A Central Problem in the Algorithmic Geometry of Numbers: Lattice Reduction

Around the Algorithm of Lenstra, Lenstra, Lovász*

Brigitte Vallée
Université de Caen, F-14032 Caen Cedex
Département de Mathématiques

We first recall the general framework of the Geometry of Numbers, both classical and algorithmical; we define the main problems of lattices and concentrate on their numerous applications. We then define different notions of reduction and describe Gauss' algorithm that completely solves the problem in the two dimensional case. After this we make precise the notion of Lovász reduction. We then describe the Lenstra-Lenstra-Lovász algorithm which builds in polynomial time a Lovász reduced basis and we analyze its complexity. Then we mention other algorithms stemming from it, which allow simple computations in lattices. We continue by presenting the range of the applications of such an algorithm, starting with internal applications in lattice theory. We conclude with external applications in different areas: Theory of Numbers, Algebra and Cryptography. This survey aims to describe the amplitude of the posed problems as well as the quality of the answers obtained. For more details the reader is refered to the given references. A general reference which contains an exhaustive bibliography and detailed technical treatment on the subject is R. Kannan, Algorithmic Geometry of Numbers, Annual reviews in Computer Science (1988).

1. General lattice problems

$\mathbb{R}^p$ is endowed with the canonical Euclidean structure, $|v|$ denotes the norm of v, and $(u\,|\,v)$ is the scalar product of u and v. In the sequel $[r]$ denotes the integer which is closest to the real r.

A lattice of $\mathbb{R}^p$ is the set of linear combinations, with integer coefficients, of linearly independent vectors of $\mathbb{R}^p$; this is a discrete subgroup of $\mathbb{R}^p$. Such a lattice is called an 'integer lattice' if it is included in $\mathbb{Z}^p$.

If $b = (b_1, b_2, \ldots, b_n)$ is a system of n linearly independent vectors of $\mathbb{R}^p$ $(n \leqslant p)$ then the lattice generated by b, denoted by $L(b)$, is the set $\{\Sigma_{i=1}^{n} \lambda_i b_i \mid \lambda_i \in \mathbb{Z}\}$ and n is called the rank or dimension of the lattice.

* Translated from the French 'Un Problème Central en Géometrie Algorithmique des Nombres: La Réduction des Réseaux. Autour de l'Algorithme de Lenstra, Lenstra, Lovász' by Evangelos Kranakis. Informatique théorique et Applications 23 (1989), 345-376.

CWI Quarterly 3, 95-120

1.1. The determinant of a lattice

Let $(b_1,b_2,\ldots,b_n)$ and $(c_1,c_2,\ldots,c_n)$ be two bases of the same lattice, and let B and C be their matrices $(n\times p)$ in the canonical basis of $\mathbb{R}^p$. There exists a unimodular matrix U (matrix with integer entries whose determinant is ± 1) satisfying $C = U\cdot B$.

Note that the n-dimensional volume of the parallelepiped constructed over an arbitrary base of the lattice is independent of the base: it is an invariant of the lattice, denoted by $d(L)$, and is called the determinant of the lattice. This quantity is easily calculated: if $G(b)$ denotes the Gram matrix of the basis b, i.e. the matrix B^tB with general entry $(b_i|b_j)$, then

$$\det G(b) = \det(B^tB) = d(L)^2.$$

A lattice is pseudo-integer if it admits a basis b whose Gram matrix $G(b)$ has integer coefficients—then, all these bases satisfy the same property. In algorithmic practice one is restricted to pseudo-integer or more often to integer lattices.

1.2. The successive minima of lattices

In a lattice, there are other objects which depend only on the lattice and not on the basis defining it; in particular, the successive minima. For an n-dimensional lattice L in $\mathbb{R}^p$ the i-th minimum $\Lambda_i(L)$ is defined as the smallest positive real t for which there exist i linearly independent vectors v of L satisfying $|v|^2 \leqslant t$.

It is clear that the n numbers $\Lambda_i(L)$ are well defined and satisfy

$$\Lambda_1(L) \leqslant \Lambda_2(L) \leqslant \cdots \leqslant \Lambda_n(L).$$

Also, there exists a set—not necessarily unique—of n linearly independent vectors of L, also called successive minima and denoted by $\lambda_i(L)$, satisfying

$$|\lambda_k(L)|^2 = \Lambda_k(L), \quad k = 1,2,\ldots,n.$$

In particular, $\lambda_1(L)$ denotes the shortest vector of the lattice L.

1.3. Theoretical and practical lattice problems

The theoretical problems are of two types:

1. Connect the intrinsic quantities of a lattice, in particular the $\Lambda_i(L)$ and $d(L)$.
2. Construct or exhibit, by algorithmic means, the intrinsic objects of a lattice, in particular its successive minimal vectors $\lambda_i(L)$.

Sometimes it happens that in wanting to solve the first type of problems one finds an explicit method which solves at the same time the second type of problems. This is never the case for this theory, and this explains the necessity and importance of algorithmics in the geometry of numbers.

With regard to practical problems, one poses different sorts of questions, which are essentially computation problems on integer lattices; we cite a few among them:

— Let L be a lattice given by a basis b and let v be a vector. Decide whether or not v is a member of the lattice $L(b)$.
— Let L be a lattice generated by a system c of not necessarily independent vectors. Find a basis b of the lattice.
— Let $b=(b_1,b_2,\ldots,b_n)$ be a system of n vectors of $\mathbb{Z}^p$. Find the lattice L of relations, i.e. the set

$$\{v = (v_1,v_2,..,v_n)\in\mathbb{N}^n \mid \sum_{i=1}^{n} v_i b_i = 0\}$$

1.4. Theoretical, non-constructive results: Hermite, Minkowski [7]

Hermite showed the existence of a constant γ_n, called Hermite's constant, satisfying.

$$\gamma_n = \max\{\frac{\Lambda_1(L)}{d(L)^{2/n}} \mid L \text{ is a lattice of rank } n\}.$$

Minkowski later showed that also

$$d^2(L) \leqslant \prod_{i=1}^{n} \Lambda_i(L) \leqslant \gamma_n^n d^2(L).$$

The proofs of these results are non-constructive and the first known upper bound of γ_n, due to Hermite, is exponential in n:

$$\gamma_n \leqslant (\frac{4}{3})^{n-1}.$$

This upper bound is optimal only in the case $n=2$. Later, Minkowski refined this upper bound to obtain the following inequalities:

$$\gamma_n \leqslant \begin{cases} \frac{2}{3}n & \text{if } n \text{ is even} \\ \frac{1}{2}n & \text{if } n \text{ is odd.} \end{cases}$$

Only the first eight values of this constant are 'exactly' known (see [28]).

1.5. Algorithmic problems of lattices

The following problems are posed for an integer lattice L given by its rank n and a basis b of length $M=\max_i|b_i|$:

1. determine $\lambda_1(L)$, a shortest vector of the lattice L,
2. determine the $\lambda_i(L)$, a sequence of successive minimal vectors of the lattice L.

