leads exactly to the same bound for Rankin's constant.

4.4 A lower bound on Rankin's constant

In [1], Ajtai showed that $\beta_k \geq k^{\varepsilon}$ for small size of blocks and for some $\varepsilon > 0$, and presented worst cases for Schnorr's algorithm, which implies that the reduction power of semi block $2k$-reduction is limited. The following result proves an explicit lower bound on Rankin's constant, which suggests (but does not prove) that the approximation factor of any block-reduction algorithm (including Schnorr's semi block $2k$-reduction) based on the LLL strategy is limited.

Theorem 3. *Rankin's constant satisfies* $(\gamma_{2k,k})^{\frac{2}{k}} \geq \frac{k}{12}$ *for all* $k \geq 1$.

This lower bound also applies to Schnorr's constant β_k because of Theorem 1. Theorem 3 is mainly based on the following lower bound for Rankin's constant proved in [28,3] as a generalization of Minkowski-Hlawka's theorem:

$$\gamma_{n,m} \geq \left(n \frac{\prod_{j=n-m+1}^{n} Z(j)}{\prod_{j=2}^{m} Z(j)} \right)^{\frac{2}{n}}$$

where $Z(j) = \zeta(j)\Gamma(\frac{j}{2})/\pi^{\frac{j}{2}}$, $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t} \cdot dt$ and ζ is Riemann's zeta function: $\zeta(j) = \sum_{p=1}^{\infty} p^{-j}$. As an application, for $k < 100$, it can be verified numerically that $(\gamma_{2k,k})^{\frac{2}{k}} \geq \frac{k}{9}$. More generally, we first bound $\ln Z(j)$, and we compare it to an integral to get the expected lower bound. The full proof of Theorem 3 is given in Section B of the Appendix.

5 Improving Schnorr's algorithm

The main subroutine in Schnorr's algorithm tries to solve the following problem: given a $2k$-dimensional lattice L, find a basis $[\mathbf{b}_1, \ldots, \mathbf{b}_{2k}]$ of L such that the two k-dimensional blocks $S_1 = L(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ and $S_2 = \pi_{k+1}(L)$ minimize $\mathrm{vol}(S_1)$, because $\mathrm{vol}(S_1)/\mathrm{vol}(S_2) = \mathrm{vol}(S_1)^2/\mathrm{vol}(L)$ where $\mathrm{vol}(L)$ does not change. In Schnorr's algorithm, the quality of the output basis (which was expressed as a function of β_k in Sections 3.3 and 3.4) essentially depends on the upper bound that can be achieved on the ratio $\mathrm{vol}(S_1)/\mathrm{vol}(S_2)$.

5.1 The smallest volume problem

Rankin's constant and Schnorr's algorithm suggest the *smallest volume problem*: given a n-dimensional lattice L and an integer m such that $1 \leq m \leq n$, find an m-dimensional sublattice S of L such that $\mathrm{vol}(S)$ is minimal, that is, $\mathrm{vol}(S)/\mathrm{vol}(L)^{m/n} = \sqrt{\gamma_{n,m}(L)}$.

If $m = 1$, the problem is simply the shortest vector problem (SVP). If $m = n-1$, the problem is equivalent to the shortest vector problem in the dual lattice. When $(n, m) = (2k, k)$, we call this problem the *half-volume problem.* For any $m \leq n$, the minimality of the volume implies that any solution to this problem is a pure sublattice of L, so one way to solve this problem is to find a basis $[\mathbf{b}_1, \ldots, \mathbf{b}_n]$ such that $\mathrm{vol}\,(L(\mathbf{b}_1, \ldots, \mathbf{b}_m))$ is minimal.

We say that a basis of a n-dimensional lattice L is m*-Rankin reduced* if its first m vectors solve the smallest volume problem. Note that this is not exactly a basis reduction problem, as any notion of reduction of the basis of S is irrelevant. The only thing that matters is to minimize the volume of S. If we apply the LLL algorithm on a Rankin-reduced basis, the volume of the first m vectors can never increase: this means that LLL swaps never involve the pair $(m, m+1)$, and therefore the output basis is both LLL-reduced and Rankin-reduced. We thus have proved the following lemma:

Lemma 1. *Let L be a n-dimensional sublattice and $1 \leq m \leq n$. There exists an LLL-reduced basis of L which is m-Rankin-reduced.*

Since the number of LLL reduced bases can be bounded independently of the lattice (see [6] because LLL-reduction implies Siegel reduction), the smallest volume problem can be solved by a gigantic exhaustive search (which is constant in fixed dimension though).

