Speeding-up Lattice Reduction with Random Projections (Extended Abstract)

Ali Akhavi[1] and Damien Stehlé[2]

[1] Université de Caen/GREYC, Bd Maréchal Juin, F-14032 Caen Cedex, France.
ali.akhavi@info.unicaen.fr – http://users.info.unicaen.fr/~akhavi
[2] CNRS/LIP/INRIA/ENS/UCBL, 46 allée d'Italie, F-69364 Lyon Cedex 07, France.
damien.stehle@ens-lyon.fr – http://perso.ens-lyon.fr/damien.stehle

Abstract. Lattice reduction algorithms such as LLL and its floating-point variants have a very wide range of applications in computational mathematics and in computer science: polynomial factorization, cryptology, integer linear programming, *etc.* It can occur that the lattice to be reduced has a dimension which is small with respect to the dimension of the space in which it lies. This happens within LLL itself. We describe a randomized algorithm specifically designed for such rectangular matrices. It computes bases satisfying, with very high probability, properties similar to those returned by LLL. It significantly decreases the complexity dependence in the dimension of the embedding space. Our technique mainly consists in randomly projecting the lattice on a lower dimensional space, by using two different distributions of random matrices.

1 Introduction

A *lattice* L is a set of integer linear combinations of some linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d \in \mathbb{R}^n$. These vectors are called a *basis* of the lattice. A given lattice has infinitely many bases, but their cardinality d is always the same: it is called the lattice *dimension*. The dimension n of the basis vectors is called the *degree* of the lattice. The degree of a given lattice cannot be smaller than its dimension. In this article, we are interested in lattices whose degrees are much higher than their dimensions: we will informally call *rectangular* such lattices. When the degree and the dimension match, the lattice is *full-dimensional*.

Lattices are an algorithmic tool that proved crucial in many areas in computer science and mathematics, ranging from cryptology [12, 1, 19] to computer arithmetic [6, 7, 24] and algorithmic number theory [20, 11]. They became popular in 1982, when Arjen Lenstra, Hendrik Lenstra Jr, and László Lovász introduced the renowned algorithm now known under the acronym LLL [17]. Given a lattice basis made of integer vectors, the LLL algorithm discloses a short non-zero lattice vector in time polynomial in the bit-size of the input. This algorithm has complexity $O(d^5 n \log^3 B)$, where B is the maximum of the norms of the input vectors. The practical variants of LLL rely on floating-point arithmetic (for the underlying Gram-Schmidt orthogonalization), and the best fully proved such variant is due to Nguyen and Stehlé [18]. The so-called L^2 algorithm has bit-complexity $O(d^4 n \log B(d + \log B))$. We will consider this variant here, though the technique we introduce works with any other variant of LLL.

Our main result is to provide a randomized algorithm taking as input a lattice basis and computing another basis such that with overwhelming probability (e.g., as d grows to infinity) this basis satisfies properties similar to those returned by LLL. If one neglects all terms polynomial in $\log d, \log n$ and $\log\log B$, then it runs in time $\tilde{O}(d^2(d^3+n)\log B(d+\log B))$: the cost dependence in the degree of the lattice is considerably weakened. Moreover, the bit-size of the integers involved in the algorithm is essentially the same as the bit-size of the initial basis. A simpler strategy than the one we develop is based on the Gram matrix (the symmetric matrix of the pairwise inner-products): one can compute the LLL-transformation by reducing the Gram matrix. It is deterministic and decreases the cost dependence in n, but it suffers from two drawbacks: the bit-sizes of the entries of the Gram matrix are essentially twice bigger than the ones of the input basis and the floating-point inaccuracies can be significantly larger if we start from the Gram matrix. Strong heuristics [22] tend to show that one can use half the precision required by the L^2 algorithm by disregarding the Gram matrix.

Rectangular lattices arise in the two following situations. First of all, they sometimes occur in Coppersmith's methods to find small roots of polynomials over the integers and modulo an integer [12]. These methods have numerous applications in cryptology. The involved lattice bases are full-dimensional but highly structured. This structure sometimes creates situations where subsets of the input basis vectors suffice to provide the short vectors found by LLL. The number of vectors to be considered can be drastically decreased, while their embedding dimension remains constant, thus creating rectangular lattices. This arises in [4], where Coppersmith's method is used to cryptanalyse RSA when the secret exponent is unusually small, and in [23], where it is used to find bad cases for the rounding of mathematical functions, in the field of computer arithmetic. In [4], the ratio between the degree and the dimension is constant, while in [23] the degree grows as the square of the dimension. Another context where rectangular lattices arise is the LLL algorithm itself (and most of its variants, including L^2), even for full-dimensional lattices. In LLL, the basis is reduced incrementally. There is a main loop whose main parameter is an index k. The meaning of this index is that in the current basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$, the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k-1}$ are already LLL-reduced and one is trying to extend this property to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$. At the beginning of the execution, the index k is set to 2, while at the end it reaches $d+1$. As long as the index k has not been beyond some arbitrary k_0, we are in fact applying LLL to the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{k_0} \in \mathbb{Z}^n$. The smaller the considered k_0, the more rectangular the lattice being reduced. Our technique may be used within LLL to speed it up by a constant factor.