Interest on these problems is due to the conjunction of the following three factors:

— these are problems which are probably difficult
— which nevertheless admit approximate solutions
— that in turn would permit the resolution of other varied and essential problems in the theory of numbers in algebra or in cryptography.

In this paper we are going to describe the second factor of interest in Sections 2 and 3, and the third factor in Section 4. We discuss for a moment the first factor of interest.

1.6. The probable difficulty of these problems

The second problem is NP-hard in the parameters $(n, \log M)$ [22]. Nothing is known on the easiness of the first: at this moment no polynomial algorithm in $(n, \log M)$ is known which solves this problem, which at first sight is easier than the second. Current opinion seems to also believe on its 'hardness', based on the following three arguments:

— this problem is NP-hard for the norm sup [27]);
— the associated non-homogeneous problem, which consists of searching the point on a lattice L with minimal distance from a given point of $\mathbb{Q}^n$, is also NP-hard, even for the Euclidean norm [27];
— The inequalities of Minkowski are not any sharper for the first minimum than they are for the geometric mean of the other successive minima.

Two different but complementary points of view exist in order to overcome this almost certain difficulty:

1. Search for 'approximate' polynomial algorithms in $(n, \log M)$.
2. Search, for fixed dimension n, for exact polynomial algorithms in $\log M$.

By the way, these are not divergent points of view.

The second point of view is fruitful in small dimension: Gauss' algorithm is a polynomial algorithm that finds the two minima of a lattice in dimension 2. Its complexity is well-known and it can be generalized to dimension 3 [25].

The first point of view uses low-dimensional algorithms of this type—which are exact for low dimensions—in order to construct approximate algorithms in higher dimensions. One obtains a so-called reduced basis: this is a basis formed by 'very short' and 'very orthogonal' vectors which permits the good description of the lattice and, in addition,

1. gives a good approximation of the intrinsic objects of the lattice
2. enables us to calculate easily in the lattice.

We will see in Section 4 how such a basis can resolve, in an astonishingly satisfying manner, the totality of algorithmic problems, both theoretical and practical. We describe now more precisely the notion of reduction of lattices.

2. Lattice reduction

We search for a basis consisting of 'almost orthogonal vectors'; we remark that in general a lattice does not have an orthogonal basis. The Gram-Schmidt orthogonalization procedure associates with a basis b of a lattice in $\mathbb{R}^p$ an orthogonal basis b^* of the $\mathbb{Q}$-vector space generated by b, but in general this is not included in the lattice $L(b)$.

2.1. The Gram-Schmidt orthogonalization procedure

This procedure associates with an ordered system $b=(b_1,b_2,\ldots,b_n)$ the system $b^*=(b_1^*,b_2^*,\ldots,b_n^*)$ and the matrix $m=(m_{ij})$ which expresses the system b in the system b^* and is defined as follows:

(i) $b_1^*=b_1$

(ii) b_i^* is the orthogonal projection of b_i to the subspace H_{i-1} generated by the first $i-1$ vectors of b. One can write

$$b_i = b_i^* + \sum_{j<i} m_{ij} b_j^*,$$

where m_{ij} is defined by the relation $m_{ij}=\frac{(b_i|b_j^*)}{|b_j^*|^2}$, which allows easily the calculation of the b_i^* and the m_{ij} by induction on i.

The matrix m is lower-triangular possessing a diagonal consisting of 1's. We remark that $d(L)$ is equal to the product of the squares of the lengths of the vectors b_i^*.

If b is a system of n vectors of $\mathbb{Z}^p$ of length M, the calculation of the pair (b^*,m) is polynomial in the size of $(n,\log M)$. Let L_i be the lattices generated by the first i vectors of b and let $d_i=d(L_i)^2$. Hadamard's inequality gives

$$d_i \leqslant \prod_{j=1}^{i} |b_j|^2 \leqslant M^{2i}.$$

On the other hand, the rationals appearing in b^* or in m have the d_j's as denominators; more precisely,

$$|b_i^*|^2 = \frac{d_i}{d_{i-1}}, \quad \text{for} \quad 2\leqslant i\leqslant n$$

$$d_{i-1}b_i^* \in \mathbb{Z}^p, \quad \text{for} \quad 2\leqslant i\leqslant n$$

$$d_j m_{ij} \in \mathbb{Z}, \quad \text{for} \quad 1\leqslant i<j\leqslant n.$$

We also remark that the quantity $D=\prod_{j=1}^{n-1} d_j$ is a common denominator for all the rationals appearing in the pair (b^*,m).

2.2. The defects of length and orthogonality

Fortunately, the two desired conditions—fairly short and almost orthogonal vectors—are compatible by virtue of results of Hermite and Minkowski. Let $b=(b_1,b_2,\ldots,b_n)$ be a basis of a lattice L; the following two parameters measure the quality of the basis:

The ratio $\rho(b)=\frac{\Pi_{i=1}^{n}|b_i|^2}{d(L)^2}$ is called 'orthogonality defect' of the base b.

The ratio $\mu_i(b)=\frac{|b_i|^2}{\Lambda_i(L)}$ is called the 'i-th length-defect' of the base b.

The results of 1.4 give the connection between these two parameters which satisfy the double inequality:

$$1 \leqslant \frac{\rho(b)}{\Pi_{i=1}^{n} \mu_i(b)} \leqslant \gamma_n^n.$$

2.3. The different notions of reduction

There is more than one reduction; historically one distinguishes four notions of reduction. Minkowski sought to minimize directly the length-defects. In the other three reductions, one wants to minimize $|b_i^*|$ at the same time with $|b_i|$; these reductions are described easily by the matrix m which expresses the system b as a function of the system b^*: these are the reductions in the sense of Korkine-Zolotarev, Siegel and finally Lovász.

One can show that the reduction in the sense of Siegel is the most general one [7]: every basis reduced by one of the other three ways is also reduced in the sense of Siegel. This reduction, which is also the least fine, suffices in many applications because the reduced base obtained is of fairly good quality.

The first two reductions—the one of Minkowski is in a sense the most natural, and the one of Korkine-Zolotarev appears to be the best with respect to both defects of orthogonality and length [13]—cannot be obtained, it seems, in polynomial time. Therefore we will favor the other two reductions and show that they can be obtained 'easily'.

Although these reductions differ a bit in arbitrary dimension, they all coincide in dimension 2 with the celebrated Gauss reduction which we now describe, before making precise these different notions of reduction.

2.4. The reduction of Gauss in dimension 2 [4]

The successive minima of a lattice always form a system of independent vectors, but in general they do not generate a lattice; however, this is true in small dimension:

> The successive minima of a lattice of dimension $n \leqslant 4$ form a basis of the lattice, called minimal basis of the lattice: in this case this is the 'best basis' of the lattice.