5.2 Block-Rankin reduction

A basis is $2k$*-Block-Rankin reduced* with factor $\delta \in [\frac{1}{2}; 1[$ if it is LLL-reduced with factor $(\frac{1}{2}, \delta)$ and all the blocks S_i and L_i defined as in Section 3 satisfy: $\mathrm{vol}(S_i)/\mathrm{vol}(S_{i+1}) \leq \frac{1}{\delta}\gamma_{2k,k}(L_i)$. Compared to Schnorr's semi block-$2k$ reduction, this reduction notion enables to replace β_k in the bounds of the approximation factor and Hermite's factor by $\gamma_{2k,k}^{2/k}$.

Assume that an algorithm to k-Rankin-reduce a $2k$-dimensional basis is available. Then it is easy to see that Algorithm 1, inspired from LLL and Schnorr's semi block-$2k$ reduction, achieves block-Rankin reduction using a polynomial number of calls to the Rankin subroutine.

5.3 The 4-dimensional case

Here, we study Rankin-reduction for $(n, m) = (4, 2)$. We first notice that HKZ reduction does not necessarily solve the half-volume problem. Consider indeed the following HKZ-reduced row basis:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1+\varepsilon & 0 \\ 0 & 0 & \frac{1+\varepsilon}{2} & \frac{\sqrt{3}}{2}(1+\varepsilon) \end{bmatrix}$$

Algorithm 1 2k-block-Rankin reduction

Input: A basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice and $\delta \in [\frac{1}{2}; 1[$
Output: A semi block $2k$-reduced basis.
$i \leftarrow 1$;
while $i \leq n/k$ **do**
  LLL-reduce S_i with factor δ, do the transformations on the basis vectors, not just on their projections
  return B **if** $i = n/k$.
  $B_{\text{tmp}} \leftarrow B$; k-Rankin reduce L_i in Btmp
  if $\text{vol}(S_i)$ in $B_{\text{tmp}} \leq \delta \text{vol}(S_i)$ in B **then**
    $B \leftarrow B_{\text{tmp}}$; $i \leftarrow i - 1$
  else
    $i \leftarrow i + 1$
  end if
end while

The volume ratio $\frac{\|\mathbf{b}_1^*\|\|\mathbf{b}_2^*\|}{\|\mathbf{b}_3^*\|\|\mathbf{b}_4^*\|}$ is equal to $\sqrt{\frac{4}{3}}$. If we swap the two 2-blocks, the new basis is no longer HKZ-reduced, but the ratio decreases to almost $\sqrt{\frac{3}{4}}$. This example can easily be generalized to any even dimension, which gives an infinite family of HKZ bases which do not reach the minimal half-volume.

However, the following lemma shows that Algorithm 2 can efficiently solve the half-volume problem in dimension 4, given as input an HKZ basis:

Lemma 2. *Let $(\mathbf{b}_1, ..., \mathbf{b}_4)$ be an HKZ-reduced basis of a lattice L. To simplify notations, let λ_1 and λ_2 denote respectively $\lambda_1(L)$ and $\lambda_2(L)$. For all $\mathbf{c}_1$ and $\mathbf{c}_2$ in L such that $\text{vol}(\mathbf{c}_1, \mathbf{c}_2) \leq \text{vol}(\mathbf{b}_1, \mathbf{b}_2)$ and $(\mathbf{c}_1, \mathbf{c}_2)$ is reduced: $\|\mathbf{c}_1\| \leq \|\mathbf{c}_2\|$ and $\|\mathbf{c}_1^*\|^2 \leq \frac{4}{3}\|\mathbf{c}_2^*\|^2$.*

1. *Then $\mathbf{c}_1$ satisfies: $\lambda_1^2 \leq \|\mathbf{c}_1\|^2 \leq \frac{4}{3}\lambda_1^2$.*
2. *If $\lambda_2 > \sqrt{\frac{4}{3}}\lambda_1$, then $\text{vol}(\mathbf{c}_1, \mathbf{c}_2) = \text{vol}(\mathbf{b}_1, \mathbf{b}_2)$ given by HKZ reduction.*
3. *Otherwise $\mathbf{c}_2$ satisfies$\|\mathbf{c}_2\|^2 \leq (\frac{4}{3}\lambda_1)^2$.*