To achieve the result, we develop a few tools, which may be of independent interest. Firstly, we decrease the degree of the lattice by applying a random projection technique: we multiply the $n \times d$ input basis matrix by a random $d \times n$ matrix, and show that by reducing the randomly projected lattice we get useful information for the initial lattice with very high probability. This resembles the famous Johnson-Lindenstrauss theorem [15], which shows that one can randomly map N vectors in a $O(\log N)$-dimensional space without modifying significantly

the pairwise distances between the vectors. We cannot directly apply such a method since we do not consider the input vectors solely, but their infinitely many integer linear combinations (i.e., in our case N would be infinite). Moreover, we need to keep the Euclidean structure of the initially spanned vector space. In particular, we do not decrease the degree of our lattice below its dimension.

In this paper we consider two random projections. In both models each row of the projection matrix are chosen independently with a common distribution μ_n. In the first model, called the Gaussian model, $\nu_n = \mathcal{N}(0, I_n)$, the standard normal distribution. In other words, each entry of the projection matrix is sampled independently with the standard normal distribution $\mathcal{N}(0, 1)$. These random matrices have been studied extensively and we will rely on a result about their condition numbers, due to Chen and Dongarra [9]. In the second random model, called the unit ball model, ν_n is the uniform distribution inside $\mathcal{B}_n(\mathbf{0}, 1)$, the n-dimensional ball of radius 1 that is centered in $\mathbf{0}$. So each row of the projection matrix is sampled uniformly and independently inside $\mathcal{B}_n(\mathbf{0}, 1)$. Such random matrices have been already studied in [13, 2, 3]. We will rely on some of the results of these papers. Notice that Rouault [21] recently studied the asymptotic behavior of the determinant of the lattice generated by the rows of a rectangular random matrix with both distributions that we consider here.

All the proofs in this paper are done in continuous random models, i.e. entries of our random matrices are real numbers, which is unsuitable to devise an algorithm. In practice, random matrices are generated with the associated discretised law. Due to space limitation, we chose to skip these difficulties and to to describe them in the full version of the paper.

We performed tests on our reduction technique. They worked very well for many different classes of random projections, including easily samplable ones (such as entries chosen independently and uniformly in $\{-1, 0, 1\}$). As theoretically predicted, the speed-ups can be made arbitrarily large by increasing the ratio between the lattice degree and the lattice dimension.

Related work. Chen and Storjohann [10] introduced in 2005 a probabilistic technique to compute a reduced basis of a lattice given by a generating family: one is given more vectors than the lattice dimension. Our work can be seen as dual to theirs. We deal with vertically rectangular matrices by multiplying them on the left by a random matrix, whereas they deal with horizontally rectangular matrices by multiplying them on the right by a random matrix. They use the arithmetic structure of the lattice whereas we consider its geometric embedding. The two techniques may be used together.

Road-map of the paper. In Section 2, we provide the necessary background on lattices. In Section 3, the core of the paper, we describe our randomized algorithm and perform its complexity analysis. Section 4 is devoted to show its correctness with two different sources of random matrices. Finally, in Section 5, we describe our experiments.

Notations. All costs are given for the bit-complexity model and we assume that we have a perfect source of random bits. We use only naive arithmetic and

naive linear algebra. The results may be improved by using fast arithmetic and fast linear algebra. We let $\mathcal{B}_n(\boldsymbol{a}, R)$ denote the n-dimensional ball of radius R centered in $\boldsymbol{a}$. If B is a matrix, we denote by $L(B)$ the lattice spanned by its columns. We denote by $\|B\|_2$ the matrix norm induced by the Euclidean norm, also called the spectral. The maximum of the absolute values of B's entries is the usual max norm denoted by $\|B\|$.

2 Some Reminders on Lattices

We refer to [8] and [11] for comprehensive introductions to lattices and their computational aspects. We give below only the material that is necessary to the description and proof of our probabilistic reduction technique.

Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d \in \mathbb{R}^n$ be linearly independent vectors. Their *Gram-Schmidt orthogonalization* is defined as follows: the vector $\boldsymbol{b}_i^*$ is the component of the vector $\boldsymbol{b}_i$ which is orthogonal to the linear span of the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{i-1}$. We have $\boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \frac{r_{j,i}}{r_{j,j}} \boldsymbol{b}_j^*$ where $r_{j,i} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\|\boldsymbol{b}_j^*\|}$. If B is a full-rank $n \times d$ matrix, its *QR-factorization* is the unique pair of matrices (Q, R) such that $B = Q \cdot R$, Q is an $n \times d$ matrix made of orthonormal column vectors and R is an upper triangular $d \times d$ matrix with positive diagonal coefficients. The Gram-Schmidt orthogonalization and the QR-factorization of the matrix made of the $\boldsymbol{b}_i$'s are closely related: the i-th column of Q is $\frac{\boldsymbol{b}_i^*}{\|\boldsymbol{b}_i^*\|}$ and the matrix R is made of the $r_{i,j}$'s.

Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d$ and $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_d$ be two bases of the same lattice. If B and C are the matrices whose columns are the $\boldsymbol{b}_i$'s and $\boldsymbol{c}_i$'s, then there exists a $d \times d$ integer matrix T of determinant ± 1 such that $B = C \cdot T$. Such a matrix is called *unimodular*. Moreover, if two matrices can be obtained one another by unimodular matrices, their columns span the same lattice. Let L be a lattice. The length of any shortest non-zero vector is called the lattice *minimum* and denoted by $\lambda(L)$.