Algorithm Gauss
Input: a basis (u,v) of a lattice L
Output: A minimal basis (u,v) of the lattice L
repeat
1. if necessary, exchange u and v in order that $|u| \leqslant |v|$
2. translate v parallel to u in order to shorten it to the maximum: more precisely, choose in the set $\{w = \epsilon(v - mu) \mid \epsilon = \pm 1,\ m \in \mathbb{Z}\}$ the vector which satisfies $0 \leqslant \frac{(w|u)}{(u|u)} \leqslant \frac{1}{2}$ (this last is easily calculable in terms of $r = \frac{(v|u)}{(u|u)}$: one chooses $m = [r]$ and $\epsilon = \text{sign}(r - m)$)

until $|v| \geqslant |u|$

In dimension 2, the algorithm of Gauss constructs in polynomial time a minimal base of the lattice; it generalizes, in dimension 2, the centered Euclidean algorithm:

$$a = bq + r \quad \text{with} \quad -\frac{b}{2} < r \leqslant \frac{b}{2}.$$

One can modify the stopping test by changing it into a less refined test. If t is a real satisfying the double inequality $1 < t \leqslant \sqrt{3}$, one obtains an algorithm, called t-Gauss, which 'runs' a little less slowly, but which possesses a comparable exit configuration: the triangle constructed from the basis contains the two minima of the lattice.

Algorithm t-Gauss
Input: a base (u,v) of a lattice L
Output: A quasi-minimal base (u,v) of L
repeat
1. exchange u and v in order that $|u| \leqslant |v|$
2. translate v parallel to u

until $|v| \geqslant \frac{1}{t}|u|$

2.5. Study of the complexity of Gauss' algorithm
Let $k(t)$ and k be the number of iterations of the algorithms t-Gauss and Gauss, respectively, on the same basis (u,v) of length M. It is clear that $k(t) \leqslant \log_t M + 1$. One can show that for $1 \leqslant t \leqslant \sqrt{3}$, we have $k(t) \leqslant k \leqslant k(t) + 1$, which demonstrates the (not entirely trivial) polynomial complexity of Gauss' algorithm. A more refined study of the worst case of Gauss' algorithm [25] gives the best upper bound possible

$$k \leqslant \log_{1+\sqrt{2}} M + 3$$

which is similar to the bound obtained in the centered Euclidean Algorithm [3].

2.6. The effect of Gauss' algorithm on the orthogonalized (u^*, v^*)
At the beginning of the algorithm of Gauss the vector v satisfies the two conditions

i) $|v| \geqslant \frac{1}{t}|u|$

ii) $0 \leqslant (u|v) \leqslant \frac{1}{2}(u|u)$

The orthogonal projection of v on u, by definition equal to v^*, satisfies

$$|v^*|^2 \geqslant \left(\frac{1}{t^2} - \frac{1}{4}\right)|u^*|^2.$$

If we put $s=\sqrt{\frac{4t^2}{4-t^2}}$ we obtain that $|u^*|\leqslant s|v^*|$. We remark that if $1\leqslant t\leqslant\sqrt{2}$, then $\frac{4}{3}\leqslant s^2\leqslant 12$. The value $s^2=2$ corresponding to the value $t=\frac{2}{\sqrt{3}}$ is usually chosen in order to simplify the calculations.

2.7. The properness of a base

The simplest idea for bringing b closer to b^* is to reduce the coefficients of the matrix m without modifying neither b nor the lattice $L(b)$; this justifies the following definition which uses the notation of 2.1:

> A basis $b=(b_1,b_2,\ldots,b_n)$ is proper if all the entries m_{ij} (for $j<i$) of the associated matrix m have value $\leqslant\frac{1}{2}$.

We make this condition geometrically explicit: each vector b_i is orthogonally projected on the hyperplane H_{i-1} in the interior of the rectangular parallelepiped constructed by the b_j's, for $j<i$, and defined as being the set of vectors

$$\sum_{j=1}^{i-1}\alpha_j b_j^* \quad \text{for} \quad -\tfrac{1}{2}<\alpha_j\leqslant\tfrac{1}{2}.$$

There exists a 'totally-proper' algorithm which given a system $b=(b_1,b_2,\ldots,b_n)$ makes it proper; this consists of a succession of calls of a procedure Proper (i) which generalizes the second stage of Gauss' algorithm; this procedure translates b_i parallel to each vector b_j, for $j<i$, and does not modify neither b_i^* nor $L(b)$.

```
Algorithm Proper (i)
 for     j := i − 1 to 1 do
         r_j := [m_ij];
         b_i := b_i − r_j b_j;
Algorithm Totally-Proper
 for     i := 2 to n do
         Proper (i)
```

2.8. Reduction in the sense of Siegel

The fact that a base is proper does not guarantee that it should be almost orthogonal; at the moment it is known that the projection of each b_{i+1} on H_i is very small. To be able to minimize the angle θ_{i+1} formed by b_{i+1} with the hyperplane H_i it is necessary in addition to minimize the orthogonal projection of b_{i+1} on H_i, i.e. the length of b_{i+1}^*. This is the purpose of the condition in Siegel's reduction:

$$|b_{i+1}^*|\geqslant\frac{1}{s}|b_i^*|,\quad 1\leqslant i\leqslant n-1.$$

A proper base which also satisfies Siegel's condition with parameter s is called s-reduced in the sense of Siegel.

This condition is enough in order to ensure the quality of the basis and order to majorize the length- and orthogonality-defects. One obtains the following results:

$$|\sin\theta_i| \geqslant \frac{1}{s^{i-1}} \quad \text{and} \quad |b_i| \leqslant |b_i^*| s^{i-1}, \quad \text{for} \quad 1 \leqslant i \leqslant n-1.$$

From this one can deduce

$$\rho(b) \leqslant s^{n(n-1)/2} \text{ and } s^{-2(i-1)} \leqslant \mu_i(b) \leqslant s^{2(n-1)}, \quad \text{for} \quad 1 \leqslant i \leqslant n.$$

2.9. Reduction in the Sense of Lovász

Two essential questions remain open.

i) Does every lattice admit a reduced basis in the sense of Siegel? If yes, for which values of the parameter s?

ii) If yes, does there exist an algorithm, which starting from an arbitrary base of length M of a lattice L of rank n, constructs a basis in the sense of Siegel in time polynomial in $(n, \log M)$?

All these questions will receive positive answers. Given the fact that the existence and the constructibility of a proper basis does not pose any problems, the question can be rephrased as follows:

How can Siegel's condition be guaranteed?