Proof. Because $\mathbf{c}_1$ belongs to L, $\|\mathbf{c}_1\| \geq \lambda_1$. Since $(\mathbf{b}_1, ..., \mathbf{b}_4)$ is an HKZ basis, the first vector is a shortest vector: $\|\mathbf{b}_1\| = \lambda_1$ and the second vector satisfies $\|\mathbf{b}_2^*\| \leq \lambda_2$, so $\text{vol}(\mathbf{b}_1, \mathbf{b}_2) \leq \lambda_1\lambda_2$. We also know that $\text{vol}(\mathbf{c}_1, \mathbf{c}_2) = \|\mathbf{c}_1\| \cdot \|\mathbf{c}_2\| \cdot \sin(\mathbf{c}_1, \mathbf{c}_2) \geq \frac{\sqrt{3}}{2}\|\mathbf{c}_1\| \cdot \|\mathbf{c}_2\|$ because $[\mathbf{c}_1, \mathbf{c}_2]$ is reduced. Since we have chosen $\|\mathbf{c}_1\| \leq \|\mathbf{c}_2\|$, then $\|\mathbf{c}_2\| \geq \lambda_2$. Thus $\lambda_1\lambda_2 \geq \frac{\sqrt{3}}{2}\|\mathbf{c}_1\| \cdot \lambda_2$, and $\|\mathbf{c}_1\|^2 \leq \frac{4}{3}\lambda_1^2$. If furthermore $\lambda_2 > \sqrt{\frac{4}{3}}\lambda_1$, then necessarily $\mathbf{c_1} = \pm\mathbf{b_1}$, then the HKZ reduction implies the minimality of $\text{vol}(\mathbf{b_1}, \mathbf{b_2})$. If $\lambda_2 \leq \sqrt{\frac{4}{3}}\lambda_1$, then $\text{vol}(\mathbf{c_1}, \mathbf{c}_2)^2 = \|\mathbf{c}_1\|^2 \cdot \|\mathbf{c}_2^*\|^2 \leq (\lambda_1\lambda_2)^2$, so $\|\mathbf{c}_2^*\|^2 \leq \lambda_2^2 \leq \frac{4}{3}\lambda_1^2$. And we also have $\|\mathbf{c}_2^*\|^2 \geq \frac{3}{4}\|\mathbf{c}_1^*\|^2$. □

Because the input basis is HKZ-reduced, it is easy to see that the number of vectors $\mathbf{c}_1$ enumerated in Algorithm 2 is bounded by a constant. It follows that

Algorithm 2 4-dimensional Rankin-reduction

Input: An HKZ reduced basis $[\mathbf{b}_1, \ldots, \mathbf{b}_4]$ of a 4-dim lattice
Output: A Rankin-reduced basis $[\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4]$ minimizing $\mathrm{vol}(\mathbf{c}_1, \mathbf{c}_2)$

if $\|\mathbf{b}_2^*\| > \sqrt{\frac{4}{3}}\|\mathbf{b}_1\|$ **then**
  return $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4)$
end if
$(\mathbf{u}, \mathbf{v}) \leftarrow (\mathbf{b}_1, \mathbf{b}_2)$
for each lattice vector $\mathbf{c}_1$ shorter than $\sqrt{\frac{4}{3}}\|\mathbf{b}_1\|$ **do**
  find the shortest vector $\mathbf{c}_2$ in the lattice projected over $\mathbf{c}_1^{\perp}$ (We can limit the enumeration to $\|\mathbf{c}_2\|$ lower than $\frac{\mathrm{vol}(\mathbf{b}_1, \mathbf{b}_2)}{\|\mathbf{c}_1\|}$).
  if $\mathrm{vol}(\mathbf{c}_1, \mathbf{c}_2) < \mathrm{vol}(\mathbf{u}, \mathbf{v})$ **then** $(\mathbf{u}, \mathbf{v}) \leftarrow (\mathbf{c}_1, \mathbf{c}_2)$
end for
compute $\mathbf{c}_3$ and $\mathbf{c}_4$ a reduced basis of the lattice projected over $(\mathbf{u}, \mathbf{v})^{\perp}$
return $(\mathbf{u}, \mathbf{v}, \mathbf{c}_3, \mathbf{c}_4)$

the cost of Algorithm 2 is at most a constant times more expensive than a HKZ reduction of a 4-dimensional lattice.

If we plug Algorithm 2 into Algorithm 1, we obtain a polynomial-time reduction algorithm whose provable quality is a bit better than Schnorr's semi block-4 reduction: namely, the constant $\beta_2 \leq \gamma_4^{2/3}\gamma_3 = 2^{2/3} \approx 1.587$ in the approximation factor and the Hermite factor is replaced by the potentially smaller constant $\gamma_{4,2} = 3/2$. On the other hand, both algorithms only apply exhaustive search in dimension 4.