Consider the $\boldsymbol{b}_i$'s as a basis of a lattice L. The *determinant* of L is defined by $\det L = \prod_{i=1}^{d} \|\boldsymbol{b}_i^*\|$. This does not depend on the choice of the basis. Hadamard's inequality gives that $\det L \leq \prod_{i=1}^{d} \|\boldsymbol{b}_i\|$. Let $\delta \in (1/4, 1]$ and $\eta \in [1/2, \sqrt{\delta})$. The $\boldsymbol{b}_i$'s are said (δ, η)-*LLL-reduced* if for any $i < j$, we have $|r_{i,j}| \leq \eta \cdot r_{i,i}$, and for any i, we have $\delta \cdot r_{i-1,i-1}^2 \leq r_{i,i}^2 + r_{i-1,i}^2$. When introduced in [17], LLL-reduction referred to the pair $(3/4, 1/2)$. The vectors of a LLL-reduced basis are relatively short. In particular, we have $\|\boldsymbol{b}_1\| \leq (\delta - \eta^2)^{-\frac{d-1}{4}} (\det L)^{\frac{1}{d}}$ and $\prod_{i=1}^{d} \|\boldsymbol{b}_i\| \leq (\delta - \eta^2)^{-\frac{d(d-1)}{4}} (\det L)$. We refer to [18] for a proof of this fact and for the cost of the algorithm mentioned in the following theorem. The property on the unimodular transformation matrix is classical and a proof can be found in [16].

Theorem 1. *Let $\eta \in (1/2, 1)$ and $\delta \in (\eta^2, 1)$. There exists an algorithm such that when given as input any linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d \in \mathbb{Z}^n$ it computes a (δ, η)-LLL-reduced basis of the lattice they span in time $O(d^4 n (d + \log B) \log B)$, where $B = \max_i \|\boldsymbol{b}_i\|$. Furthermore, the bit-lengths of the entries of the transformation matrix are bounded by $O(d \log B)$.*

3 Probabilistic Reduction of Rectangular Lattices

3.1 High-Level Description of the Algorithm

We are given an $n \times d$ integer matrix B and try to find a small integer linear combination of its columns. Instead of applying an LLL-type algorithm directly, we apply a random $d \times n$ dimensional projection P to the matrix and LLL-reduce the $d \times d$ projected matrix $B' = P \cdot B$. By doing so, we decrease the cost with respect to n. We wish that with high probability the unimodular transformation T obtained by LLL-reducing B' somehow reduces B as well. Figure 1 sums up the general method. The top arrow is computationally expensive and is approximately and probabilistically simulated by the succession of plain arrows, that are cheaper. The main result of the paper is the theorem following the description of the algorithm.

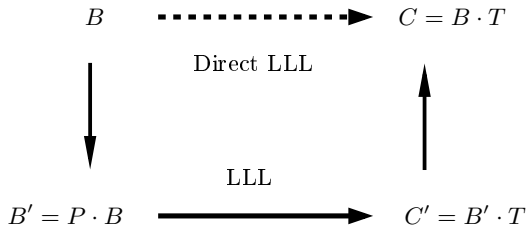

Fig. 1. High-level description of the algorithm of Figure 2

Input: A lattice basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d) \in \mathbb{Z}^{n \times d}$.
Output: Another basis of the same lattice, hopefully made of short vectors.
Parameters: (δ, η) such that $\eta \in (1/2, 1)$ and $\delta \in (\eta^2, 1)$.
1. Generate a random $d \times n$ matrix P with a fixed distribution.
2. Compute $B' = P \cdot B$.
3. Compute $C' = \mathrm{LLL}_{\delta,\eta}(B')$.
4. Compute $T = (B')^{-1} \cdot C'$.
5. Return $B \cdot T$.

Fig. 2. Probabilistic reduction of a rectangular lattice

Theorem 2. *Let $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d) \in \mathbb{Z}^{n \times d}$ be a basis of a lattice L with $B = \max \|\boldsymbol{b}_i\|$. The algorithm of Figure 2 will compute a basis $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_d)$ of L with the expected running time:*

$$O\left(d^5 \log nB(d + \log nB) + d^2 n \log nB(\log nB + d \log\log nB)\right).$$

If the entries of the random matrix P are independent Gaussian random variables, then for all $x < 1$ then with probability greater than $1 - x$,

1. *The vector $\boldsymbol{c}_1$ satisfies $\|\boldsymbol{c}_1\| \leq \frac{2^8 d^2}{x^3}(\delta - \eta^2)^{-\frac{d-1}{4}} \cdot (\det L)^{\frac{1}{d}}$.*
2. *The basis $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_d)$ satisfies $\prod_{i \leq d} \|\boldsymbol{c}_i\| \leq \left(\frac{2^8 d^2}{x^3}(\delta - \eta^2)^{-\frac{d-1}{4}}\right)^d \cdot (\det L)$.*

If the rows of the random matrix P are independent random vectors each one picked up uniformly inside the n dimensional unit ball then for any d, there exists $n_0(d)$ such that for any $n \geq n_0(d)$, with probability greater than $1 - 2^{-d}$,

1. *The vector $\boldsymbol{c}_1$ satisfies $\|\boldsymbol{c}_1\| \leq 2^{4d}(\det L)^{1/d}$.*
2. *The basis $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_d)$ satisfies $\prod_{i \leq d} \|\boldsymbol{c}_i\| \leq 2^{4d^2}(\det L)$.*

Notice that one can take $x = 2^{-d}$ and obtain that with probability exponentially close to 1 the output still satisfies properties similar to what would have been returned by LLL. On both model the length of the first vector may also be expressed as an approximation of the first minimum of the lattice by a factor similar to what would have been returned by LLL. Subsection 3.2 proves the complexity statement and Section 4 the correctness in the continuous models.