We will try to guarantee a much stronger condition: the condition of Lovász. We have remarked in 2.6 that Gauss' algorithm—in dimension 2—gives an inequality concerning the orthogonalized vectors which is similar to Siegel's. More precisely, let P_i be the orthogonal of H_{i-1} in H_{i+1} and let B_i be the system ('box') formed by the projections u_i and v_i of b_i and b_{i+1}, respectively, on P_i. By definition of b_{i+1}^* we have $b_{i+1}^* = v_i^*$. If one applies the t-Gauss algorithm on the systems B_i, we obtain, at the end, the following three conditions which are valid for all $1 \leqslant i \leqslant n-1$ (the parameters t and s are connected as in 2.6 by the relation $s = \sqrt{\frac{4t^2}{4-t^2}}$):

i) $0 \leqslant (v_i|u_i) \leqslant \frac{1}{2}(n_i|u_i)$

ii) $|v_i| \geqslant \frac{1}{t}|u_i|$

iii) $|u_i^*| \leqslant s|v_i^*|$

The first condition is a 'properness' condition; the second is Lovász' condition; the third is Siegel's condition. According to 2.6, $(i)+(ii) \Rightarrow (i)+(iii)$. This justifies the following definition.

A proper basis which satisfies Lovász' condition for the parameter t is called t-reduced in the sense of Lovász

and allows us to confirm that

if s and t are connected by the relation $s = \sqrt{\frac{4t^2}{4-t^2}}$ then a t-reduced basis in the sense of Lovász is s-reduced in the sense of Siegel.

3. The Algorithm of Lenstra-Lenstra-Lovász [15]

This algorithm constructs a t-reduced basis in the sense of Lovász, starting from a basis b of length M in a lattice of rank n, in time polynomial in the size $(n, \log_t M)$ of the input. For $t = 1$ this algorithm terminates without knowing, at present, how to make precise its complexity any further.

3.1. The main phases of the algorithm

This algorithm consists of three principal phases:

- an initialization phase; essentially it consists of calculating the system b^*, the matrix m and the list l formed by the elements $l_i = |b_i^*|^2$, by the Gram-Schmidt orthogonalization procedure described in 2.1. These last two objects—and only them—will be essential throughout the algorithm.
- translation phases of the vectors b_i in parallel on H_{i-1} which are realized by the procedure **Proper** described in 2.7. Recall also that these phases do not modify the system b^*.
- exchange phases of the vectors b_i and b_{i+1} in order to realize,
  - i) the condition of t-Gauss on the system B_i defined in 2.9
  - ii) and the condition of s-Siegel on the system b^*.

The triple (b^*, m, l) is modified at the time of this exchange and we must recalculate a part of it by means of the NewOrtho procedure which we are going to describe a bit later.

The choice—translate or exchange—is done by applying the test of the t-Gauss algorithm on the system B_i and this choice is then reflected on the vectors $(b_1, b_2, \ldots, b_n)$ of the basis b.

We remark that the vectors u_i and v_i of the system B_i are to be found from the matrix m: these are the row-vectors of the box B_i depicted inside the matrix below.

$$m = \begin{array}{c} \\ b_1 \\ b_2 \\ \vdots \\ b_i \\ b_{i+1} \\ \vdots \\ b_n \end{array} \begin{array}{c} \begin{array}{ccccccccc} b_1^* & b_2^* & & & b_i^* & b_{i+1}^* & & \cdots & b_n^* \end{array} \\ \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ m_{2,1} & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ m_{i1} & m_{i2} & \cdots & 1 & 0 & \cdots & 0 \\ m_{i+1,1} & m_{i+1,2} & \cdots & m_{i+1,i} & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{ni} & m_{n,i+1} & \cdots & 1 \end{bmatrix} \end{array}.$$

In particular we have the following relations

$$u_i = b_i^*, \quad v_i = b_{i+1}^* + m_{i+1,i} b_i^* \qquad (\S)$$

and the three quantities entering in Gauss' algorithm applied to B_i are calculated easily as a function of the list l and the matrix m:

$$|u_i|^2 = l_i, \quad |v_i|^2 = l_{i+1} + m_{i+1,i}^2 l_i$$

and also

$$\frac{(v_i|u_i)}{(u_i|u_i)} = m_{i+1,i}.$$

3.2. *General description of the algorithm*

Algorithm LLL(t);
Input: a basis b of a lattice in $\mathbb{R}^P$ of rank n
Output: a basis b of L which is t-reduced in the sense of Lovász
Gram-Schmidt; */this gives output (b^*, m, l)/*
$i := 1$;
while $i < n$ do
1. translate v_i parallel to u_i, where v_i, u_i are as in (§), and b_{i+1} parallel to b_i, i.e.
calculate $r_i = [m_{i+1,i}]$ and set $v_i := v_i - r_i u_i$;
$b_{i+1} := b_{i+1} - r_i b_i$;
2. **Test** if $|v_i|^2 \geqslant \frac{1}{t^2}|u_i|^2$;
if yes, */the box B_i is reduced in the sense of t-Gauss /*
then: translate b_{i+1} parallel to b_j, for $j < i$, by means of **Proper** $(i+1)$,
/modify the index/ $i := i + 1$;
if no **then**
exchange b_i and b_{i+1};
recalculate by the procedure **NewOrtho** the triple (b^*, m, l);
the box B_{i-1} is not necessarily reduced any more;
/modify the index/ **if** $i \neq 1$ **then** $i := i - 1$

The variable i designates an index, the current index of the algorithm which varies from 1 to $n-1$: this is the largest index k for which the system $(b_1, b_2, \ldots, b_k)$ is t-reduced in the sense of Lovász. The matrix m_i formed by the first i rows and first i columns of the matrix m and the list formed by the first i systems of the list l already have the desired forms.

One then considers the $i+1$-st row of m, representing the vector b_{i+1} and carries out the operations of translation or exchange following the result of the test of t-Gauss.

Now it remains to make precise the procedure NewOrtho.

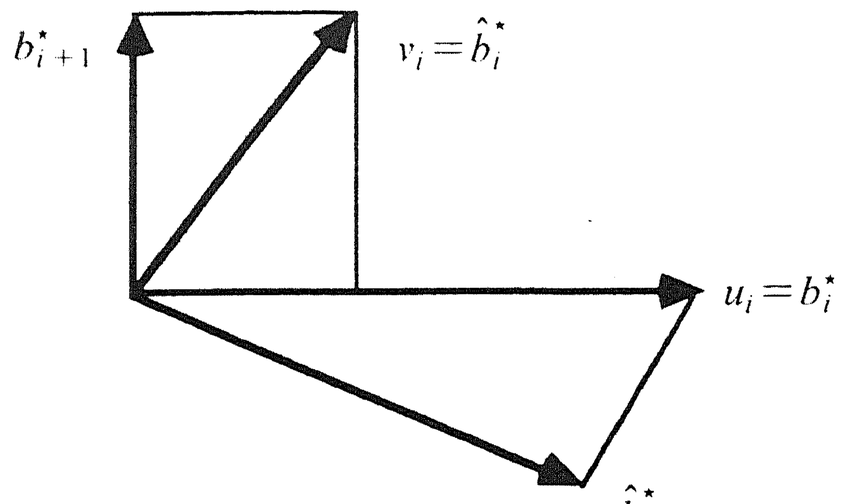
3.3. The modification of b^*, l and m which are accomplished in the second stage

If the test is positive, neither b^* nor l are modified; only the $i+1$-st row of the matrix m is replaced by a linear combination of the $i+1$-st and the preceding rows, in conformity with the description of the procedure Proper $(i+1)$.

