Symplectic Lattice Reduction and NTRU

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

[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] CNRS/École normale supérieure, DI, 45 rue d'Ulm, 75005 Paris, France
http://www.di.ens.fr/~pnguyen

Abstract. NTRU is a very efficient public-key cryptosystem based on polynomial arithmetic. Its security is related to the hardness of lattice problems in a very special class of lattices. This article is motivated by an interesting peculiar property of NTRU lattices. Namely, we show that NTRU lattices are proportional to the so-called symplectic lattices. This suggests to try to adapt the classical reduction theory to symplectic lattices, from both a mathematical and an algorithmic point of view. As a first step, we show that orthogonalization techniques (Cholesky, Gram-Schmidt, QR factorization, etc.) which are at the heart of all reduction algorithms known, are all compatible with symplecticity, and that they can be significantly sped up for symplectic matrices. Surprisingly, by doing so, we also discover a new integer Gram-Schmidt algorithm, which is faster than the usual algorithm for all matrices. Finally, we study symplectic variants of the celebrated LLL reduction algorithm, and obtain interesting speed ups.

1 Introduction

The NTRU cryptosystem [12] is one of the fastest public-key cryptosystems known, offering both encryption (under the name NTRUEncrypt) and digital signatures (under the name NTRUSign [11]). Besides efficiency, another interesting feature of NTRU compared to traditional public-key cryptosystems based on factoring or discrete logarithm is its potential resistance to quantum computers: no efficient quantum algorithm is known for NP-hard lattice problems. The security and insecurity of NTRU primitives has been a popular research topic in the past 10 years, and NTRU is now being considered by the *IEEE P1363.1* standards [16].

The security of NTRU is based on the hardness of two famous lattice problems, namely the shortest and closest vector problems (see for instance the survey [23]), in a very particular class of lattices called convolution modular lattices by [20]. More precisely, it was noticed by the authors of NTRU and by Coppersmith and Shamir [6] that ideal lattice reduction algorithms could heuristically recover NTRU's secret key from the public key. This does not necessarily imply that NTRU is insecure, since there is a theoretical and experimental gap between existing reduction algorithms (such as LLL [18] or its block improvements by Schnorr [25]) and ideal lattice reduction (which is assumed to be solving NP-hard lattice problems), while NTRU is so far the only lattice-based cryptosystem known that can cope with high dimensions without sacrificing performances. Nor does it mean that the security of NTRU primitives is strictly equivalent to the hardness of lattice problems. In fact, the main attacks on NTRU primitives have bypassed the hard lattice problems: this was notably the case for the decryption failure attacks [15] on NTRUEncrypt, the attacks [7,8] on the ancestor NSS [13] of NTRUSign [11], as well as the recent attack [21] on NTRUSign [11] without perturbation. Almost ten years after the introduction of NTRU [12], no significant weakness on NTRU lattices has been found, despite the very particular shape of NTRU lattice bases: both the public and secret NTRU bases are $2N \times 2N$ matrices formed by four blocks of $N \times N$ circulant matrices. It is this compact representation that makes NTRU much more efficient than other lattice-based or knapsack-based schemes (see the survey [23]). A fundamental open question is whether this particular shape makes NTRU lattices easier to reduce or not.

Our Results. We propose to exploit the structure of NTRU lattices in lattice reduction algorithms. As a starting point, we observe a peculiar property of NTRU lattices: we show that NTRU lattices are proportional to the so-called *symplectic* lattices (see the survey [2]). As their name suggests, symplectic lattices are related to the classical symplectic group [28]: a lattice is said to be symplectic if it has at least one basis whose Gram matrix is symplectic, which can only occur in even dimension. Such lattices are *isodual*: there exists an isometry of the lattice onto its dual. Interestingly, most of the well-known lattices in low even dimension are proportional to symplectic lattices, *e.g.*the roots lattices $\mathbb{A}_2$, $\mathbb{D}_4$ and $\mathbb{E}_8$, the Barnes lattice P_6, the Coxeter-Todd lattice K_{12}, the Barnes-Wall lattice BW_{16} and the Leech lattice Λ_{24} (see the bible of lattices [5]). Besides, there is a one-to-one correspondence between symplectic lattices and principally polarized complex Abelian varieties, among which Jacobians form an interesting case (see [3]). This has motivated the study of symplectic lattices in geometry of numbers.

However, to our knowledge, symplectic lattices have never been studied in reduction theory. The long-term goal of this paper is to explore the novel concept of symplectic lattice reduction in which the classical reduction theory is adapted to symplectic lattices, from both a mathematical and an algorithmic point of view in order to speed up reduction algorithms. As a first step, we show that the Gram-Schmidt orthogonalization process – which is at the heart of all lattice reduction algorithms known – preserves symplecticity, and that is made possible by a slight yet essential change on the classical definition of a symplectic matrix, which is fortunately compatible with the standard theory of the symplectic group. We then exploit this property to speed up its computation for symplectic lattices. In doing so, we actually develop a new and faster method to compute integral Gram-Schmidt, which is applicable to all matrices, and not just symplectic matrices.

The method is based on duality: it is faster than the classical method, because it significantly reduces the number of long-integer divisions. When applied to symplectic matrices, a further speed up is possible thanks to the links between symplecticity and duality: in practice, the method then becomes roughly 30 times faster than the classical GS method, which is roughly the time it would take on a matrix of halved dimension. Finally, we study symplectic versions of the celebrated LLL lattice basis reduction algorithm [18] and obtain a speedup of 6 for NTRU lattices of standard size. We restrict to the so-called integral version of LLL to facilitate comparisons: it might be difficult to compare two floating-point variants with different stability properties. We leave the cases of floating-point variants [22] and improved reduction algorithms [25] to future work, but the present work seems to suggest that reduction algorithms might be optimized to NTRU lattices in such a way that a $2n$-dimensional NTRU lattice would not take more time to reduce than an αn-dimensional lattice for some $\alpha < 2$. This is the case for Gram-Schmidt orthogonalization and LLL.

Related Work. Incidentally, the compatibility of the symplectic group with respect to standard matrix factorizations has recently been studied in [19]: however, because they rely on the classical definition of a symplectic matrix, they fail to obtain compatibility with Gram-Schmidt orthogonalization or the QR decomposition.

Road map. The paper is organized as follows. In Section 2, we provide necessary background on lattice reduction and the symplectic group. In Section 3, we explain the relationship between NTRU lattices and symplecticity. In Section 4, we show that the Gram-Schmidt orthogonalization process central to all lattice reduction algorithms known is fully compatible with symplecticity. In Section 5, we present a new integral Gram-Schmidt algorithm, which leads to significant speed-ups for symplectic matrices. The final section 6 deals with symplectic variants of integral LLL.

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 Joe Silverman, Mike Szydlo and William Whyte for useful conversations.

2 Background

Let $\|.\|$ and $\langle ., . \rangle$ be the Euclidean norm and inner product of $\mathbb{R}^n$. Vectors will be written in bold, and we will use row-representation for matrices. The notations $\mathcal{M}_n(\mathbb{R})$ represents the $n \times n$-dimensional matrices over $\mathbb{R}$, and $GL_n(\mathbb{R})$ the n-dimensional invertible matrices of $\mathcal{M}_n(\mathbb{R})$. 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 matrix norm $|M|$ represents the maximum of the euclidean norms of the rows of M. The notation $\lceil x \rfloor$ denotes a closest integer to x.