3.2 Complexity Analysis

We now prove the complexity statements of Theorem 2. We assume the reader is familiar with the Chinese Remainder Theorem (CRT for short). We refer to [14] for an introduction to the CRT.

From the previous subsection, we know that Step 1 of the algorithm of Figure 2 costs $O(dn \log n)$ bit operations. Step 2 is a multiplication of a $d \times n$ matrix whose entries have length $O(\log n)$ with an $n \times d$ matrix whose entries have length $O(\log B)$. The entries of the $d \times d$ matrix B' have length $O(\log nB)$. The matrix multiplication is performed with the CRT. One takes $O\left(\frac{\log nB}{\log\log nB}\right)$ prime numbers, each of them of length $O(\log\log nB)$. The construction of the primes is computationally negligible. The matrix multiplications modulo the primes cost $O(d^2 n \log nB \log\log nB)$. The conversions of the input matrices into matrices modulo the primes cost $O(dn \log^2 nB)$, whereas the conversion of the output matrices modulo the primes into the integer matrix B' costs $O(d^2 \log^2 nB)$. Theorem 1 gives us that Step 3 costs $O\left(d^5 \log nB(d + \log nB)\right)$. At Step 4, we use again the CRT. Thanks to Theorem 1, we know that the entries of the matrix T have length $O(d \log nB)$. By an analysis similar to the one developed for Step 2, we get that the cost is bounded by $O\left(d^4 \log nB(\log d + \log\log nB) + d^4 \log^2 nB\right)$. At Step 5, we have to multiply an $n \times d$ matrix whose entries have length $O(\log B)$ with a $d \times d$ matrix whose entries have length $O(d \log nB)$. We split each entry of the matrix T into d blocks of roughly $\Theta(\log nB)$ bits, which gives rise to d matrices of dimensions $d \times d$ and whose entries have length $O(\log nB)$. We thus have d balanced matrix multiplications to perform. For each of them we use the CRT. The overall bit-cost of this step is $O\left(d^3 n \log nB \log\log nB + d^2 n \log^2 nB\right)$. This concludes the proof for the bit-complexity bound of the algorithm of Figure 2 claimed by Theorem 2.

4 Probabilistic correctness in two continuous models

We consider an input basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)$ given by an $n \times d$ matrix B. Let $B = Q_B R_B$ be its QR-factorization. Let P be a $d \times n$ random matrix, either from the Gaussian model or from the unit ball model. Let $B' = P \cdot B$ and P' the $d \times d$

matrix $P \cdot Q_B$. We are to show that, with high probability, if an integer linear combination of the columns of $B' = P'R_B$ is a short vector of the lattice $L(B')$, then the same combination of columns of B will be a short vector in $L(B)$. Let $\boldsymbol{c}' \in L(B')$ be defined by $\boldsymbol{c}' = B'\boldsymbol{x} = P'R_B\boldsymbol{x}$, with $\boldsymbol{x} \in \mathbb{Z}^d$. Let $\boldsymbol{c}$ be defined by the same linear combination of the $\boldsymbol{b}_i$'s: $\boldsymbol{c} = B\boldsymbol{x} = Q_B R_B \boldsymbol{x}$.

Our goal is to compare the ratios $\frac{\|\boldsymbol{c}'\|}{(\det L(B'))^{1/d}}$ and $\frac{\|\boldsymbol{c}\|}{(\det L(B))^{1/d}}$. Lemma 1 provides an upper bound to $\det L(B')/\det L(B)$ which holds with high probability. Moreover if $\boldsymbol{c}' \in L(B')$ is the first vector of the basis output by LLL, then $\|\boldsymbol{c}\| \leq 2^{O(d)}(\det L(B'))^{\frac{1}{d}}$. To compare $\|\boldsymbol{c}'\|$ and $\|\boldsymbol{c}\|$, we proceed as follows.

Since the columns of Q_B are orthonormal, we have $\|\boldsymbol{c}\| = \|R_B\boldsymbol{x}\|$. We get $\|\boldsymbol{c}\| = \|(P')^{-1}\boldsymbol{c}'\| \leq \left\|(P')^{-1}\right\|_2 \cdot \|\boldsymbol{c}'\|$. Lemma 3 provides an upper bound to $\left\|(P')^{-1}\right\|_2$ which also holds with high probability in the Gaussian model.

Similarly, if $(\boldsymbol{c}'_1, \ldots, \boldsymbol{c}'_d)$ is an LLL-reduced basis of $L(B')$, then $\prod_{i\leq d}\|\boldsymbol{c}'_i\| \leq 2^{O(d^2)} \det L(B')$. If $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_d)$ is the basis of $L(B)$ where any $\boldsymbol{c}_i$ is expressed in terms of the input basis B with the same integer linear combination than $\boldsymbol{c}'_i$ in terms of B', then: $\prod_{i=1}^d \|\boldsymbol{c}_i\| = \prod_{i=1}^d \|(P')^{-1}\boldsymbol{c}'_i\| \leq \left\|(P')^{-1}\right\|_2^d \cdot \prod_{i=1}^d \|\boldsymbol{c}'_i\|$.