If, on the contrary, the test is negative, the exchange of the vectors b_i and b_{i+1} in the system b modifies the two vectors b_i^* and b_{i+1}^*. Let $(\hat{b},\hat{b}^*,\hat{m})$ be the new triple calculated by the procedure NewOrtho:

v_i is the projection of $b_i = b_{i+1}$ on P_i:
one has $b_i^* = v_i$; $\hat{b}_{i+1}^*$ is the orthogonal projection of $\hat{b}_{i+1} = b_i$ on $\hat{H}_i$ and the projection of u_i on the orthogonal of v_i.

One obtains the following figure:



One also obtains the following equations:

$$\hat{m}_{i+1,i} = m_{i+1,i}\frac{|u_i|^2}{|v_i|^2}$$

and also

$$\begin{bmatrix}\hat{b}_{i+1}^* \\ \hat{b}_i^*\end{bmatrix} = \begin{bmatrix}-\hat{m}_{i+1,i} & 1-\hat{m}_{i+1,i} & m_{i+1,i} \\ 1 & & m_{i+1,i}\end{bmatrix}\begin{bmatrix}b_{i+1}^* \\ b_i^*\end{bmatrix},$$

which makes it possible to obtain the new triple $(\hat{b}^*,\hat{m},\hat{l})$ from the old triple (b^*,m,l) l according to the following formulas:

$$\hat{m}_{j,i} = m_{j,i}\cdot\hat{m}_{i+1,i} + m_{j,i+1}\frac{|b_{i+1}^*|^2}{|\hat{b}_i^*|^2},$$

$$\hat{m}_{j,i+1} = m_{j,i} - m_{j,i+1}\cdot m_{i+1,i},$$

$$\hat{l}_{i+1} = |\hat{b}_{i+1}^*|^2.$$

3.4. The number of iterations accomplished by the algorithm

The index i is an index which increases if the test in 2 is positive and decreases if this test is negative. We majorize the number k_- of times that one 'passes through 2' with a negative test as a function of t, M, n; this majorization will suffice to conclude because the number k_+ of times that one 'passes through 2' with a positive test satisfies:

$$k_+ \leqslant k_- + n - 1.$$

The total number k of iterations of the algorithm will satisfy $k \leqslant n - 1 + 2k_-$. Like in the study of the complexity of the Gram procedure done in 2.1 it is the quantity

$$D = \prod_{j=1}^{n-1} d_j$$

defined in paragraph 2.1 which is going to play an essential role. We remark that every passage through 2 that has a negative test with a given value i:

— the lattices L_j, for $j < i$ and for $j \geqslant i+1$ are not modified, thus the corresponding d_j's are not either;
— on the contrary the lattice L_i and its Gram determinant are modified.

We had previously

$$d_i = \prod_{j=1}^{i} |b_i^*|^2 = |u_i|^2 \prod_{j=1}^{i-1} |b_i^*|^2.$$

We have now

$$\hat{d}_i = \prod_{j=1}^{i} |\hat{b}_i^*|^2 = |v_i|^2 \prod_{j=1}^{i-1} |b_i^*|^2.$$

If the t-Gauss test is negative one deduces that

$$\hat{d}_i < \frac{1}{t^2} d_i \quad \text{and} \quad \hat{D} < \frac{1}{t^2} D.$$

On the other hand, in each passage through 2 with a positive test, none of the lattices L_i is modified. Hence D remains unchanged in this stage. Finally, D decreases throughout the algorithm, and one has

at the start : $D \leqslant M^{n(n-1)}$,
at the end : $D \geqslant 1$.

Hence the following majorization is obtained:

$$k_- \leqslant \frac{n(n-1)}{2} \log_t M.$$

3.5. The complexity of the algorithm

It remains to bound the size of the numbers appearing in the algorithm as a function of $(n, \log M)$. It is clear that the numbers l_i are rationals whose numerators and denominators decrease throughout the algorithm. In contrast, the situation is less clear for the integers $|b_i|^2$ and the rational matrix m: in effect everything goes well when projected on the planes P_i, but at the time of the lifting, even when chosen in a minimal manner, one cannot affirm that the size of these quantities decreases. A few techniques, which we do not develop here, allow one to show that the following majorizations are valid throughout the algorithm—including the stages of lifting (see [15] formulas (1.30)-(1.34)).

One always has:

$$|m_{ij}| \leqslant \sqrt{n}(2M)^{n-1} \quad \text{and also} \quad |b_i| \leqslant n(2M)^n$$

and therefore $|m_{ij}|$ is a rational whose numerator and denominator's sizes are majorized by a quantity which is polynomial in $(n, \log M)$. We note that these bounds have been improved by Schnorr [20]. The operations applied on these numbers are very simple: calculations of 'nearest integer' and squarings. One deduces the following theorem:

> The algorithm of Lenstra, Lenstra, Lovász associated with the parameter t constructs, starting from a basis b of length M of a lattice L of rank n, a t-reduced basis in the sense of Lovász in time polynomial in $(n, \log_t M)$. More precisely, the number of arithmetical operations applied in this algorithm is $O(n^4 \log_t M)$ and the numbers on which it operates are of size $O(n \log M)$.

3.6. Improvements of the algorithm

Schönhage [19] observes that in the original algorithm sometimes one is losing time in useless back-and-forth moves along the diagonal. His principal idea is to proceed more often with local exchanges in the interior of blocks (where the index i can only vary within a small interval of length k) and only from time-to-time with global exchanges (during which the index i can vary from 1 to n). Choosing $k = \sqrt{n}$, he obtains an improvement of the complexity of the algorithm, which requires only $O(n^3 \log M)$ arithmetic operations. Schnorr's idea [20] is to round off the rationals used in the algorithm, but preserving the precision of the results. One then uses a more powerful reduction corresponding to the value $t = 1.05$ of the parameter and mainly a new method of autocorrection in the approximate calculation of the inverse of a matrix. He shows that one can reduce the size of the integers used up to $O(n + \log M)$. One can also combine the two methods.

3.7. The special case $t = 1$

One does not know how to prove the polynomial complexity of the algorithm $LLL(1)$. It is currently conjectured that the number of iterations of this algorithm is still polynomial in the size of $(n, \log M)$.

Several arguments tend to favor such a conjecture:

— Lagarias and Odlyzko [12] have given an experimental study of this conjecture: in practice the number of iterations of $LLL(1)$ does not exceed by more than three times the number of iterations of $LLL(t)$ for $t^2=4/3$, which is a usual value of the parameter.
— We have shown in 2.5 that the number of iterations of the algorithm of t-Gauss is largely independent of the value of the parameter t.