2.1 Lattices

We refer to the survey [23] for a bibliography on lattices. In this paper, by the term lattice, we mean a discrete subgroup of $\mathbb{R}^n$. The simplest lattice is $\mathbb{Z}^n$. It turns out that in any lattice L, not just $\mathbb{Z}^n$, there must exist linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d \in L$ such that:

$$L = \left\{ \sum_{i=1}^{d} n_i \mathbf{b}_i \mid n_i \in \mathbb{Z} \right\}.$$

Any such d-tuple of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d$ 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 $GL_d(\mathbb{Z})$. The *dimension* of a lattice L is the dimension d of the linear span of L. Let $[\mathbf{b}_1, \ldots, \mathbf{b}_d]$ be vectors: the lattice is full-rank if $d = n$, which is the usual case. We denote by $G(\mathbf{b}_1, \ldots, \mathbf{b}_d)$ their *Gram matrix*, that is the $d \times d$ symmetric matrix $(\langle \mathbf{b}_i, \mathbf{b}_j \rangle)_{1 \le i,j \le d}$ formed by all the inner products. The *volume* $\mathrm{vol}(L)$ (or *determinant*) of the lattice L is the square root of the determinant of the Gram matrix of any basis of L: here, the Gram matrix is symmetric definite positive. The dual lattice $L^\times$ of a lattice L is:

$$L^\times = \{ \mathbf{v} \in \mathrm{span} L,\ \forall \mathbf{u} \in L,\ \langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z} \}.$$

They have the same dimension and their volumes satisfy $\mathrm{vol}(L) \cdot \mathrm{vol}(L^\times) = 1$. If $B = [\mathbf{b}_1, ..., \mathbf{b}_d]$ is a basis of L, and $\delta_{i,j}$ the Kronecker symbol, then the dual family $B^\times = [\mathbf{b}_1^\times, ..., \mathbf{b}_d^\times]$ with $\mathbf{b}_i^\times \in \mathrm{span}(L)$ satisfying $\langle \mathbf{b}_i^\times, \mathbf{b}_j \rangle = \delta_{i,j}$ is a basis of $L^\times$ called the dual basis of B. The Gram matrices of B and $B^\times$ are inversed, and when L is a full rank lattice, $B^\times = B^{-t}$.

2.2 The Symplectic Group

The symplectic group is one of the classical groups [28], whose name is due to Weyl. Given four matrices $A, B, C, D \in \mathcal{M}_n(\mathbb{R})$ we denote by $Q[A, B, C, D]$ the $(2n) \times (2n)$ matrix with A, B, C, D as its quadrants:

$$Q[A, B, C, D] = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$

Symplectic matrices are matrices preserving a nondegenerate antisymmetric bilinear form. Let σ be an isometry of $\mathbb{R}^n$. Then $\sigma^2 = -1$ if and only if there exists an orthonormal basis of $\mathbb{R}^n$ over which the matrix of σ is $J_{2n} = Q[0, I_n, -I_n, 0]$, where I_n is the $n \times n$ identity matrix. Usually, a matrix $M \in \mathcal{M}_{2n}(\mathbb{R})$ is said to be *symplectic* if and only if

$$M^t J_{2n} M = J_{2n}. \tag{1}$$

where M^t is the transpose of M. This is equivalent to M being invertible and having inverse equal to:

$$M^{-1} = -J_{2n} M^t J_{2n}. \tag{2}$$

The set of such matrices is denoted by $Sp(2n, \mathbb{R})$, which is a subgroup of the special linear group $SL_{2n}(\mathbb{R})$: a symplectic matrix has determinant $+1$. A matrix is symplectic if and only if its transpose is symplectic. The matrix $Q[A, B, C, D]$ is symplectic if and only if $AD^t - BC^t = I_n$ and both the matrices AB^t and CD^t are symmetric. It follows that a triangular matrix $Q[A, 0, C, D]$ may only be symplectic if A and D are diagonal, which is too restrictive to make the symplectic group fully compatible with standard matrix factorizations involving triangular matrices.

To fix this, we consider a variant of the usual symplectic group obtained by equation (1) with another matrix J_{2n}. Fortunately, this is allowed by the theory, as while as J_{2n} is a nonsingular, skew-symmetric matrix. From now on, we thus let $J_{2n} = Q[0, R_n, -R_n, 0]$ where R_n is the reversed identity matrix: the identity where the rows (or the columns) are in reverse order, that is, the (i, j)-th coefficient is the Kronecker symbol $\delta_{i,n+1-j}$. This new matrix J_{2n} still satisfies $J_{2n}^2 = -I_{2n}$, and is therefore compatible with symplecticity. From now on, by a symplectic matrix, we will mean a matrix satisfying equation (1) or (2) with this choice of J_{2n}, and this will be our symplectic group $Sp(2n, \mathbb{R})$. Now, a matrix $Q[A, B, C, D]$ is symplectic if and only if the following conditions hold:

$$BA^s = AB^s,\ DC^s = CD^s, AD^s - BC^s = R_n \tag{3}$$

where $M^s = R_n M^t R_n$ for any $M \in \mathcal{M}_n(\mathbb{R})$, which corresponds to reflecting the entries in the off-diagonal $m_{i,j} \leftrightarrow m_{n+1-j,n+1-i}$. The matrix $R_n M$ reverses the rows of M, while MR_n reverses the columns. In other words, compared to the usual definition of symplectic matrices, we have replaced the transpose operation M^t and I_n by respectively the reflection M^s and R_n. This will be the general rule to switch from the usual symplectic group to our variant. In fact, it can be checked that the reflection $M \mapsto M^s$ is an involution of $Sp(2n, \mathbb{R})$: M^s is symplectic (though R_n is not symplectic), and $(M^s)^s = M$. Now a triangular matrix $Q[A, 0, C, D]$ may be symplectic without requiring A and D to be diagonal. Naturally, M^{-s} will mean the inverse of M^s.

To conclude this subsection, let us give a few examples of symplectic matrices with our own definition of $Sp(2n, \mathbb{R})$, which will be very useful in the rest of the paper:

- Any element of $SL_2(\mathbb{R})$, that is, any 2x2 matrix with determinant 1.
- A diagonal matrix of $\mathcal{M}_{2n}(\mathbb{R})$ with coefficients $d_1, ..., d_{2n}$ is symplectic if and only if $d_i = 1/d_{2n+1-i}$ for all i.
- Any $\begin{bmatrix} A & 0 & B \\ 0 & M & 0 \\ C & 0 & D \end{bmatrix}$ including $\begin{bmatrix} I & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & I \end{bmatrix}$ where $\begin{cases} M \in Sp(2n, \mathbb{R}) \\ Q[A, B, C, D] \in Sp(2m, \mathbb{R}) \end{cases}$.
- $Q[U, 0, 0, U^{-s}]$ for any invertible matrix $U \in GL_n(\mathbb{R})$.
- $Q[I_n, 0, A, I_n]$ for any $A \in \mathcal{M}_n(\mathbb{R})$ such that $A = A^s$, that is, A is *reversed-symmetric.*

The symplecticity can be checked by equations (1), (2) or (3). In particular, these equations prove the following elementary lemma, which gives the structure of symplectic triangular matrices.

Lemma 1. *A lower-triangular $2n$-dimensional matrix L can always be decomposed as follows:*

$$L = \begin{bmatrix} \alpha & 0 & 0 \\ \mathbf{u}^t & M & 0 \\ \beta & \mathbf{v} & \gamma \end{bmatrix} \quad \textit{where} \quad \begin{cases} \alpha, \beta, \gamma \in \mathbb{R} \\ \mathbf{u}, \mathbf{v} \in \mathbb{R}^{2n-2} \\ M \in \mathcal{M}_{2n-2}(\mathbb{R}) \textit{ is triangular} \end{cases} .$$

Then the matrix L is symplectic if and only if M is symplectic (and triangular), $\gamma = \frac{1}{\alpha}$ and $\mathbf{u} = -\alpha \mathbf{v} J_{2n-2} M^t$.

2.3 Symplectic Lattices

A lattice L is said to be *isodual* if there exists an isometry σ of L onto its dual (see the survey [2]). One particular case of isodualities is when $\sigma^2 = -1$, in which case the lattice is called "*symplectic*". There exists an orthogonal basis of span(L) over which the matrix of σ is J_{2n} there is at least a basis of L whose Gram matrix is symplectic.

A symplectic lattice has volume equal to 1. In this paper, we will say that an integer full-rank lattice $L \in \mathbb{Z}^{2n}$ is *q-symplectic* if the lattice $L/\sqrt{q}$ is symplectic where $q \in \mathbb{N}^*$. Its volume is equal to q^n. Our q-symplectic lattices seem to be a particular case of the modular lattices introduced by Quebbemann [24], which are connected to modular forms.