To achieve computations in the unit ball model, we decompose once more the matrix P': let $P' = R_{P'}Q_{P'}$ be the transpose of the QR-decomposition of $(P')^t$. Since the rows of $Q_{P'}$ are orthonormal, we have $\|\boldsymbol{c}\| = \|Q_{P'}R_B\boldsymbol{x}\|$. We get $\|\boldsymbol{c}\| = \|(R_{P'})^{-1}\boldsymbol{y}\| \leq d\left\|(R_{P'})^{-1}\right\| \cdot \|\boldsymbol{c}'\|$. Analogously to the previous case, Lemma 5 provides an upper bound to $\left\|(R_{P'})^{-1}\right\|$ in the unit ball model. There is also an analogous upper bound for $\prod_{i=1}^d \|\boldsymbol{c}_i\|$ that is $d^d\left\|(R_{P'})^{-1}\right\|^d \prod_{i=1}^d \|\boldsymbol{c}_i\|$.

Notice that in Theorem 2, one could also compare the first vector output by our algorithm with the first minimum of the lattice (as it is usually done in the LLL case): we use the facts that $|\boldsymbol{c}_1\| \leq \|(P')^{-1}\|_2 \cdot \|\boldsymbol{c}'_1\|$, $\|\boldsymbol{c}'_1\| \leq 2^{O(d)} \cdot \lambda(L'(B))$ and $\lambda(L'(B)) \leq \|P'\|_2 \cdot \lambda(L(B))$. For the last inequality, consider $\boldsymbol{s} \in \mathbb{Z}^d$ such that $\|B\boldsymbol{s}\| = \lambda(L(B))$. For the same reasons as above, $\|B'\boldsymbol{s}\| \leq \|P'\|_2 \cdot \|B\boldsymbol{s}\|$. It now suffices to see that $\lambda(L(B')) \leq \|B'\boldsymbol{s}\|$.

Lemma 1. *For any $\lambda > 1$, the following holds with probability at least $1 - 1/\lambda^2$:*

(i) In the Gaussian model, $(\det L(B'))^2 \leq d^d \cdot (1 + 3\lambda) \cdot (\det L(B))^2$.
(ii) In the unit ball model, $(\det L(B'))^2 \leq \frac{d!}{(n+2)^d} \cdot (1 + 2d\lambda) \cdot (\det L(B))^2$.

Proof. We have $B' = P \cdot B = P \cdot Q_B \cdot R_B$, which gives that $\det L(B') = \det(P \cdot Q_B) \det R_B = \det(P \cdot Q_B) \det L(B)$. It thus suffices to focus on the determinant of the $d \times d$ matrix $P' = P \cdot Q_B$.

Notice first that the matrix Q_B can be extended to an $n \times n$ orthogonal matrix $Q'_B = (Q_B|\cdot)$. We are interested in $P \cdot Q_B$, i.e., the $d \times d$ left sub-matrix of $P \cdot Q'_B$. Since the both distributions of P that we consider are invariant under right multiplication by an orthonormal matrix, the random matrices P and $P \cdot Q'_B$ follow the same distribution. The distribution of $P \cdot Q_B$ is thus the same as the distribution of the left $d \times d$ sub-matrix of P, denoted by P_l.

Proof of (i). The random matrix P is Gaussian. Let the rows of the $d \times d$ left sub-matrix of P be denoted by $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_d$. Thanks to Hadamard's inequality, we have $\det P' \overset{(d)}{=} \det P_l \leq \prod_{i=1}^d \|\boldsymbol{p}_i\|$. Let X be $\prod_{i=1}^d \|\boldsymbol{p}_i\|$.

Any $\|\boldsymbol{p}_i\|^2$ is the sum of d squared independent Gaussians. Thus $\mathbb{E}(\|\boldsymbol{p}_i\|^2) = d$ and $\mathbb{E}(\|\boldsymbol{p}_i\|^4) = d(d+2)$. Since they are independent, one gets:

$$\mathbb{E}(X^2) = d^d \quad \text{and} \quad \sigma(X^2) = \mathbb{E}(X^2) \cdot \sqrt{f(d)},$$

where $f(d) = \left(\frac{d+2}{d}\right)^d - 1 \leq 9$. Chebyshev's inequality gives that for $\lambda > 0$:

$$\mathbb{P}\left\{X^2 - \mathbb{E}(X^2) > 3\lambda \mathbb{E}(X^2)\right\} \leq 1/\lambda^2.$$