3.8. The practicality of the algorithm

The success of the algorithm also stems from the simplicity of its implementation: this algorithm is much simpler to program than it is to understand, something not so usual for an algorithm!

All the operations of the algorithm are carried out on the system b or on the triple (b^*,m,l); it is easy to be convinced that b^* need only be calculated at the initialization phase, and that it can be disposed of after this. One can thus only work with the three parameters b,m,l. We now show how simple adaptations of the algorithm LLL permit the satisfactory resolution of practical problems concerning the basis:

— find a basis of a lattice
— find linear integer relations.

3.9. The search for a lattice-basis from a system of generators [5]

Let $b=(b_1,b_2,\ldots,b_n)$ be a system of generators of a lattice L of rank n; it is desired to find a basis of the lattice in order to be able to calculate the determinant of the lattice, for example. The idea is to operate as if the b_i were independent.

One can generalize the Gram-Schmidt orthogonalization procedure to a system of non-independent vectors: we obtain a pair (b^*,l) and a list I formed from the indices i for which b_i^* and l_i are zero. By definition, L' is the lattice generated by the system $\{b_i: i\notin I\}$, which by definition is a system of independent vectors. Certainly the two lattices L,L' generate the same vector space of dimension s and $L'\subset L$. Also, $d(L)\leqslant d(L')$.

The idea is therefore to make the indices $i\in I$ decrease until they are all at the beginning. Then the two lattices will be the same and the system of vectors corresponding to the final indices will be the desired system. For this we use a modified LLL algorithm whose general structure is similar to that of the initial algorithm:

Algorithm for the Construction of Basis

Input: a system $b=(b_1,b_2,\ldots,b_n)$ of generators of a lattice L of $\mathbb{R}^P$.

Output: a basis b of L

Gram;
$q:=0$;
for $i\in I$ **do**
$q:=q+1$;
for $j:=i-1$ **to** 0 **do**
1. **while** $l_j\neq 0$ **do**
translate v_j in parallel on u_j and b_{j+1} in parallel on b_j;
exchange b_j and b_{j+1};
recalculate the triple (b^*,m,l) by the procedure NewOrtho;
2. Translate b_{j+1} in parallel on b_k for $k<j$ by means of **Proper** $(j+1)$

In the initialization procedure one calculates the columns of the matrix m with index $i\notin I$ as before. The columns of index $i\in I$ will be equal, by convention, to those of the corresponding identity matrix. Then one also proceeds there by a succession of exchanges and translations, but only on the boxes B_i associated with an index i satisfying $i+1\in I$; for these indices, the boxes contain two collinear vectors u_i and v_i; since they are now flat, the Gauss algorithm coincides with Euclid's algorithm, and the procedure NewOrtho is just an exchange between u_i^* and v_i^*. The study of the complexity of this algorithm is very similar to the classical algorithm. The quantity which decreases here along with the algorithm is the following:

$$D=\prod_{i\in I}2^i\prod_{i\notin I}|b_i^*|=\prod_{i\in I}2^i d(L').$$

In each internal loop , the set I is not modified, but L' itself is modified by the exchange of vectors of the box and $d(L')$ is divided by 2. In an external loop L' is not modified, but in contrast, an index $i\in I$ decreases by 1: it is the turn of the first quantity to be divided by 2.

3.10. Search for a short linear relations among n vector of $\mathbb{Z}^P$

Let $y=(y_1,y_2,\ldots,y_n)$ be a system formed by these vectors, and Y the matrix whose columns are y_i. Let $x=(x_1,x_2,\ldots,x_p)$ be a system formed by the rows of the matrix Y and L the lattice (of rank $q\leqslant p$) of $\mathbb{Z}^P$ generated by x. It is desired to construct a short vector of the so-called lattice of relations, i.e. the lattice R of vectors $v=(v_1,v_2,\ldots,v_n)$ of $\mathbb{Z}^n$ satisfying

$$\sum_{i=1}^{n}v_iy_i=0 \quad \text{and} \quad (v|x_i)=0$$

for $1 \leqslant i \leqslant p$. One proceeds in the following way.

1. Construct a basis $b = (b_1, b_2, \ldots, b_n)$ of the lattice $\mathbb{Z}^n$ in such a way that the first q vectors of b generate the same $\mathbb{Q}$ vector-space H as x.
2. The last $n - q$ vectors of the dual basis $c = (c_1, c_2, \ldots, c_n)$ of b are then a basis of the lattice of relations.
3. It remains to search for a short vector of the lattice.

It is clear that *LLL solves* stage 3. It is also true that an algorithm similar to that of the previous paragraph allows us to solve the first stage: starting from the canonical base of $\mathbb{Z}^n$ we define

1. the system b^* formed by the vectors b_i^*, orthogonal projections of the vectors b_i on the subspaces $K_{i-1} = H + H_{i-1}$.
2. the pair (m, l) and the corresponding set I whose cardinal is q, the dimension of H.

Working on the triple (b^*, m, l) by a series for exchanges and translations, we try to decrease the indices $i \in I$ until $I = \{1, 2, \ldots, q\}$: thus we have obtained the desired base b.

We remark that the first phase of this algorithm allows us to calculate a normal Hermite basis of an integer lattice, i.e. a basis b satisfying the following properties: for all i, b_i is a vector in the hyperplane generated by the first i vectors of the canonical base. This same first phase also allows us to complete a primitive vector to a unimodular matrix.

4. The range of applications of the algorithm

Here we are concerned with showing how the *LLL* algorithm allows the satisfactory solution of both internal problems in the theory of lattices as well as many other external problems. The internal applications (the first three) permit the polynomial solution, in fixed dimension, of the difficult problems of the theory of lattices. The external applications (at least the first three of them) are so essential in algorithmics that they have been a powerful motor even for the elaboration of the *LLL* algorithm. The last applications have developed afterwards by using the already existing algorithm.

This survey does not pretend to be exhaustive on this subject: we only wanted to give a glimpse of the importance of the use of this algorithm.

4.1. The shortest vector of the lattice

The first vector b_1 of the reduced basis thus obtained is very short, due to the majorization of the first length-defect of a basis which is s-reduced in the sense of Siegel. For the usual values of the parameters s and t one obtains

$$|b_1|^2 \leqslant 2^{n-1} \Lambda_1(L).$$

We will see later how this vector b_1 can play in the applications the same role as the vector $\lambda_1(L)$, even if it is longer in general. Here we pose the question:

> How can we find $\lambda_1(L)$ starting from a basis which is reduced in the sense of Siegel?

We recall that to this date no polynomial algorithm is known for resolving this

problem. Essentially, there are three algorithms of decreasing complexity but of increasing complication: the first two are due to Lenstra and the third to Kannan.