2.4 Orthogonalization

Cholesky. Let $G \in \mathcal{M}_n(\mathbb{R})$ be symmetric definite positive. There exists a unique lower triangular matrix $L \in \mathcal{M}_n(\mathbb{R})$ with strictly positive diagonal such that $G = LL^t$. The matrix L is the *Cholesky factorization* of G, and its Gram matrix is G.

The $\mu D \mu^t$ factorization. This factorization is the analogue of the so-called "LDL decomposition" in [9, Chapter 4.1]. Let $G \in \mathcal{M}_n(\mathbb{R})$ be symmetric definite. There exists a unique lower triangular matrix $\mu \in \mathcal{M}_n(\mathbb{R})$ with unit diagonal and a unique diagonal matrix $D \in \mathcal{M}_n(\mathbb{R})$ such that $G = \mu D \mu^t$. The couple (μ, D) is the *$\mu D \mu^t$ factorization* of G. When G is positive definite, then D is positive diagonal, and the relation between the $\mu D \mu^t$ and Cholesky factorizations of G is $L = \mu \sqrt{D}$.

QR or LQ. Let $M \in GL_n(\mathbb{R})$. There exists a unique pair $(Q, R) \in \mathcal{M}_n(\mathbb{R})^2$ such that $M = QR$, where Q is unitary and R is upper triangular with strictly positive diagonal. This is the standard QR factorization. Since we deal with row matrices, we prefer to have a lower triangular matrix, which can easily be achieved by transposition. It follows that there also exists a unique pair $(L, Q) \in \mathcal{M}_n(\mathbb{R})^2$ such that $M = LQ$, where Q is unitary and L is lower triangular with strictly positive diagonal. Note that L is the Cholesky factorization of the Gram matrix MM^t of M.

Gram-Schmidt. Let $[\mathbf{b}_1, \ldots, \mathbf{b}_d]$ be linearly independent vectors represented by the $d \times n$ matrix B. Their *Gram-Schmidt orthogonalization* (GSO) is the orthogonal family $[\mathbf{b}_1^*, \ldots, \mathbf{b}_d^*]$ defined recursively as follows: $\mathbf{b}_1^* = \mathbf{b}_1$ and $\mathbf{b}_i^*$ is the component of $\mathbf{b}_i$ orthogonal to the subspace spanned by $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. We have $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$ where $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2$ for all $i < j$. We let $\mu \in \mathcal{M}_d(\mathbb{R})$ be the lower triangular matrix whose coefficients are $\mu_{i,j}$ above the diagonal, and 1 on the diagonal. If B^* is the $d \times n$ row matrix representing $[\mathbf{b}_1^*, \ldots, \mathbf{b}_d^*]$, then $B = \mu B^*$. If we let G be the Gram matrix BB^t of B, then μ is exactly the matrix from the $\mu D \mu^t$ decomposition of G, and its Cholesky factorization $L = (\ell_{i,j})$ is related to the GSO by: $\ell_{i,j} = \mu_{i,j} \|\mathbf{b}_j^*\|$ for $i < j$. The matrices L and B have the same Gram matrix, so the GSO can be viewed as a trigonalization of the lattice Λ spanned by B. Note that $\mathrm{vol}(\Lambda) = \prod_{i=1}^d \|\mathbf{b}_i^*\|$.

Integral Gram-Schmidt. In practice, we are interested in the case where the $\mathbf{b}_i$'s are in $\mathbb{Z}^n$. Then the $\mathbf{b}_i^*$'s and the $\mu_{i,j}$'s are in general rational. To avoid rational arithmetic, it is customary to use the following integral quantities (as in [27] and in the complexity analysis of [18]): for all $1 \leq i \leq d$, let: $\lambda_{i,i} = \prod_{j=1}^i \|\mathbf{b}_j^*\|^2 = \mathrm{vol}(\mathbf{b}_1, \ldots, \mathbf{b}_i)^2 \in \mathbb{Z}$. Then let $\lambda_{i,j} = \mu_{i,j} \lambda_{j,j}$ for all $j < i$, so that $\mu_{i,j} = \frac{\lambda_{i,j}}{\lambda_{j,j}}$. It is known that $\lambda_{i,j} \in \mathbb{Z}$. When using the GSO for lattice reduction, one does not need to compute the $\mathbf{b}_i^*$'s themselves: one only needs to compute the $\mu_{i,j}$'s and the $\|\mathbf{b}_i^*\|^2$. Since $\|\mathbf{b}_i^*\|^2 = \lambda_{i,i} / \lambda_{i-1,i-1}$ (if we let $\lambda_{0,0} = 1$), it follows that it suffices to compute the integral matrix $\lambda = (\lambda_{i,j})_{1 \leq i,j \leq d}$ for lattice reduction purposes. This is done by Algorithm 1, whose running time is $O(nd^4 \log^2 |B|)$ where $|B|$ is an upper bound of the $\|\mathbf{b}_i\|$'s.

Algorithm 1 Standard GS

```
Input: A set of $d$ linearly independent vectors $[\mathbf{b_1}, ..., \mathbf{b_d}]$ of $\mathbb{Z}^n$
Output: The $\lambda$ matrix of the GSO of $[\mathbf{b_1}, ..., \mathbf{b_d}]$.
for $i = 1$ to $d$ do
    $\lambda_{i,1} \leftarrow \langle \mathbf{b}_i, \mathbf{b}_1 \rangle$
    for $j = 2$ to $i$ do
        $S = \lambda_{i,1} \lambda_{j,1}$
        for $k = 2$ to $j - 1$ do
            $S \leftarrow (\lambda_{k,k} S + \lambda_{j,k} \lambda_{i,k}) / \lambda_{k-1,k-1}$
        end for
        $\lambda_{i,j} \leftarrow \langle \mathbf{b}_i, \mathbf{b}_j \rangle \lambda_{j-1,j-1} - S$
    end for
end for
```

2.5 LLL reduction

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

LLL reduction. A basis $[\mathbf{b}_1, \ldots, \mathbf{b}_d]$ is LLL-reduced [18] 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 satisfies the $(d-1)$ Lovász conditions $(\delta - \mu_{i,i-1}^2) \left\| \mathbf{b}_{i-1}^* \right\|^2 \leq \| \mathbf{b}_i^* \|^2$, which means that the GSO vectors never drop too much. Such bases have several useful properties (see [4,18]), notably the following one: the first basis vector is relatively short, namely

$$\|\mathbf{b}_1\| \leq \beta^{(d-1)/4} \mathrm{vol}(L)^{1/d} \text{ , where } \ \beta = 1/(\delta - \eta^2).$$

LLL-reduction usually refers to the factor $(3/4, 1/2)$ because this was the choice considered in the original paper by Lenstra, Lenstra and Lovász [18]. But the closer δ and η are respectively to 1 and 1/2, the more reduced the basis is. The classical LLL algorithm obtains in polynomial time a basis reduced with factor $(\delta, 1/2)$ where δ can be arbitrarily close to 1. Reduction with a factor (1,1/2) is closely related to a reduction notion introduced by Hermite [10].

The LLL algorithm. The basic LLL algorithm [18] computes an LLL-reduced basis in an iterative fashion: there is an index κ such that at any stage of the algorithm, the truncated basis $[\mathbf{b}_1, \ldots, \mathbf{b}_{\kappa-1}]$ is LLL-reduced. At each loop iteration, κ is either incremented or decremented: the loop stops when κ eventually reaches the value $d+1$, in which case the entire basis $[\mathbf{b}_1, \ldots, \mathbf{b}_d]$ is already LLL-reduced.

LLL uses two kinds of operations: swaps of consecutive vectors and Babai's nearest plane algorithm [1], which performs at most d translations of the form $\mathbf{b}_\kappa \leftarrow \mathbf{b}_\kappa - m\mathbf{b}_i$, where m is some integer and $i < \kappa$. Swaps are used to achieve Lovász conditions, while Babai's algorithm is used to size-reduce vectors.