5.4 Higher blocksizes

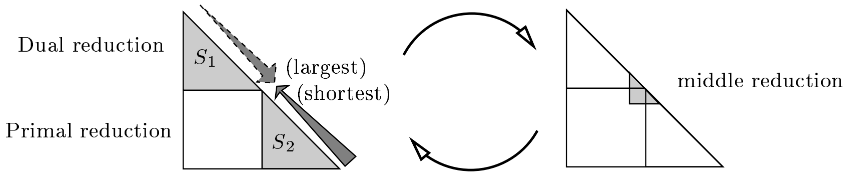
The 4-dimensional case suggests two potential improvements over Schnorr's semi block $2k$-algorithm:

- If the half-volume problem can be solved in roughly the same time (or less) than a full $2k$-HKZ reduction, then Algorithm 1 would give potentially better approximation factors at the same cost.
- If one can approximate the half-volume problem in much less time than a full $2k$-HKZ reduction, we may still obtain good approximation factors in much less time than semi block $2k$-reduction, by plugging the approximation algorithm in place of Rankin reduction in Algorithm 1.

However, we do not know how to solve the half-volume problem exactly in dimension higher than 4, without using a gigantic exhaustive search. Perhaps a good approximation can be found in reasonable blocksize, by sampling short (but not necessarily shortest) lattice vectors, and testing random combinations of such vectors.

We now present an approximation algorithm for the half volume problem, which we call *transference reduction*. Transference reduction achieves a volume

Fig. 1. Transference reduction



ratio lower than $\frac{1}{95}k^2$ in a $2k$-dimensional lattice by making only $O(k)$ calls to a k-dimensional exhaustive search, which is thus cheaper than a full $2k$-HKZ reduction. Note that $2k$-HKZ reduction achieves a smaller volume ratio using a $2k$-dimensional exhaustive search. Let $(\mathbf{b}_1, \ldots, \mathbf{b}_{2k})$ be a basis of a $2k$-dimensional lattice. The idea of the algorithm is to perform exhaustive searches in the two halves of the basis in order to find a pair of vectors which can be highly reduced. The reduction of this pair of vectors happens in the middle of the basis so that the first half-volume decreases.

As in the previous sections, we call $S_1 = L(\mathbf{b}_1, \ldots \mathbf{b}_k)$ and $S_2 = L(\pi_{k+1}(\mathbf{b}_{k+1}), \ldots, \pi_{k+1}(\mathbf{b}_{2k}))$. Using an exhaustive search, a shortest vector of S_2 is brought on the $k+1$-th position in order to make $\left\|\mathbf{b}_{k+1}^*\right\|^2 \leq \gamma_k \mathrm{vol}(S_2)^{2/k}$. The algorithm used to perform this exhaustive search in dimension k in a projected lattice is classical. Then a second exhaustive search brings a vector of S_1 maximizing $\|\mathbf{b}_k^*\|$ on the k-th position.

Lemma 3. *Finding a basis $(\mathbf{b}_1, \ldots, \mathbf{b}_k)$ of a k-dimensional lattice S maximizing $\|\mathbf{b}_k^*\|$ reduces to finding a shortest vector of the dual lattice $S^\times$.*

Proof. The vector $\mathbf{u} = \mathbf{b}_k^* / \|\mathbf{b}_k^*\|^2$ is the last vector of the dual basis. Indeed, $\langle \mathbf{u}, \mathbf{b}_i \rangle = 0$ for $i = 1..k-1$ and $\langle \mathbf{u}, \mathbf{b}_k \rangle = 1$. If $\|\mathbf{b}_k^*\|$ is maximal, then $\mathbf{u}$ is minimal. So a simple reduction is to find a shortest vector $\mathbf{u}_k$ of the dual $S^\times$, extend it into a basis $U = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$ of $S^\times$ and return the dual $U^{-t} = (\mathbf{b}_1, \ldots, \mathbf{b}_k)$. $\square$

After maximizing $\|\mathbf{b}_k^*\|$, Hermite's inequality in the reversed dual of S_1 implies that $1/\|\mathbf{b}_k^*\|^2 \leq \gamma_k / \mathrm{vol}(S_1)^{2/k}$. At this point, the ratio $(\mathrm{vol}(S_1)/\mathrm{vol}(S_2))^{2/k}$ is lower than $\gamma_k^2 \|\mathbf{b}_k^*\|^2 / \|\mathbf{b}_{k+1}\|^2$. If the middle-vectors pair $(\pi_k(\mathbf{b}_k), \pi_k(\mathbf{b}_{k+1}))$ does not satisfy Lovász condition, then it is fully reduced and the algorithm starts over from the beginning. The only step which can change the ratio $\mathrm{vol}(S_1)/\mathrm{vol}(S_2)$ is the middle-vectors reduction, in which case it drops by a geometric factor. Hence the number of swaps in the middle is at most linear in the dimension and the size of the basis. In the end, the ratio $(\mathrm{vol}(S_1)/\mathrm{vol}(S_2))^{2/k}$ is lower than $\frac{4}{3}\gamma_k^2$ (or $\frac{1}{12}k^2$).