The first brings about a simple but long and systematic search. Expressing $\lambda_1(L)$ in the basis b in the form $\lambda_1(L)=\Sigma_{i=1}^{n}\beta_i b_i$, Cramer's formulas give

$$\beta_i = \frac{\det(b_1, b_2, \ldots, b_{i-1}, \lambda_1(L), b_{i+1}, \ldots, b_n)}{\det(b_1, b_2, \ldots, b_n)}.$$

Using Hadamard's inequality and the definition of $\lambda_1(L)$ we obtain $|\beta_i| \leqslant \rho(b)$. Since b is reduced in the sense of Siegel the orthogonality-defect $\rho(b)$ is majorized,

$$\rho(b) \leqslant 2^{n(n-1)/4}$$

from which one obtains

$$|\beta_i| \leqslant 2^{n(n-1)/4}$$

for all $1 \leqslant i \leqslant n$. One then deduces from it an algorithm which must calculate the length of 2^n vectors of the lattice.

The second proceeds in a recursive fashion by using a very simple geometric argument: the length $|b_n^*|$ measures the distance between two consecutive hyperplanes of the lattice which are parallel to H_{n-1}. Since b is s-reduced in Siegel's sense, these hyperplanes are well 'spaced' and one obtains, according to paragraph 2.8, for the usual values of s and t:

$$|b_n^*|^2 \geqslant \frac{1}{2^{n-1}}|b_n|^2$$

and hence

$$|b_n^*|^2 \geqslant \frac{1}{2^{n-1}}|\Lambda_1(L)|.$$

Consequently, $\lambda_1(L)$ can be found among a small number of hyperplanes which are parallel to H_{n-1} (this 'small' number is of order $2^{(n+1)/2}$): one projects successively in this finite number of affine hyperplanes, and in each of them one can use the same kind of arguments, because the preceding inequality is true when one replaces n by $n-1$ and $\lambda_1(L)$ by these projections in these hyperplanes.

One thus obtains an algorithm which considers $2^{n(n+1)/4}$ vectors of the lattice.

Since this is an affine—and not just a vector—algorithm we will return to it in paragraph 4.2.

The third algorithm [9] constructs, starting from a reduced basis in the sense of Siegel, a reduced basis in the sense of Korkine-Zolotarev; we will return to it in paragraph 4.3.

4.2. The search for a vector of a lattice L of minimal distance from a given point N

Recall that we know that this problem is NP-hard. There is an algorithm for its solution due to Babai [1] which uses the same principles as the second algorithm of the previous section and which has the same complexity.

If on the contrary one looks only for a point which is fairly near, we obtain a polynomial algorithm, also founded on the same principles, which finds a lattice point A satisfying:

$$d(N,A) \leqslant 2^{(n-1)/2} d(N,L)$$

where by definition

$$d(N,L) = \min\{d(N,A) | A \in L\}.$$

4.3. The other reductions

We have mentioned in Section 1 the reduction in the sense of Minkowski and in the sense of Korkine-Zolotarev. It has been shown that the first reduction is NP-hard and it is probable that the second one also is, because it has the same hardness as the search for a shortest vector.

Although this second reduction would in principle be NP-hard, we have already mentioned in 4.1 that an algorithm of Kannan [9] resolves the problem polynomially for a fixed dimension n. This algorithm was improved by Schnorr [21] who introduced a hierarchy of reductions which are intermediate between the one of Lovász and the one of Korkine-Zolotarev.

Herfrich-Just [6] has also constructed, by using a technique similar to that of Kannan, a polynomial algorithm for a fixed dimension n, which starting from a base which is reduced in the sense of Siegel, determines a reduced basis in the sense of Minkowski. On the other hand in dimension 3 one can construct a polynomial algorithm which, generalizing Gauss' algorithm, constructs directly a reduced basis in the sense of Minkowski, without a previous reduction in the sense of Siegel [25].

4.4. The factorization of polynomials with integer coefficients

The fundamental idea is the following: given a polynomial $f(X)$ of degree n, integer coefficients $f_0, f_1, \ldots, f_n$ and length $M(f) = \max|f_i|$ and a very good approximation of a root α of f one can determine h, the minimal polynomial of the algebraic number α which is by definition an irreducible factor of f. The approximation $\overline{\alpha}$ of α will be

— either a complex number obtained by Newton's algorithm [10],
— or a p-adic number obtained by the factorization algorithm *mod p* due to Berlekamp following a lifting by Hensel's lemma [15].

If the approximation is sufficiently sharp one can then apply a separation principle which states: there exists a δ of size polynomial in the size $(n, \log M)$ of the input such that the following two propositions are equivalent:

i) g is a multiple of h
ii) $|g(\bar{\alpha})| \leqslant \delta$ (the absolute value is either archimedean or p-adic according to the case considered).

In the first case, proposition ii) motivates us to search for a short vector of the lattice L generated by the rows v_i $(0 \leqslant i \leqslant n)$ of the matrix

$$A = \begin{bmatrix} C & C & 0 & \cdots & 0 & 1 & 0 \\ 0 & C & 0 & \cdots & 0 & \beta_1 & \gamma_1 \\ 0 & 0 & C & \cdots & 0 & \beta_2 & \gamma_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & C & \beta_n & \gamma_n \end{bmatrix}$$

where $\beta_i = \mathscr{R}(\bar{\alpha}^i)$, $\gamma_i = \mathscr{I}(\bar{\alpha}^i)$ and C is a constant which is polynomially dependent on $M(f)$ and δ.

By applying the *LLL* algorithm on the above matrix one can construct exactly h: the first vector of the reduced basis obtained, written in the form

$$v = \sum_{i=0}^{n} h_i v_i$$

permits the construction of a polynomial h of the form

$$h = \sum_{i=0}^{n} h_i X^i.$$

In the second case, one considers a prime number p and determines, by Berlekamp's algorithm, a polynomial g of degree m satisfying the two properties

i) $g \bmod p$ is irreducible in $\mathbb{F}_p[X]$
ii) $g \bmod p$ divides $f \bmod p$ in $\mathbb{F}_p[X]$.

One then chooses a sufficiently large power of p of the form p^l and 'lifts' $g \bmod p$ to a polynomial $g \bmod p^l$. Then we are looking for a polynomial $h \in \mathbb{Z}[X]$ of degree less than or equal to q such that

i) h is an irreducible factor of f in $\mathbb{Z}[X]$
ii) $h \bmod p^l$ is a multiple of $g \bmod p^l$.

Then we work in the lattice

$$L = \{\phi \in \mathbb{Z}[x] \text{ of degree } q \mid \phi = p^l b + ag, \text{ for } a \text{ and } b \in \mathbb{Z}[X]\}$$

which admits the basis

$$V = \{p^l, p^l X, p^l X^2, \ldots, p^l X^{m-1}, g, gX, gX^2, \ldots, gX^{q-m}\}.$$

We remark that if p^l is sufficiently large as a function of the height of g (which one knows how to bound as a function of the height of f), the short vectors of L will have, in the base b, their first m components equal to zero and will then be small multiples of g.