If L is a full-rank lattice of dimension n and $|B|$ is an upper bound on the $\|\mathbf{b}_i\|$'s, then the complexity of the LLL algorithm (using integral Gram-Schmidt) without fast integer arithmetic is $O(n^6 \log^3 |B|)$. The recent L^2 algorithm [22] (based on floating-point Gram-Schmidt) 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.

3 NTRU Lattices

The NTRU [12] cryptosystem has many variants. To simplify our exposition, we focus on the usual version, but our results apply to all known variants of NTRU.

Let n be a prime number about several hundreds ($e.g.$251), and q be a small power of two ($e.g.$128 or 256). Let $\mathcal{R}$ be the ring $\mathbb{Z}[X]/(X^n - 1)$ whose multiplication is denoted by $*$. The NTRU secret key is a pair of polynomials $(f, g) \in \mathcal{R}^2$ with tiny coefficients compared to q, say 0 and 1. The polynomial f is chosen to be invertible modulo q, so that the polynomial $h = g/f \bmod q$ is well-defined in $\mathcal{R}$. The NTRU public key is the polynomial $h \in \mathcal{R}$ with coefficients modulo q. Its fundamental property is: $f * h \equiv g \bmod q$ in $\mathcal{R}$.

In order to fit multiplicative properties of polynomials of $\mathcal{R}$, we use circulant matrices. The application φ that maps a polynomial in $\mathcal{R}$ to its circulant matrix in $\mathcal{M}_n(\mathbb{Z})$ is defined by:

$$\varphi(\sum_{i=0}^{n-1} h_i X^i) = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_{n-1} & h_0 & \cdots & h_{n-2} \\ \vdots & \ddots & \ddots & \vdots \\ h_1 & \cdots & h_{n-1} & h_0 \end{bmatrix}$$

1. This application is a ring morphism.
2. Circulant matrices are reversed-symmetric: $\varphi(a)^s = \varphi(a)$ for any $a \in \mathcal{R}$.

There is a natural lattice $\mathit{\Lambda}$ in $\mathbb{Z}^{2n}$ corresponding to the set of pairs of polynomials $(u, v) \in \mathcal{R}^2$ such that $v * h \equiv u \bmod q$ (see [6,23]). This lattice can be defined by the following basis, which is public since it can be derived from the public key:

$$B = Q\left[\varphi(q), \varphi(0), \varphi(h), \varphi(1)\right].$$

This basis is in fact the Hermite normal form of $\mathit{\Lambda}$. It follows that the dimension of $\mathit{\Lambda}$ is $2n$ and its volume is q^n. Notice that $B/\sqrt{q}$ is symplectic by equation (1), and therefore the public basis B and the NTRU lattice $\mathit{\Lambda}$ are q-symplectic.

Because of the fundamental property of the public key h, there is a special lattice vector in $\mathit{\Lambda}$ corresponding to (g, f), which is heuristically the shortest lattice vector. All the vectors corresponding to the rotations $(g * X^k, f * X^k)$ also belong to $\mathit{\Lambda}$. In fact, in NTRUSign [11], the pair (f, g) is selected in such a way that there exists another pair of polynomials $(F, G) \in \mathcal{R}^2$ such that $f * G - g * F = q$ in $\mathcal{R}$. It follows that the following matrix is a secret basis of $\mathit{\Lambda}$:

$$C = Q\left[\varphi(g), \varphi(f), \varphi(G), \varphi(F)\right].$$

This is the basis used to sign messages in NTRUSign.

Hence, if a, b, c, d are polynomials in $\mathcal{R}$, the matrix $M = Q[\varphi(a), \varphi(b), \varphi(c), \varphi(d)]$ satisfies:

$$-MJ_{2n}M^tJ_{2n} = Q[\varphi(a * d - b * c), 0, 0, \varphi(a * d - b * c)].$$

In particular, the secret basis satisfies: $-CJ_{2n}C^tJ_{2n} = qI_{2n}$, which proves that C is a q-symplectic matrix like B, only with smaller coefficients. Hence, the unimodular transformation that transforms the public basis B into the secret basis C is symplectic. One may wonder if it is possible to design special (possibly faster) lattice reduction algorithms for NTRU lattices, which would restrict their elementary row transformations to the symplectic subgroup of $GL_{2n}(\mathbb{Z})$. This motivates the study of symplectic lattice reduction.

4 Symplectic Orthogonalization

All lattice reduction algorithms known are based on Gram-Schmidt orthogonalization, which we recalled in Section 2. In this section, we show that Cholesky factorization, LQ decomposition and Gram-Schmidt orthogonalization are compatible with symplecticity. The key result of this section is the following symplectic analogue of the so-called LDL^t factorization of a symmetric matrix:

Theorem 1 (Symplectic $\mu D \mu^t$). *Let G be a symmetric matrix in $Sp(2n, \mathbb{R})$. There exists a lower-triangular matrix $\mu \in Sp(2n, \mathbb{R})$ whose diagonal is 1, and a diagonal matrix $D \in Sp(2n, \mathbb{R})$ such that, $G = \mu D \mu^t$. And the pair (μ, D) is unique.*

Before proving this theorem, let us give three important consequences on Cholesky factorization, LQ factorization and Gram-Schmidt orthogonalization:

Theorem 2 (Symplectic Cholesky). *If $G \in S_p(2n, \mathbb{R})$ is a symmetric positive definite matrix, then its Cholesky factorization is symplectic.*

Proof. Apply Theorem 1 to G, then μ is lower-triangular symplectic with only 1 on the diagonal. Since G is positive definite, the diagonal matrix $D = \mu^{-1} G \mu^{-t}$ is positive definite. But D is also symplectic, so its coefficients satisfy $d_{i,i} = 1/d_{2n+1-i,2n+1-i}$ (see the end of Section 2.2). For these reasons, the square root C of D (with $c_{i,i} = \sqrt{d_{i,i}}$) is also symplectic. It is clear that $L = \mu C$ is symplectic and satisfies $G = LL^t$. Since the Cholesky factorisation of G is unique, it must be L and it is therefore symplectic. □

Theorem 3 (Symplectic LQ). *If B is symplectic, then its LQ decomposition is such that both L and Q are symplectic.*

Proof. L is the Cholesky factorization of the matrix BB^t, which is symplectic, so the previous theorem shows that L is symplectic. Then $Q = L^{-1} B$ is also symplectic, because $Sp(2n, \mathbb{R})$ is a group. □

Theorem 4 (Symplectic Gram-Schmidt). *If B is symplectic, then the μ matrix of its Gram-Schmidt orthogonalisation is also symplectic.*

Proof. Apply Theorem 1 to $G = BB^t$, then μB represents an orthogonal basis, because its Gram matrix is diagonal. □

Thus, the isometry σ represented by J_{2n} that sends the symplectic basis onto its dual basis is also an isometry between each part of the GSO of the symplectic basis and its dual basis:

Corollary 1. *Let $[\mathbf{b}_1, ..., \mathbf{b}_{2n}]$ be a symplectic basis of a $2n$-dimensional lattice, then the GSO satisfy for all $i \leq n$, $\mathbf{b}^*_{2n+1-i} = \frac{1}{||\mathbf{b}^*_i||^2} \mathbf{b}^*_i J$ and $\mathbf{b}^*_i = \frac{1}{||\mathbf{b}^*_{2n+1-i}||^2} \mathbf{b}^*_{2n+1-i} J$*

Proof. Consider the LQ factorization of $[\mathbf{b}_1, ..., \mathbf{b}_{2n}]$. The unitary matrix Q is symplectic, therefore equation (2) implies that $Q = -JQJ$. Hence, a unitary symplectic matrix always has the form:

$$\begin{pmatrix} C & D \\ -R_n D R_n & R_n C R_n \end{pmatrix} = \begin{pmatrix} A \\ R_n A J_{2n} \end{pmatrix}.$$

This proves that the directions of the $\mathbf{b}_i^*$ in this corollary are correct. Their norm are the diagonal coefficients of L, and this matrix is lower-triangular symplectic, so Lemma 1 implies that $||\mathbf{b}_{2n+1-i}^*|| = 1/||\mathbf{b}_i^*||$. □