The constant $4/3$ in this ratio can be reduced to almost 1 by adding further reduction conditions. Let $\hat{S}_1 = L(\mathbf{b}_1, \ldots, \mathbf{b}_{k+1})$ and $\hat{S}_2 = L(\pi_k(\mathbf{b}_k), \ldots, \pi_k(\mathbf{b}_{2k}))$ the widened blocks of S_1, S_2 and $\delta : \frac{1}{2} \leq \delta < 1$ a relaxing parameter. After minimizing $\left\|\mathbf{b}_{k+1}^*\right\|$ in S_2 and maximizing $\|\mathbf{b}_k^*\|$ in S_1 the following steps are performed: Using the third exhaustive search, a shortest vector of $\hat{S}_2$ is found. Only if the squared size of this shortest vector is smaller than $\delta \left\|\mathbf{b}_k^*\right\|^2$ this vector is brought on the k-th position and the algorithm starts over with minimizing $\left\|\mathbf{b}_{k+1}^*\right\|$ in S_2 and maximizing $\left\|\mathbf{b}_k^*\right\|$ in S_1. Otherwise the fourth exhaustive search in the dual of $\hat{S}_1$ checks if $\left\|\mathbf{b}_{k+1}^*\right\|^2$ approximates the maximized solution by the factor δ. If this condition does not hold, $\mathbf{b}_{k+1}$ is replaced by the maximized solution and the algorithm starts over from the beginning. Each of these two reduction steps decrease $\mathrm{vol}(S_1)^2$ by the factor δ, therefore the number of steps is still bounded by $O(k)$. In case both conditions hold the algorithm stops. For these new conditions we can again apply Hermite's inequality resulting in $\delta \cdot \left\|\mathbf{b}_k^*\right\|^2 \leq \gamma_{k+1} \mathrm{vol}(\hat{S}_2)^{2/(k+1)}$ and $\delta \cdot 1/\left\|\mathbf{b}_{k+1}^*\right\|^2 \leq \gamma_{k+1}/\mathrm{vol}(\hat{S}_1)^{2/(k+1)}$. It follows:

$$\left(\mathrm{vol}(\hat{S}_1)/\mathrm{vol}(\hat{S}_2)\right)^{2/(k+1)} \leq \gamma_{k+1}^2/\delta^2 \cdot \left\|\mathbf{b}_{k+1}^*\right\|^2 / \left\|\mathbf{b}_k^*\right\|^2 .$$

Because of $\mathrm{vol}(\hat{S}_1) = \mathrm{vol}(S_1) \cdot \left\|\mathbf{b}_{k+1}^*\right\|$, $\mathrm{vol}(\hat{S}_2) = \left\|\mathbf{b}_k^*\right\| \cdot \mathrm{vol}(S_2)$ this inequality can be transformed to $\left(\mathrm{vol}(S_1)/\mathrm{vol}(S_2)\right)^{2/k} \leq (\frac{\gamma_{k+1}}{\delta})^{2\frac{k+1}{k}} \cdot \left\|\mathbf{b}_{k+1}^*\right\|^2 / \left\|\mathbf{b}_k^*\right\|^2$. Combining this with inequality $\left(\mathrm{vol}(S_1)/\mathrm{vol}(S_2)\right)^{2/k} \leq \gamma_k^2 \left\|\mathbf{b}_k^*\right\|^2 / \left\|\mathbf{b}_{k+1}^*\right\|^2$ obtained after the first two exhaustive searches, the ratio $\left(\mathrm{vol}(S_1)/\mathrm{vol}(S_2)\right)^{2/k}$ is lower than $\gamma_k(\gamma_{k+1}/\delta)^{(k+1)/k}$ (or $\gamma_k^2(1+\varepsilon)$ with small ε if δ is near by 1). Asymptotically, Hermite's constants satisfy $\gamma_k \leq \frac{1.744k}{2\pi e}(1+\circ(1))$, so this extended transference reduction provides a ratio $\left(\mathrm{vol}(S_1)/\mathrm{vol}(S_2)\right)^{2/k}$ lower than $\frac{1}{95}k^2$.