Proof of (ii). Now P is distributed under the unit ball model. Let H be a d-dimensional linear subspace. Consider the distribution of the orthogonal projections of the rows of P onto H. Since the distribution of the rows of P is invariant under rotation, the distribution of their orthogonal projections is the same no matter onto which subspace H the projection is performed. Let us pick up $n-d$ additional vectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-d}$ in the n-dimensional unit ball and let H be the orthogonal coset of the (almost surely $(n-d)$-dimensional) space spanned by these additional vectors: $H = <\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-d}>^{\perp}$. Let the rows of P be denoted by $\boldsymbol{p}_{n-d+1}, \ldots, \boldsymbol{p}_n$. Let us denote by $\boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_n^*$ the Gram-Schmidt orthogonalization of the random vectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$. We then have $\det(P \cdot Q_B) = \prod_{i=n-d+1}^{n} \|\boldsymbol{p}_i^*\|$. Let X be the random variable corresponding to $\det(P \cdot Q)$. It is proved in [13] that the $\|\boldsymbol{p}_i^*\|^2$'s are independent random variables and that their distribution is given by $\|\boldsymbol{p}_i^*\|^2 \overset{(d)}{=} \beta\left(\frac{n-i+1}{2}, \frac{i+1}{2}\right)$. The Beta law is classical in probability theory and its moments are well known:

$$\mathbb{E}(\|\boldsymbol{p}_i^*\|^2) = \tfrac{n-i+1}{n+2} \quad \text{and} \quad \mathbb{E}(\|\boldsymbol{p}_i^*\|^4) = \tfrac{(n-i+1)(n-i+3)}{(n+4)(n+2)}.$$

Then the independence of $\|\boldsymbol{p}_i^*\|^2$'s leads to:

$$\mathbb{E}(X^2) = d!/(n+2)^d \quad \text{and} \quad \sigma^2(X^2) = \mathbb{E}(X^2) \cdot \sqrt{f(d,n)},$$

where $f(d,n) = \frac{(d+1)(d+2)}{2}\left(\frac{n+2}{n+4}\right)^d - 1$. By routine computation, one sees that $\sqrt{f(d,n)} \leq 2d$ and conclude thanks to Bienaymé's inequality. □

4.1 Probabilistic correctness in the Gaussian model

The correctness claims of Theorem 2 derive from Lemmas 1 and 3. To bound the quantity $\|(P')^{-1}\|$, we use the following result on the condition number of a Gaussian random matrix.

Lemma 2 ([9]). *Let κ be the condition number of the matrix P', i.e., $\|P'\| \cdot \|(P')^{-1}\|$. Then for any $\lambda \geq 1$, the probability that $\kappa > \lambda d$ is smaller than $4/\lambda$.*

The last ingredient to the proof of correctness of theorem 2 is the following.

Lemma 3. *Let $t < 1$. Then $\left\|(P')^{-1}\right\| \leq 32d/t^2$ holds with probability greater than $1-t$.*

Proof. Let $x < 1/2$. We upper-bound by 1 the density function of the first entry of P'. So with probability greater than $1-2x$, we have $\|P'\|_2 \geq \|P'\|_2 \geq x$. By using Lemma 2, we obtain that with probability greater than $1-2x-4/\lambda$ we have $\|(P')^{-1}\|_2 \leq \lambda d/x$. Setting $x = t/4$ and $\lambda = 8/t$ provides the result. □

By using Lemmas 1 and 3, we see that, with probability greater than $1-t-1/\lambda^2$, the first vector computed by the algorithm of Figure 2 satisfies:

$$\|\boldsymbol{c}_1\| \leq (\delta-\eta^2)^{-\frac{d-1}{4}} \frac{32 \cdot d^{\frac{3}{2}} (1+3\lambda)^{\frac{1}{2d}}}{t^2} \cdot (\det L(B))^{\frac{1}{d}}.$$

By choosing $\lambda = \sqrt{2/x}$ and $t = x/2$, we obtain the result claimed in Theorem 2.

4.2 Probabilistic correctness in the unit ball model

The correctness claims of Theorem 2 derive from Lemmas 1 and 5.

Lemma 4. *Suppose that $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ are n vectors chosen independently and uniformly in the n-dimensional unit ball. Then for any $d \leq n$ and any $v < \frac{1}{4\sqrt{n}}$:*

$$\mathbb{P}\{\min_{1\leq k\leq d} \|\boldsymbol{p}^*_{n-d+k}\| \leq v\} \leq 4\sqrt{n}v.$$

Proof. Let $\ell_i = \|\boldsymbol{p}^*_i\|$. The distributions of the ℓ_i's are given by [13]:

$$\mathbb{P}[l_{n-d+k} \leq v] = \frac{2}{B\left(\frac{d-k+1}{2}, \frac{n-d+k+1}{2}\right)} \int_0^v u^{d-k}(1-u^2)^{\frac{n-d+k-1}{2}} du.$$

Since $1-u^2 \leq 1$, the integral smaller than v^{d-k+1}. Rewriting the denominator in terms of the Gamma function, we get $\mathbb{P}[\ell_{n-d+k} \leq v] \leq \frac{2\ \Gamma\left(\frac{n+2}{2}\right)\ v^{d-k+1}}{\Gamma\left(\frac{d-k+1}{2}\right)\Gamma\left(\frac{n-d+k+1}{2}\right)}$. Using classical properties of the Gamma function, we obtain

$$\mathbb{P}[\ell_{n-d+k} \leq v] \leq 2\left(\frac{nv^2}{2}\right)^{\frac{d-k+1}{2}} \text{ and } \mathbb{P}[\min_{1\leq k\leq d} \ell_{n-d+k} \leq v] \leq 2\sum_{k=1}^{d}\left(\frac{n\ v^2}{2}\right)^{\frac{d-k+1}{2}}.$$

Finally, if $nv^2 \leq 1/2$, we have $\mathbb{P}\left[\min_{1\leq k\leq d} \ell_{n-d+k} \leq v\right] \leq 4\sqrt{n}v$. □