We now prove Theorem 1 by induction over n. There is nothing to prove for $n = 0$. Thus, assume that the result holds for $n - 1$ and let $G = (g_{i,j})$ be a symmetric matrix in $Sp(2n, \mathbb{R})$. The main idea is to reduce the first column G with a symplectic transformation, then verify that it automatically reduces the last row, and finally use the induction hypothesis to reduce the remaining $2n-2$ dimensional block in the center. The symplectic transformation has the form:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ (\alpha_2, ..., \alpha_{2n-1})^t & I_{2n-2} & 0 \\ \alpha_{2n} & (\alpha_2, ..., \alpha_{2n-1}) J_{2n-2} & 1 \end{bmatrix} \text{ where } \alpha_2, ..., \alpha_{2n-1} \in \mathbb{R}.$$

This is a symplectic matrix because of Lemma 1. Apply the transformation with $\alpha_i = -\frac{g_{i,1}}{g_{1,1}}$. Then PGP^t has the following shape:

$$PGP^t = \begin{bmatrix} g_{1,1} & 0 & \gamma \\ 0 & S & \mathbf{u}^T \\ \gamma & \mathbf{u} & \beta \end{bmatrix},$$

where S is a $(2n-2) \times (2n-2)$ symmetric matrix, $\mathbf{u}$ is a $(2n-2)$ dimensional row vector, and $\beta \in \mathbb{R}$ and $\gamma = 0$. The coefficient γ in the bottom left corner of PGP^t is equal to zero, because α_{2n} satisfies $\alpha_{2n} g_{1,1} + \alpha_{2n-1} g_{2,1} + ... + \alpha_{n+1} g_{n,1} - \alpha_n g_{n+1,1} - ... - \alpha_2 g_{2n-1,1} + g_{2n,1} = 0$.

Since PGP^t is symplectic, the image by J_{2n} of the first row $\mathbf{r}_1$ of PGP^t has the form $\mathbf{e}_{2n} = (0, ..., 0, g_{1,1})$ and its j^{th} row $\mathbf{r}_j$ satisfies $\langle \mathbf{e}_{2n}, \mathbf{r}_j \rangle = \delta_{2n,j}$ for all $j \geq 2$ (where δ is the Kronecker symbol): in other words, $\mathbf{u} = 0$ and $\beta = 1/g_{11}$.

$$PGP^t = \begin{bmatrix} g_{1,1} & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & \frac{1}{g_{1,1}} \end{bmatrix}.$$

As a result, S is symmetric positive definite and symplectic. The induction hypothesis implies the existence of a pair (μ_S, D_S) such that $S = \mu_S D_s \mu_S^t$, and we can extend μ_S to a lower-triangular matrix $U \in Sp(2n, \mathbb{R})$ using the third property at the end of Section 2.2:

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mu_S & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Hence, the product $\mu = UP^{-1}$ is a lower-triangular symplectic matrix whose diagonal is 1, and $D = \mu^{-1} G \mu^{-t} \in Sp(2n, \mathbb{R})$ is diagonal. This concludes the proof of Theorem 1 by induction.

5 Speeding-up Gram-Schmidt by Duality

The standard integral Gram Schmidt algorithm we recalled in Section 2 is based on a formula which computes $\mu_{i,j}$ from the $\mu_{i,k}$'s ($k < j$) on the same row. This leads to many integer divisions for each coefficient as in the innerloop rows 5-7 of Algorithm 1.

5.1 The general case

We now show that most of these divisions can be avoided. Consider a basis $B = [\mathbf{b}_1, ..., \mathbf{b}_d]$ and its Gram matrix $G = BB^t$. We know that if μ is the Gram-Schmidt orthogonalization of B, then $G = \mu D \mu^t$ where D is a positive diagonal matrix. If we rewrite the previous equation as $\mu = G\mu^{-t}D^{-t}$, it appears that for any integer $k < d$, if we know the $k \times k$ topleft triangle of μ and D, we can compute the $k \times k$ topleft triangle of $\mu^{-t}D^{-t}$ and the first full k columns of μ. The matrix μ^{-t} is just a rotation of μ^{-s}, which is the Gram-Schmidt orthogonalization of the dual basis of $(\mathbf{b}_d, ..., \mathbf{b}_1)$. At the end, we get not only the GSO of B, but also the one of its reverse dual basis: this method which we call "Dual Gram-Schmidt", is surprisingly faster than the classical one despite computing more information.

Theorem 5. *Let $G \in \mathcal{M}_n(\mathbb{Z})$ be a symmetric (positive) definite matrix, and μ the $\mu D \mu^t$ factorization of G. As in Section 2, we define $\lambda_{0,0} = 1$ and $\lambda_{k,k} = \det G_k$ where G_k is the $k \times k$ topleft block of G. Let $\lambda = \mu \cdot \operatorname{diag}(\lambda_{1,1}, ..., \lambda_{n,n})$ and $U = \mu^{-t} \cdot \operatorname{diag}(\lambda_{0,0}, ..., \lambda_{n-1,n-1})$. Then the following three relations hold:*

$$\lambda \in \mathcal{M}_n(\mathbb{Z}), \tag{4}$$

$$U \in \mathcal{M}_n(\mathbb{Z}), \tag{5}$$

$$\lambda = GU. \tag{6}$$

Proof. From $G = \mu D \mu^t$, we know that $\mu^{-1}G$ is uppertriangular: for all i and t with $i > t$, then $\sum_{j=1}^{i} \mu_{i,j}^{-1} g_{j,t} = 0$. If we call $G'$the $(i-1) \times (i-1)$ topleft block of G, $\mathbf{v} = (\mu_{i,1}^{-1}, ..., \mu_{i,i-1}^{-1})$ and $\mathbf{g'} = (g_{i,1}, ..., g_{i,i-1})$, then the previous equation is equivalent to $\mathbf{g'} = -\mathbf{v}G'$. By Cramer's rule, we deduce the following relation for all $j < i$, which proves relation (5):

$$u_{j,i} = \det G' \cdot \mathbf{v}_j = (-1)^{i-j} \det \begin{pmatrix} g_{1,1} & \cdots & g_{1,j-1} & g_{1,j+1} & \cdots & g_{1,i} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_{i-1,1} & \cdots & g_{i-1,j-1} & g_{i-1,j+1} & \cdots & g_{i-1,i} \end{pmatrix}.$$

The relation (6) is obtained by multiplying $\mu = G\mu^{-t}D^{-t}$ by $\operatorname{diag}(\lambda_{1,1}, ..., \lambda_{n,n})$. It also implies that $\lambda_{p,i} = \sum_{k=1}^{i} g_{p,k}u_{k,i}$ is the last row development of an integer determinant:

$$\lambda_{p,i} = \det \begin{pmatrix} g_{1,1} & \cdots & g_{1,i} \\ \vdots & \ddots & \vdots \\ g_{i-1,1} & \cdots & g_{i-1,i} \\ g_{p,1} & \cdots & g_{p,i} \end{pmatrix},$$

which concludes the proof. □

Note that these determinants prove that $\lambda_{p,i} \leq |G|^i$ and $U_{j,i} \leq |G|^{i-1}$.

We derive from $\mu^{-t}\mu^t = \mathrm{Id}$ a column formula to compute the matrix U defined in the previous theorem:

$$(\mu^{-t})_{i,j} = -\mu^t_{i,j} - \sum_{k=j+1}^{i-1} \mu^t_{i,k}(\mu^{-t})_{k,j}. \tag{7}$$

From the definition of U, we know that $(\mu^{-t})_{i,j} = \frac{U_{i,j}}{\lambda_{j-1,j-1}}$. Replacing $\mu_{i,j}$ by $\frac{\lambda_{i,j}}{\lambda_{j,j}}$, we may rewrite the formula as: $-U_{i,j} = \left(\sum_{k=j+1}^{i-1} \lambda_{k,i}U_{k,j}\right)/\lambda_{i,i}$. Hence if we know the $i \times (i-1)$ top-left triangle of λ, we can compute the i^{th} column of U using this formula (from the diagonal to the top), and then the i^{th} column of λ using relation (6) of Theorem 5. It is not necessary to keep the i^{th} column of U after that.