Table 3. Comparison between and Schnorr's semi block-$2k$ reduction and Transference reduction. (Here, $\mathrm{SVP}(k+1)$ denotes the cost of finding the shortest lattice vector in dimension $k+1$).

Algorithm	Semi-$2k$ reduction	Transference reduction
Upper bound on $\left\|\mathbf{b}_1\right\|/\mathrm{vol}(L)^{\frac{1}{n}}$	$\approx \beta_k^{\frac{n}{4k}} \lessapprox k^{n \ln 2/2k}$	$\approx \gamma_k^{\frac{n}{2k}} \lessapprox k^{n/2k}$
Upper bound on $\left\|\mathbf{b}_1\right\|/\lambda_1(L)$	$\approx \beta_k^{\frac{n}{2k}} \lessapprox k^{n \ln 2/k}$	$\approx \gamma_k^{\frac{n}{k}} \lessapprox k^{n/k}$
Cost	Poly(size of basis) *HKZ($2k$)	Poly(size of basis) *k*SVP($k+1$)
Reduction of large blocks	HKZ-reduction	Transference reduction

If we use transference reduction instead of Rankin reduction in Algorithm 1, we obtain a reduction algorithm making only $(k+1)$-dimensional exhaustive searches of a shortest vector, and providing an Hermite factor $\left\|\mathbf{b}_1\right\|/\mathrm{vol}(L)^{1/n} \lessapprox \sqrt{\gamma_k}\gamma_k^{n/2k}$ and an approximation factor $\left\|\mathbf{b}_1\right\|/\lambda_1(L) \lessapprox \gamma_k^{-\frac{3}{2}} \frac{4}{3}^{\frac{(3k-1)}{4}} \gamma_k^{\frac{n}{k}}$. These factors are asymptotically not as good as in the semi block-$2k$ reduction, but the exhaustive searches of transference reduction are much cheaper and thus allow to use a larger k. Interestingly the Hermite factor is essentially $\gamma_k^{n/2k}$, which means that the resulting algorithm may roughly be viewed as an algorithmic version of Mordell's inequality [16]: $\gamma_n \leq \gamma_k^{\frac{n-1}{k-1}}$. Similarly, LLL could be viewed as the algorithmic version of Hermite's inequality $\gamma_n \leq \gamma_2^{n-1}$, which is the particular case $k = 2$ of Mordell's inequality.

References

1. M. Ajtai. The worst-case behavior of Schnorr's algorithm approximating the shortest nonzero vector in a lattice. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pages 396–406 (electronic), New York, 2003. ACM.
2. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd STOC*, pages 601–610. ACM, 2001.
3. M. I. Boguslavsky. Radon transforms and packings. *Discrete Appl. Math.*, 111(1-2):3–22, 2001.
4. D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999.
5. D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. In *Proc. of Eurocrypt '99*, volume 1592 of *LNCS*, pages 1–11. IACR, Springer-Verlag, 1999.
6. J. W. S. Cassels. *Rational quadratic forms*, volume 13 of *London Mathematical Society Monographs*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1978.
7. H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1995. Second edition.
8. J. Conway and N. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1998. Third edition.
9. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. of Cryptology*, 10(4):233–260, 1997. Revised version of two articles from Eurocrypt '96.
10. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics: Study and Research Texts*. Springer-Verlag, Berlin, 1988.
11. C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre. *J. Reine Angew. Math.*, 40:279–290, 1850. Also available in the first volume of Hermite's complete works, published by Gauthier-Villars.
12. N. A. Howgrave-Graham and N. P. Smart. Lattice attacks on digital signature schemes. *Des. Codes Cryptogr.*, 23(3):283–290, 2001.
13. J. L. Lagrange. Recherches d'arithmétique. *Nouveaux Mémoires de l'Académie de Berlin*, 1773.
14. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.
15. L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50. SIAM Publications, 1986. CBMS-NSF Regional Conference Series in Applied Mathematics.
16. J. Martinet. *Les réseaux parfaits des espaces euclidiens*. Masson, Paris, 1996.