Lemma 5. *Let P be a random matrix chosen as previously. Let $P = RQ$ be the transpose of the QR-decomposition of P^t. Let u and v be two reals satisfying $u < 1\sqrt{3}$ and $v < \frac{1}{4\sqrt{n}}$. For any d there exists n_1 such that for all $n \geq n_1(d,u)$, with probability greater than $1 - d^2(\frac{u^2}{1+u^2})^d - 4\sqrt{n}v$, we have:*

$$\left\|R^{-1}\right\| \leq \frac{1}{v}\ (1+\frac{1}{u})^d.$$

Proof (Sketch). First, notice that as explained in the proof of Lemma 1, the rows of $P' = P \cdot Q_B$ have the same distributions as the projections $\boldsymbol{p}^*_{n-d+1}, \boldsymbol{p}_{n-d+2}[n-d+1], \ldots, \boldsymbol{p}_n[n-d+1]$ of n vectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ chosen independently and uniformly in $\mathcal{B}_n(\mathbf{0}, 1)$ in the orthogonal of the span of the $n-d$ first ones. Let us denote ℓ_i the norm of $\boldsymbol{p}^*_i$. The proof, available in the full version, the previous lemma and classical bounds on the Gamma and Beta functions together with the following tools:

– an asymptotic equivalent for $\mathbb{P}[\ell_{n+j}/\ell_{n+i} < v]$ when n grows to infinity and i and j are two constants. This is available in [2](using the Laplace method for evaluating integrals asymptotically)
– an explicit expression of the coefficients of R_P^{-1} as a function of the coefficients $r_{i,j}$ (using the fact that the matrix R_P is lower triangular and so is R_P^{-1} as well) □

By using Lemmas 1 and 5 after routine computations we see that, with probability greater than $1 - 4\sqrt{n}v - d^2(\frac{u^2}{1+u^2})^d - \lambda^{-2}$, the first vector computed by the algorithm of Figure 2 satisfies:

$$\|\boldsymbol{c}_1\| \leq (\delta - \eta^2)^{-\frac{d-1}{4}} \cdot v \cdot \left(1 + \frac{1}{u}\right)^{d-1} \frac{4 \cdot d^{\frac{3}{2}} \cdot \lambda^{\frac{1}{2d}}}{\sqrt{n+2}} (\det L(B))^{\frac{1}{d}}.$$

Finally we choose $\lambda = 2^{d/2}$, $u = \sqrt{1/8}$, $v = 1/(2^d\sqrt{n})$.

5 Experimental Data

In this section, we report experiments supporting the validity of our method. The experiments are very promising in the sense that the random projection technique seems to work with a wide range of random matrices and seems to perform better than what we proved. Indeed, the output bases are not only made of vectors of small lengths, but LLL terminates very quickly given them as input.

The experiments were performed using Magma [5] V2.14 on an AMD Opteron 2.40GHz. Each figure corresponds to an average over at least ten samples. We used the LLL routine with the default options ($\delta = 0.75, \eta = 0.51$). Magma's LLL is based on the floating-point L^2 algorithm [18]. The Magma code corresponding to our experiments is available under the GPL at: `http://perso.ens-lyon.fr/damien.stehle/DIMREDUCTION.html`. We considered the following families of random projections.

– $\mathcal{R}_1(N)$: each vector is sampled independently in the sphere $\mathcal{B}_n\left(\mathbf{0}, 10^N\right)$. The computations are performed with decimal precision N. The sampling would be uniform if the computations were performed with infinite precision.
– $\mathcal{R}_2(N)$: each entry is Gaussian variate approximated to decimal precision N.
– $\mathcal{R}_3(N)$: each entry is taken uniformly and independently in $\mathbb{Z} \cap \left[-2^N, 2^N\right)$.
– $\mathcal{R}_4$: each matrix entry is taken uniformly and independently in $\{-1, 1\}$.

The matrices to be reduced are generated in the following way. We first create a $d \times d$ random matrix of the following shape:

$$\begin{pmatrix} x_1 & x_2 & \dots & x_d \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix},$$

where the x_i's are chosen uniformly and independently in $[0, B]$ for some fixed B. When B is large enough, the columns form lattice bases that are far from being

reduced. To obtain $n \times d$ lattice bases, we multiply them by matrices sampled from $\mathcal{R}_3(100)$. This provides rectangular bases that are far from being reduced with large and balanced entries. We tested our technique with varying parameters d, n and B and for the classes of random projections described above. We also measured the time LLL takes on the output basis. We compared our technique with the direct LLL approach and with the Gram matrix approach described in the introduction (LLL-reducing the Gram matrix and applying the computed transformation to the input basis). We also compared the lengths of the first vectors of the outputs. The results are described in Figures 3 and 4.

d	20	30	40	50	20	30	40	50
Direct LLL	0.62	8.47	13.6	23.9	1.30	15.8	92.7	341.0
Gram-based approach	0.40	2.26	8.41	25.3	0.52	3.70	16.3	70.7
Random projection approach	0.22	1.19	4.20	13.1	0.25	1.42	5.19	24.8
Direct LLL on the output basis	0.01	0.03	0.07	0.10	0.02	0.09	0.41	1.22

Fig. 3. Timings in seconds of the different LLL approaches for rectangular lattices, when the random matrix is chosen from $\mathcal{R}_4$ and $n = 5d, B = 2^{100 \cdot d}$, (first four columns) and $n = d^2/2, B = 2^{100 \cdot d}$ (last four columns).