We deduce Algorithm 2 to compute the GSO of a basis B. The correctness of this algorithm is a consequence of the results described in this section. If we look at the number of operations requested, there are $i^2/2$ multiplications and one division in the innerloop lines 4-6, and $i(n+1-i)$ small multiplications between the input Gram matrix and U in the innerloop lines 7-9. This gives a total of $n^3/6$ large multiplications and $n^2/2$ divisions. In the Standard GS algorithm, there was as many multiplications, but $\Theta(n^3)$ divisions.

Algorithm 2 Dual Gram Schmidt

Input: a basis $B = (\mathbf{b}_1, ..., \mathbf{b}_d)$ or its Gram matrix G
Output: the GSO decomposition λ of B

```
 1: for $i = 1$ to $d$ do
 2:     $U_i \leftarrow \lambda_{i-1,i-1}$ (for all $k$, $U_k$ represents $U_{k,i}$)
 3:     $U_{i-1} \leftarrow -\lambda_{i,i-1}$
 4:     for  $j = i - 2$ downto 1 do
 5:         compute $U_j = -(\sum_{k=j+1}^{i} \lambda_{k,j}U_k)/\lambda_{j,j}$
 6:     end for
 7:     for  $j = i$ to $n$ do
 8:         compute $\lambda_{j,i} = \sum_{k=1}^{i} \langle \mathbf{b}_j, \mathbf{b_k} \rangle U_k$
 9:     end for
10: end for
```

5.2 The symplectic case

We derive an algorithm specialized to q-symplectic bases to compute the λ matrix of the GSO, and we show why it is faster than the Dual Gram-Schmidt procedure applied to symplectic bases (see Algorithm 1 of Section 2). Let B be a q-symplectic basis. We know that L in the LQ decomposition of B is q-symplectic and the μ matrix corresponding to the GSO of B is symplectic. We will also use the integer dual matrix U we introduced in Theorem 5. Let us denote the quadrants of μ, and λ by:

$$\mu = \begin{pmatrix} \mu_a & 0 \\ \mu_\gamma & \mu_\delta \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_a & 0 \\ \lambda_\gamma & \lambda_\delta \end{pmatrix} \qquad \text{and} \qquad U = \begin{pmatrix} U_\alpha & 0 \\ U_\gamma & U_\delta \end{pmatrix}.$$

Because of Theorem 4, we know that $\mu_\delta = \mu_\alpha^{-s}$. Together with Corollary 1, we have $U_\alpha = R_n \lambda_\delta R_n$ for symplectic matrices and the following relation for q-symplectic matrices: $U_\alpha \cdot \text{diag}(q^2, q^4, ..., q^{2n}) = R_n \lambda_\delta R_n$. For this reason, it is only necessary to run DualGS up to the middle of the matrix, and fill the columns of λ_δ using those of U_α. Given as input a symplectic basis $B = (\mathbf{b}_1, ..., \mathbf{b}_{2n})$, Algorithm 3 computes the λ matrix of the GSO of B in time $O(n^5 \log^2 B)$ using standard arithmetic.

Algorithm 3 Symplectic Gram-Schmidt

```
Input: A $q$-symplectic basis $B = [\mathbf{b_1}, ..., \mathbf{b_{2n}}]$
Output: The $\lambda$ matrix of the GSO of $B$.
precompute: $q^{2i}$ for $i = 1$ to $n$
for $i = 1$ to $n$ do
    $U_i \leftarrow \lambda_{i-1,i-1}$ (for all $k$, $U_k$ represents $U_{k,i}$)
    $U_{i-1} \leftarrow -\lambda_{i,i-1}$
    for $j = i - 2$ downto 1 do
        $U_j \leftarrow -(\sum_{k=j+1}^{i} \lambda_{k,j} U_k)/\lambda_{j,j}$
    end for
    for $j = 1$ to $i$ do
        $\lambda_{2n+1-j,2n+1-i} \leftarrow q^{2(n+1-i)} \cdot U_j$
    end for
    for $j = i$ to $2n$ do
        compute $\lambda_{j,i} = \sum_{k=1}^{i} G_{j,k} U_k$
    end for
end for
```

5.3 Experiments

We performed tests on randomly generated bases, whose coefficients are uniformly distributed 128-bit integers (see Table 1). On these random matrices, the speedup is rather moderate, but it will be much more significant when considering symplectic matrices.

Table 1. Timing of Gram-Schmidt algorithms on random matrices

n	StandardGS in seconds	DualGS in seconds	**DualGS speedup**
100	37.3	25.1	1.49
200	881	579	1.52
300	5613	3644	1.54

Table 2. Timing of Gram-Schmidt algorithms on NTRUSign bases

$2n$	Standard GS	DualGS	SympGS	**speedup DualGS**	**speedup SympGS**
502	179	122	8.7	1.46	20.5
802	1822	1254	67	1.45	27.1
1214	12103	8515	390	1.42	31.0

We also performed tests on secret bases of NTRUSign as described in Section 3 (see Table 2). Roughly speaking, Algorithm 2 is at least 3 times as fast as Standard GS, and the specialized algorithm is from 10 to 30 times as fast as Standard GS. We give in function of the dimension of the input matrix, the running time in seconds to compute the GSO for each algorithm. Note that the speed-up of 31 in Symplectic GS seems to indicate that the cost of computing the GSO of a $2n$-dimensional symplectic basis is roughly the one of computing the GSO of a standard n-dimensional standard matrix.

6 Symplectic LLL

When applied to a symplectic basis, the standard LLL algorithm will likely not preserve symplecticity, because its elementary operations are not necessarily symplectic. In this section, we show how one can slightly modify the notion of size-reduction used in LLL to make it compatible with symplecticity, and we then deduce a symplectic version of LLL. We do not know if every symplectic lattice contains an LLL-reduced basis which is also symplectic. But we show how to obtain efficiently a symplectic basis which is *effectively LLL-reduced* (as introduced in [14]). Such bases satisfy the most important properties of LLL-reduced bases, those which are sufficient to guarantee the shortness of the first basis vector.

6.1 Symplectic size-reduction

The first condition in LLL-reduction is size-reduction, which we recalled in Section 2. Unfortunately, size-reduction transformations are not necessarily symplectic. However, we show that it is still possible to force half of the coefficients of μ to be very small using symplectic transformations, and at the same time, bound the size of the remaining coefficients.

We say that a matrix $B \in \mathcal{M}_{2n}(\mathbb{R})$ is *semi-size reduced* if its GSO satisfies: for all $j \leq n$, for all $i \in [j+1, 2n+1-j]$, $|\mu_{i,j}| \leq \frac{1}{2}$.

Theorem 6. *If $B \in Sp(2n, \mathbb{R})$ is semi-size reduced, then its GSO μ is bounded by $||\mu||_\infty \leq n \cdot (\frac{3}{2})^n$.*

Proof. For the block μ_δ , Equation (7) gives for $i \geq n+1$ and $j \geq n+1$, $|\mu_{i,j}| \leq \frac{1}{2} + \frac{1}{2}\sum_{k=j+1}^{i-1} |\mu_{i,k}|$, which is bounded by a geometric sequence of ratio $\frac{3}{2}$. Hence, the bottom diagonal block is bounded by $|\mu_{i,j}| \leq \frac{1}{2}(\frac{3}{2})^{i-j-1}$, and this bound can be reached for $\mu_{i\in[1,n], j\in[1,i-1]} = -1/2$.