17. D. Micciancio and S. Goldwasser. *Complexity of lattice problems.* The Kluwer International Series in Engineering and Computer Science, 671. Kluwer Academic Publishers, Boston, MA, 2002. A cryptographic perspective.
18. J. Milnor and D. Husemoller. Symmetric bilinear forms. *Math. Z*, 1973.
19. P. Q. Nguyen and I. E. Shparlinski. The insecurity of the digital signature algorithm with partially known nonces. *J. Cryptology*, 15(3):151–176, 2002.
20. P. Q. Nguyen and D. Stehlé. Floating-point LLL revisited. In *Proc. of Eurocrypt '05*, volume 3494 of *LNCS*, pages 215–233. IACR, Springer-Verlag, 2005.
21. P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Proc. of CALC '01*, volume 2146 of *LNCS*. Springer-Verlag, 2001.
22. A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *Proc. of Cryptology and Computational Number Theory*, volume 42 of *Proc. of Symposia in Applied Mathematics*, pages 75–88. AMA, 1989.
23. R. A. Rankin. On positive definite quadratic forms. *J. London Math. Soc.*, 28:309–314, 1953.
24. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
25. C. P. Schnorr. A more efficient algorithm for lattice basis reduction. *J. of algorithms*, 9(1):47–62, 1988.
26. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, 66:181–199, 1994.
27. C. P. Schnorr and H. H. Hörner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Proc. of Eurocrypt '95*, volume 921 of *LNCS*, pages 1–12. IACR, Springer-Verlag, 1995.
28. J. L. Thunder. Higher-dimensional analogs of Hermite's constant. *Michigan Math. J.*, 45(2):301–314, 1998.

A Proof of Theorem 2

The right-hand product (8) of Hermite's constants can be bounded using the absolute upper bound $\gamma_j \leq (1+j)/4$ by: $\beta_k^k \leq \prod_{j=0}^{k-1}(1+\frac{k+j+1}{4})^{\frac{2k}{k+j}}$.

$$\beta_k^k \leq \left(1+\frac{k}{2}\right)^{\sum_{j=0}^{k-1}\frac{2}{1+j/k}} \cdot \prod_{j=0}^{k-1}\left(\frac{1+\frac{k+j+1}{4}}{1+\frac{k}{2}}\right)^{\frac{k}{k+j}}.$$

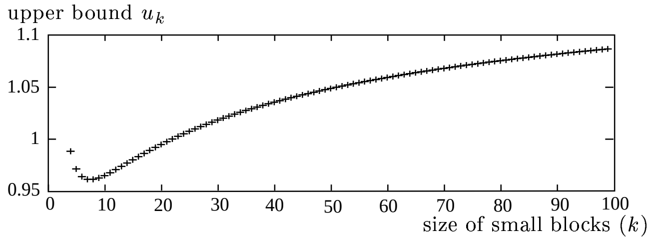
The first sum can be compared with an integral:

$$\frac{1}{k}\sum_{j=0}^{k-1}\frac{2}{1+j/k} \leq \frac{1}{k} + 2\int_0^1 \frac{1}{1+x}dx,$$

and the last product is always smaller than 1: more precisely, its asymptotical equivalent is $\exp\left(-k(\ln 2/2)^2\right) \approx 0.887^k$. Hence we obtain the absolute upper bound: $\beta_k^k \leq (1+\frac{k}{2})^{1+2k\cdot\ln 2}$ or $\beta_k \leq (1+\frac{k}{2})^{1/k+2\ln 2}$.

If we use the best asymptotical bound known for Hermite's constant $\gamma_j \leq \frac{1.744n}{2\pi e}(1+\circ(1))$, we obtain using the same argument, the asymptotical upper bound:

$$\beta_k \leq \exp(-k(\ln 2/2)^2)\left(\frac{1.744k}{\pi e}\right)^{1/k+2\ln 2} \leq \frac{1}{10}k^{2\ln 2}.$$



This curve shows the numerical upper bound u_k of $\ln \prod_{j=0}^{k-1} \gamma_{k+j+1}^{2/(k+j)} / \ln k$, obtained by using the exact values of Hermite's constant γ_i for $1 \le i \le 8$ and $i = 24$, and the bound $\gamma_i \le 1 + \frac{i}{4}$ elsewhere. Thus $u_k \le 1.1$ for $1 \le k \le 100$.