Figure 3 shows that the random projection technique can be significantly faster than the direct technique, in particular when n is much larger than d, even if one includes the running-time of LLL on the output basis. Figure 4 shows that the output quality is similar to that of the direct LLL approach. The vector found by the random projection method is most often longer than the one computed by the direct LLL approach, but the ratio remains small. The technique seems to provide reasonably short vectors for all the afore-mentioned families of projections.

d	10	20	30	40	50
$\mathcal{R}_1(100)$	2.34/0.99	3.24/1.06	2.95/1.18	3.90/0.99	5.55/0.88
$\mathcal{R}_2(100)$	3.04/1.02	12.9/1.00	4.13/0.98	4.57/1.00	4.19/0.94
$\mathcal{R}_2(1000)$	3.02/1.04	3.07/0.87	4.40/1.12	5.55/1.16	5.02/1.07
$\mathcal{R}_3(3)$	3.54/1.03	6.67/1.04	3.10/1.02	6.52/0.98	6.21/0.95
$\mathcal{R}_3(10)$	2.98/0.96	2.97/0.99	4.37/1.09	5.58/1.03	3.70/0.99
$\mathcal{R}_4$	3.91/0.96	3.73/1.06	7.00/1.00	4.20/1.03	3.49/1.00

Fig. 4. Ratios between the lengths of the first output vectors after the random projection technique (respectively after LLL on the output basis) and after the direct LLL approach (left of each entry, respectively right of each entry), with $n = 3d$ and $B = 2^{100 \cdot d}$.

ACKNOWLEDGMENTS. We are grateful to Richard Brent, Philippe Flajolet, Guillaume Hanrot, Luis Pardo, Brigitte Vallée and Gilles Villard for helpful discussions. This work was partially funded by the LaRedA project of the Agence Nationale de la Recherche. It was initiated while the first author was hosted within the computer science laboratory of the University of Paris 7 (LIAFA) and completed while the second author was hosted within the Magma group at the University of Sydney.

References

1. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc, of STOC 1997*, pages 284–293. ACM, 1997.
2. A. Akhavi. Random lattices, threshold phenomena and efficient reduction algorithms. *TCS*, 287(2):359–385, 2002.
3. A. Akhavi, J.-F. Marckert, and A. Rouault. On the reduction of a random basis. In *Proc. of the ANALCO'07 New Orleans*. SIAM, 2007.
4. D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Trans Inform Theor*, 46(4):233–260, 2000.
5. W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *JSC*, 24(3–4):235–265, 1997.
6. N. Brisebarre and S. Chevillard. Efficient polynomial L-approximations. In *Proc. of ARITH'18*, pages 169–176. IEEE, 2007.
7. N. Brisebarre and G. Hanrot. Floating-point L2-approximations to functions. In *Proc. of ARITH'18*, pages 177–186. IEEE, 2007.
8. J. W. S. Cassels. *An Introduction to the Geometry of Numbers*. Springer, 1971.
9. Z. Chen and J. Dongarra. Condition numbers of gaussian random matrices. *SIAM J Matrix Anal A*, 27(3):603–620, 2005.
10. Z. Chen and A. Storjohann. A BLAS based C library for exact linear algebra on integer matrices. In *Proc. of ISSAC'05*, pages 92–99. ACM, 2005.
11. H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1995.
12. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J of Cryptology*, 10(4):233–260, 1997.
13. H. Daudé and B. Vallée. An upper bound on the average number of iterations of the LLL algorithm. *TCS*, 123(1):95–115, 1994.
14. J. von zur Gathen and J. Gerhardt. *Modern Computer Algebra*. Cambridge University Press, 2003.
15. W. B. Johnson and J. Lindenstrauss. Extension of Lipschitz mappings into a Hilbert space. *Comm Contemp Math*, 26:189–206, 1984.
16. H. Koy and C. P. Schnorr. Segment LLL-reduction of lattice bases. In *Proc. of CALC'01*, volume 2146 of *LNCS*, pages 67–80. Springer, 2001.
17. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math Ann*, 261:513–534, 1982.
18. P. Nguyen and D. Stehlé. Floating-point LLL revisited. In *Proc. of Eurocrypt 2005*, volume 3494 of *LNCS*, pages 215–233. Springer, 2005.
19. P. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Proc. of CALC'01*, volume 2146 of *LNCS*, pages 146–180. Springer, 2001.
20. A. M. Odlyzko and H. J. J. te Riele. Disproof of Mertens conjecture. *J reine angew Math*, 357:138–160, 1985.
21. A. Rouault. Asymptotic behavior of random determinants in the laguerre, gram and jacobi ensembles. *Latin American Journal of Probability and Mathematical Statistics (ALEA)*, 3:181–230, 2007.
22. C. P. Schnorr. Progress on LLL and lattice reduction. In *Proc. of the LLL+25 conference*. To appear.
23. D. Stehlé. On the randomness of bits generated by sufficiently smooth functions. In *Proc. of ANTS VII*, volume 4076 of *LNCS*, pages 257–274. Springer, 2006.
24. D. Stehlé, V. Lefèvre, and P. Zimmermann. Searching worst cases of a one-variable function. *IEEE Trans Comp*, 54(3):340–346, 2005.