For the bound on block μ_γ , apply Equation (1) to μ^s in order to get a column formula. This gives for $i \geq n+1$ and $j \geq 2n-i$:

$$\mu^s_{n+1-j,n+1-i} = \mu^s_{i,j} + \sum_{k=j+1}^{n} \mu^s_{i,k}\mu^s_{2n+1-j,2n+1-k} - \sum_{k=n+1}^{i-1} \mu^s_{i,k}\mu^s_{2n+1-j,2n+1-k}.$$

After reindexing the matrix and applying the triangular inequality to this sum, we obtain

$$|\mu_{i,j}| \leq \frac{1}{2} + \frac{2n+1-2j}{4} + \frac{1}{2}\sum_{k=2n+2-j}^{i-1} |\mu_{k,j}|.$$

It is still bounded by a geometric sequence of ratio $\frac{3}{2}$, but the initial term $\mu_{i+1,i}$ is less than $\frac{2n+3-2j}{4} \leq n$. Thus $|\mu_{i,j}| \leq n \cdot (\frac{3}{2})^{i-2n+j-2}$. □

6.2 Lovász conditions and effective reduction

A basis satisfying all Lovász conditions and $|\mu_{i,i-1}| \leq 1/2$ is referred to as *effectively LLL-reduced* in [14]. Such bases have the same provable bounds on the size of $\mathbf{b}_1$ (which is typically the most interesting vector in a basis) as LLL-reduced bases. Besides, it is easy to derive an LLL-reduced basis from an effectively LLL-reduced basis, using just size reductions (no swaps). In general the reason for weakly reducing the other $\mu_{i,j}$ for $1 \leq j < i-1$ is to prevent coefficient explosion in the explicit $\mathbf{b}_i$, but there are many other strategies for this that don't require as strict a condition as $|\mu_{i,j}| \leq 1/2$, $1 \leq j < i-1$ (see [17,26]). It is not difficult to see that this notion of *"effective LLL-reduction"* can be reached by symplectic transformations.

Lemma 2. *A symplectic $2n$-dimensional basis B is* effectively-reduced *if and only if its first $n+1$ vectors are* effectively LLL-reduced.

Proof. Let μ be the GSO matrix of B, for $i \leq n$, since μ is symplectic, we know that $\mu_{2n+2+i,2n+1-i} = \mu_{i,i-1}$. Using Corollary 1, the Lovasz condition for the i^{th} index is equivalent to $\delta \frac{1}{||\mathbf{b^*_{2n+2-i}}||^2} \leq \frac{1}{||\mathbf{b^*_{2n+1-i}}||^2} - \mu_{2n+2+i,2n+1-i} \frac{1}{||\mathbf{b^*_{2n+2-i}}||^2}$, which is precisely the Lovasz condition for the $2n+2-i^{\text{th}}$ index. □

This means that for all $i \leq n$, every operation made on the rows i and $i-1$ that reduces B can be blindly applied on the rows $2n-i+2$ and $2n-i+1$ without knowing the GSO of the second block. A symplectic basis is said to be *symplectic-LLL reduced* if it is both *effectively LLL-reduced* and *semi-size-reduced.*

6.3 A Symplectic-LLL algorithm

It is easy to find polynomial algorithms for symplectic-LLL reduction, but the difficulty is to make them run faster than LLL in practice. Here, we present an algorithm which reaches symplectic-LLL reduction with an experimental running-time 6 times smaller than LLL on NTRU public bases of standard size (but the speed up may be larger for higher-dimensional lattices).

Symplectic LLL is an iterative algorithm that reduces a symplectic lattice L from its center. It takes as input the integer GSO λ of a symplectic lattice and outputs the GSO of a symplectic-LLL reduced basis and the unimodular transformation that achieves the reduction. More precisely, it only keeps one half of the GSO of symplectic matrices, since the other half can be easily deduced with (1) or Lemma 1. Here, we chose to keep the left triangle $\lambda' = \lambda_{i,j}, 1 \leq j \leq n, j \leq i \leq 2n+1-j$. During the algorithm, every elementary operation (swap or a linear combination of rows) is deduced from λ', and λ' is updated incrementally like in the standard integer LLL (see [18,4]). As a result, symplecticLLL can generate the complete sequence of elementary operations of the reduction without knowing the basis. Unfortunately, having only the GSO of the LLL reduced basis is not sufficient to compute its coefficients, so every operation that occur in symplectic LLL algorithm is in fact performed on a third part matrix U. If U is initially equal to the input basis (resp. the identity matrix), then it is transformed into the LLL-reduced basis (resp. the unitary transformation).

In this paragraph, we explain the principles of SymplecticLLL on the projected lattice vectors, but in practice, all operations are done on the GSO λ' (see Algorithm 4 for details). Let $C_k = [\pi_{n+1-k}(\mathbf{b}_{n+1-k}), ..., \pi_{n+1-k}(\mathbf{b}_{n+k-1})]$ where $1 \leq k \leq n$. The $2k$-dimensional lattice $L(C_k)$ is symplectic, and its GSO matrix μ is the $2k \times 2k$ block located in the center of the GSO of the basis. When the algorithm begins, the counter k is set to 1. At the start of each loop iteration, C_{k-1} is already symplectic-LLL-reduced (there is no condition if $k = 1$). If $k = 1$ then the projected lattice C_1 is Lagrange-reduced and the counter k is set to 2. If the determinant of the transformation computed by Lagrange reduction is -1, we negate $\mathbf{b}_{n+1}$ to preserve the symplecticity of the basis. In the general case, C_k is semi-size-reduced, which means that $\lambda_{i,n+1-k}$ is made lower than $\frac{1}{2}\lambda_{n+1-k,n+1-k}$ with symplectic combinations of rows for $i = n+2-k$ to $n+k$. If the pair $(n+1-k, n+2-k)$ does not satisfy Lovász condition (by

Algorithm 4 symplectic LLL

Input: A GSO matrix λ of a q-symplectic basis (at least the left triangle λ')

Output: The GSO λ' of the reduced basis, and the unitary transformation U

```
$k \leftarrow 1$,$U = I_{2n}$ (or $U = U_{\text{init}}$ initially given by the user)
**while** $k \le n$ **do**
    **if** $k = 1$ **then**
        compute $\lambda_{n+1,n+1} = q^2 \lambda'_{n-1,n-1}$
        find the $2 \times 2$ unimodular transformation $P$ that Lagrange-reduces the GSO of $C_1$
        ensure that the determinant of $P$ is not $-1$, negate one row of $P$ if necessary
        Apply $P$ on the two middle rows of $\lambda'$ and $U$, and update $\lambda'_{n,n}, \lambda'_{n+1,n}$
        $k \leftarrow 2$
    **end if**
    **for** $i = n + 2 - k$ to $n + k$ **do**
        $r \leftarrow \lfloor \lambda_{i,n+1-k} / \lambda_{n+1-k,n+1-k} \rceil$
        $\mathbf{u}_i \leftarrow \mathbf{u}_i - r\,\mathbf{u}_{n+1-k}$ and $\lambda'_i \leftarrow \lambda'_i - r\,\lambda'_{n+1-k}$
        $\mathbf{u}_{n+k} \leftarrow \mathbf{u}_{n+k} + r\,\mathbf{u}_{2n+1-i}$ and $\lambda'_{n+k} \leftarrow \lambda'_{n+k} + r\,\lambda'_{2n+1-i}$ **if** $i \le n$
        $\mathbf{u}_{n+k} \leftarrow \mathbf{u}_{n+k} - r\,\mathbf{u}_{2n+1-i}$ and $\lambda'_{n+k} \leftarrow \lambda'_{n+k} - r\,\lambda'_{2n+1-i}$ **if** $n+1 \le i \le n+k-1$
    **end for**
    **if** Lovász does not hold for the pair $(n - k + 1, n - k + 2)$ **then**
        compute $\lambda_{n+k,n-k+2}$ using Lemma 1
        swap $\mathbf{u}_{n+k} \leftrightarrow \mathbf{u}_{n+k-1}$ and $\lambda'_{n+k,j} \leftrightarrow \lambda'_{n+k-1,j}$ for $1 \le j \le n - k$
        swap $\mathbf{u}_{n-k+2} \leftrightarrow \mathbf{u}_{n-k+1}$ and $\lambda'_{n-k+2,j} \leftrightarrow \lambda'_{n-k+2,j}$ for $1 \le j \le n - k$
        update $\lambda'_{n-k+2,n-k+1}$ and $\lambda'_{i,n-k+1}, \lambda'_{i,n-k+2}$ for $n - k + 3 \le i \le n + k$ using the same swap formula as standard LLL
        $k \leftarrow k - 1$
    **else**
        $k \leftarrow k + 1$
    **end if**
**end while**
```

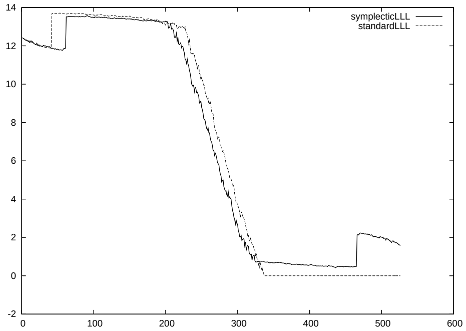
symplecticity neither does the pair $(n + k - 1, n + k)$), then the pairs of consecutive vectors $(\mathbf{b}_{n-k+1}, \mathbf{b}_{n-k+2})$ and $(\mathbf{b}_{k-1}, \mathbf{b}_k)$ are swapped and the counter k is decremented, otherwise k is incremented. The loop goes on until k eventually reaches the value $n + 1$.