Fig. 2. "Exponents" of the upper-bound on β_k

B Proof of Theorem 3

The Stirling equivalent of Γ satisfies $0 \le \ln(\Gamma(x+1)) - \ln\left((\frac{x}{e})^x \sqrt{2\pi x}\right) \le \frac{K}{x}$ where $K < 0.0835$ is a constant. Since the function $k \to k^{-j}$ is decreasing for $j \ge 2$, we may compare its integral with ζ, and we deduce the following bound:

$$\zeta(j) \le 1 + \frac{1}{2^j} + \frac{1}{(j+1)2^{j+1}}$$

Combining these two relations, we obtain the following upper bound for $Z(j)$:

$$\ln(Z(j)) \le \frac{j}{2} \ln \frac{j}{2} - \frac{j}{2}(\ln \pi + 1) + \rho(j)$$

where $\rho(j) = \left(1 - \ln\left(\frac{j}{2} - 1\right) + \frac{1}{2}\ln\left(2\pi\left(\frac{j}{2} - 1\right)\right) + \frac{K}{(\frac{j}{2}-1)}\right) + \left(\frac{1}{2^j} + \frac{1}{(j+1)2^{j+1}}\right)$. For $j > 13$, we have $\rho(j) < 0$, therefore it can be removed from the upper bound.

$$\forall j \ge 13,\ \ln(Z(j)) \le \frac{j}{2} \ln \frac{j}{2} - \frac{j}{2}(\ln \pi + 1)$$

The lower bound is a consequence of Stirling's formula and the relation $1 \le \zeta(j)$.

$$\forall j \ge 13,\ \left(\frac{j}{2} - 1\right) \ln\left(\frac{j}{2} - 1\right) - \frac{j}{2}(\ln(\pi) + 1) \le \ln(Z(j)) \le \frac{j}{2} \ln \frac{j}{2} - \frac{j}{2}(\ln \pi + 1)$$

We now use the upper bound and an integral to bound the denominator $\prod_{j=2}^{k} Z(j)$:

$$\begin{aligned}\sum_{j=2}^{k} \ln(Z(j)) &\le \sum_{j=2}^{13} \ln(Z(j)) + \int_{14}^{k+1} \left(\frac{t}{2} \ln \frac{t}{2} - \frac{t}{2}(\ln \pi + 1)\right) dt \\ &\le \frac{(k+1)^2}{4}\left(\ln(k+1) - \ln \pi - \frac{1}{2} - \ln 2\right) + c\end{aligned}$$

where $c = \sum_{j=2}^{13} \ln(Z(j)) - \frac{225}{4}(3\ln 2 - \ln\pi - \frac{1}{2} - \ln 2)$. And we apply the lower bound on the numerator $\prod_{j=n-m+1}^{n} Z(j)$:

$$\begin{aligned}\sum_{j=k+1}^{2k} \ln(Z(j)) &\geq \int_k^{2k} \left(\left(\frac{t}{2}-1\right)\ln\left(\frac{t}{2}-1\right) - \frac{t}{2}(\ln(\pi)+1)\right) dt \\ &\geq k^2\left(\ln(k-1) - \frac{1}{4}\ln(k-2) + \frac{\ln 2}{4} - \frac{9}{8} - \frac{3}{4}\ln\pi\right) + r(k)\end{aligned}$$

where $r(k) = k\ln(k-2) - 2\ln(k-1) - \ln 2 + \frac{1}{2}k + \ln 2 - \ln(k-2) + \ln(k-1)$ is equivalent to $r(k) \sim -k \cdot \ln k$. Finally, we obtain a lower bound for $\gamma_{2k,k}$:

$$\ln \gamma_{2k,k}^{k} \geq \frac{k^2}{2}\ln(k) + \left(\frac{\ln 2}{2} - 1 - \frac{\ln\pi}{2}\right)k^2 + s(k)$$

where $s(k) = r(k) - k^2\left(\ln(\frac{k-1}{k}) + \frac{1}{4}\ln(\frac{k-2}{k})\right) - (\frac{2k+1}{4})\left(\ln(k+1) - \ln\pi - \frac{1}{2} - \ln 2\right) - \frac{1}{4}k^2\ln(\frac{k-1}{k}) + (-49\ln 7 + \frac{195}{4}\ln\pi - \ln 2/4 + \frac{585}{8} - \sum_{j=2}^{13}\ln(Z(j)))$. We can show that this function is equivalent to $-\frac{3}{2}k\ln k$, and that for $k > 100$, $\left|s(k)/k^2\right| \leq 0.06$. As a final step, we multiply the result by $2/k^2$ and apply exponentiation to obtain the bound $(\gamma_{2k,k})^{\frac{2}{k}} \geq \frac{k}{12}$ for all $k > 100$. Note that we already had the bound $\frac{k}{9}$ for $k \leq 100$ using numerical computation of $Z(j)$. Asymptotically, we have obtained the following lower bound: $(\gamma_{2k,k})^{\frac{2}{k}} \geq \frac{2k}{\pi e^2}$.