Experiments show that this basic symplecticLLL algorithm is already as fast as LLL in dimension 80, and becomes faster in higher dimension. The quality of the output basis is similar to the one of StandardLLL. (see Figure 1). Note also that the curve of $\log \|\mathbf{b}_i^*\|$ obtained after symplecticLLL is symmetric because of Corollary 1. In other words, both the basis and its dual are reduced in the same time. We now describe optimizations.

6.4 Optimizations

The following two optimizations do not modify the output of the algorithm, but considerably speed up the algorithm in practice:

Fig. 1. Quality of the input basis ($\log_2 \|\mathbf{b}_i^*\|$ as a function of i)



Early reduction. Let C_i be the $2i$-dimensional central projection of the input basis for $2 \leq i \leq n$. Suppose that Algorithm 4 found the unimodular matrix $U_p \in Sp(2p, \mathbb{Z})$ such that $U_p C_p$ is symplecticLLL reduced and the reduced GSO λ'_p. If we want to reduce the initial C_{p+1} using Algorithm 4, we know that when the counter k reaches $p+1$ for the first time, the current state U_{p+1} and λ'_{p+1} is:

$$U_{p+1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & U_p & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{and} \qquad \lambda'_{p+1} = \begin{pmatrix} \lambda_{n-p,n-p} & 0 \\ U_p \lambda_{\cdot,n-p} & \lambda'_p \\ \lambda_{n+p+1,n-p} & 0 \end{pmatrix}.$$

So we can launch Algorithm 4 on λ'_{p+1} with $U_{\mathrm{init}} = U_{p+1}$ to finish the reduction. Using this simple trick for $p = 2$ to n, the first transformations of Algorithm 4 apply to lower dimensional matrices. On NTRU matrices, the execution is almost two times faster than the basic symplecticLLL. In the Standard LLL algorithm, the analogue is to update only the first p rows of the GSO, where p is the maximum index of vectors already modified by the main loop of LLL since the beginning of the execution.

Integer triangular matrices. This last optimization only works on matrices for which every $\|\mathbf{b}_k^*\|^2$ is an integer (at the beginning). It is the case of all NTRU public key matrices, and all integer triangular matrices. The key result is that in the previous algorithm, each λ'_p is initially divisible by $D_{n-p} = \prod_{i=1}^{n-p} \|\mathbf{b}_i^*\|^2$. The only improvement is to use reduced GSO λ'_p / D_{n-p} instead of λ'_p in the previous algorithm. Then the first transformations of Algorithm 4 apply to matrices of lower dimension, but also with smaller coefficients. On NTRU matrices, the execution becomes almost 4 times faster than the basic symplecticLLL.

Table 3. Experimental results

n *half of dim.*	q *max. coefs*	Standard LLL *in seconds*	SympLLL Early red. *in seconds*	SympLLL Integer triang. optim. *in seconds*	**speedup Early reduction**	**speedup integer triang.**
40	64	3.09	2.27	1.98	1.36	1.56
83	64	26.89	6.62	4.46	4.06	6.02
107	64	44.7	6.13	4.51	7.29	9.91
167	128	410.8	98.86	65.40	4.15	6.28
253	128	2028	553	294	3.66	6.89
317	128	3688	1131	519	3.26	7.10

References

1. L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
2. A.-M. Bergé. Symplectic lattices. In *Quadratic forms and their applications (Dublin, 1999)*, volume 272 of *Contemp. Math.*, pages 9–22. Amer. Math. Soc., Providence, RI, 2000.
3. P. Buser and P. Sarnak. On the period matrix of a Riemann surface of large genus. *Invent. Math.*, 117(1):27–56, 1994. With an appendix by J. H. Conway and N. J. A. Sloane.
4. H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1995. Second edition.
5. J. Conway and N. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1998. Third edition.
6. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *Proc. of Eurocrypt '97*, volume 1233 of *LNCS*. IACR, Springer-Verlag, 1997.
7. C. Gentry, J. Jonsson, J. Stern, and M. Szydlo. Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001. In *Proc. of Asiacrypt '01*, volume 2248 of *LNCS*. Springer-Verlag, 2001.
8. C. Gentry and M. Szydlo. Cryptanalysis of the revised NTRU signature scheme. In *Proc. of Eurocrypt '02*, volume 2332 of *LNCS*. Springer-Verlag, 2002.
9. G. H. Golub and Charles F. Van Loan. *Matrix Computations*. The John Hopkins University Press, third edition, 1996.
10. 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.
11. J. Hoffstein, N. A. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In *Proc. of CT-RSA*, volume 2612 of *LNCS*. Springer-Verlag, 2003.
12. J. Hoffstein, J. Pipher, and J. Silverman. NTRU: a ring based public key cryptosystem. In *Proc. of ANTS III*, volume 1423 of *LNCS*, pages 267–288. Springer-Verlag, 1998. First presented at the rump session of Crypto '96.

13. J. Hoffstein, J. Pipher, and J. H. Silverman. NSS: An NTRU lattice-based signature scheme. In *Proc. of Eurocrypt '01*, volume 2045 of *LNCS*. Springer-Verlag, 2001.
14. N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and coding (Cirencester, 1997)*, volume 1355 of *Lecture Notes in Comput. Sci.*, pages 131–142. Springer, Berlin, 1997.
15. N. A. Howgrave-Graham, P. Q. Nguyen, D. Pointcheval, J. Proos., J. H. Silverman, A. Singer, and W. Whyte. The impact of decryption failures on the security of NTRU encryption. In *Proc. of the 23rd Cryptology Conference (Crypto '03)*, volume 2729 of *LNCS*, pages 226–246. IACR, Springer-Verlag, 2003.
16. IEEE P1363.1 Public-Key Cryptographic Techniques Based on Hard Problems over Lattices, June 2003. IEEE., Available from `http://grouper.ieee.org/groups/1363/lattPK/index.html`.
17. B. A. LaMacchia. Basis reduction algorithms and subset sum problems. Technical Report AITR-1283, 1991.
18. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.
19. D. S. Mackey, N. Mackey, and F. Tisseur. Structured factorizations in scalar product spaces. *SIAM J. of Matrix Analysis and Appl.*, 2005. To appear.
20. A. May and J. H. Silverman. Dimension reduction methods for convolution modular lattices. In *Proc. of CALC '01*, volume 2146 of *LNCS*. Springer-Verlag, 2001.
21. P. Q. Nguyen and O. Regev. Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures. In *Proc. of Eurocrypt '06*, *LNCS*. Springer-Verlag, 2006.
22. 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.
23. 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.
24. H.-G. Quebbemann. Modular lattices in Euclidean spaces. *J. Number Theory*, 54(2):190–202, 1995.
25. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
26. M. Seysen. Simultaneous reduction of a lattice basis and its reciprocal basis. *Combinatorica*, 13(3):363–376, 1993.
27. B. M. M. de Weger. Solving exponential Diophantine equations using lattice basis reduction algorithms. *J. Number Theory*, 26(3):325–367, 1987.
28. H. Weyl. *The classical groups.* Princeton Landmarks in Mathematics. Princeton University Press, 1997. Their invariants and representations, Fifteenth printing, Princeton Paperbacks.