4.5. Simultaneous Diophantine approximations

The problem to be solved is the following: given an n-tuple $(\alpha_1,\alpha_2,\ldots,\alpha_n)$ of real numbers, one is looking for n integers $(p_1,p_2,\ldots,p_n)$ and an integer q such that the n rational numbers $(p_1/q, p_2/q,\ldots,p_n/q)$ are good approximations of the given numbers.

A non-constructive answer to this question is known, due to Dirichlet and based on the theorem of Minkowski: for each n, for each n-tuple $(\alpha_1,\alpha_2,\ldots,\alpha_n)$, for each pair (ϵ,Q) satisfying $\epsilon>0$ and $Q\geqslant\epsilon^{-n}$, there exist integers $(p_1,p_2,\ldots,p_n)$ and an integer q satisfying

$$0<q\leqslant Q \quad \text{and} \quad |q\alpha_i-p_i|\leqslant\epsilon$$

for all $1\leqslant i\leqslant n$. Lagarias [11] was able to give an approximate but constructive version of this theorem by applying the *LLL* algorithm to the lattice L generated by the rows v_i of the matrix

$$A=\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n & \epsilon/Q \end{pmatrix}.$$

Also there, the first vector of the reduced basis obtained, written in the form

$$v=\sum_{i=1}^{n} p_i' v_i + q' v_{n+1}$$

allows us to construct a good approximation

$$\left(\frac{p'_1}{q'},\frac{p'_2}{q'},\ldots,\frac{p'_n}{q'}\right).$$

One can make precise the constructive theorem obtained with the *LLL* algorithm associated to the usual value of the parameter.

For each n, for each n-tuple $(\alpha_1,\alpha_2,\ldots,\alpha_n)$, for each pair (ϵ,Q) satisfying $\epsilon>0$ and $Q\geqslant 2^{n^2}\epsilon^{-n}$ one can construct integers $(p'_1,p'_2,\ldots,p'_n)$ and an integer q' such that

$$0<q'\leqslant Q \quad \text{and} \quad |q\alpha_i-p_i|\leqslant\epsilon,$$

for all $1\leqslant i\leqslant n$.

4.6. Linear programming on integers

The principal problem is the following:

given a polytope P of $\mathbb{R}^n$, with non-zero volume, determine points with integer coordinates in the interior of this polytope.

In general it is known that this problem is NP-hard. But, there also, one can

search for an algorithm which is polynomial when the dimension n is fixed. This is based on [16] and also described in [17], and uses geometric arguments connected to lattice reduction—the spacing of hyperplanes of the lattice parallel to H_{n-1}—as well as other arguments connected to the ellipsoid method in linear programming—the possibility of wedging a polytope between two co-centric and homothetic ellipsoids.

One begins by considering $\mathbb{P}$ to be an ellipsoid, and then one returns to this case by wedging a polytope between two ellipsoids.

Let f be the linear transformation which transforms P into a unit sphere S. Let L be the image of the lattice $\mathbb{Z}^n$ under f. The problem is then transformed into the following: determine the points of L situated in the interior of S.

One reduces the lattice L by applying the algorithm *LLL*. One thus obtains a reduced base $(b_1, b_2, \ldots, b_n)$. Then we proceed in a recursive manner by using the spacing argument which allows us to bound the number of affine hyperplanes parallel to H_{n-1} which intersect the sphere S.

4.7. Attack of the Merkle-Hellmann system

The Merkle-Hellmann cryptographic system is based on the difficulty of the so-called knapsack problem. Given n non-negative integers α_i—the packages—and an integer M—the sack—find an element $X=(x_i)_{1\leqslant i\leqslant n}$ in $\{0,1\}^n$ which is a solution of the equation

$$\sum_{i=1}^{n} \alpha_i x_i = M.$$

Which packages should one take in order to fill up the sack exactly? This problem is easy if the sequence is superincreasing: $\alpha_i > \Sigma_{j<i} \alpha_j$. One can use it in a cryptographic system whose public key is the system of a_i's: given a message formed by the word X one codes it in M. Then one of two things can happen:

— if the sequence is not superincreasing nobody can decode,
— however if it is, then everybody can decode.

One uses a super-increasing sequence, then one hides its super-increasingness by applying a transformation $a \rightarrow va \bmod u$: the pair (u, v^{-1}) will then be the secret key which makes possible the decoding. However, this system is not secure: Shamir [18] has shown that one could find this key and break the code by using a very short vector of a well-chosen lattice.

4.8. The predictability of the sequence of bits produced by the linear congruence generator

The linear congruence generator is perhaps the most celebrated pseudo-random generator. One chooses a modulus m and a multiplier a, relatively prime to m, and an input x_1; one then considers the sequence (x_i) defined by

$$x_{i+1} = ax_i \bmod m.$$

Stern [23] has shown, improving the results of Frieze [2] that, even if none of the parameters is known, the sequence y_i formed by a 'sufficiently large'

proportion of the most significant bits of the x_is is predictable and hence the generator is not cryptographically secure. One works in the lattices X and Y generated by the vectors

$$u_i = \begin{pmatrix} x_{i+1} - x_i \\ x_{i+2} - x_{i+1} \\ x_{i+3} - x_{i+2} \end{pmatrix} \text{ and } v_i = \begin{pmatrix} y_{i+1} - y_i \\ y_{i+2} - y_{i+1} \\ y_{i+3} - y_{i+2} \end{pmatrix},$$

respectively. Given the first k vectors v_i one finds from algorithm 3.10 a short integer relation between them of the form

$$\sum_{i=1}^{k} \lambda_i v_i = 0.$$

One deduces from this that the vector $\Sigma_{i=1}^{k} \lambda_i u_i$ is such a short vector of the lattice X, that it is equal to zero. In fact, if the lattice is geometrically sufficiently 'regular'—which is almost always the case—it does not have any non-zero vector which is very short. If k is appropriately chosen as a function of the presumed size of the data, one constructs a polynomial $\mathbb{P}$ defined by $\mathbb{P}(t)=\Sigma_{i=1}^{k} \lambda_i t^i$ and satisfying $\mathbb{P}(a)=0 \bmod m$. If one reiterates this construction one obtains a sequence of l polynomials P_j all included in a lattice L of base

$$q_0(t) = m, \quad q_i(t) = t^i - a^i,$$

for $1 \leqslant i \leqslant k$. The lattice L has the number m (to be determined) as determinant. By algorithm 3.9 one can find the determinant $\hat{m}$ of the lattice generated by the P_j's. The integer $\hat{m}$ is a multiple of m which decreases rapidly when l increases; one therefore obtains the value of m and a very probable value of a obtained by searching for a polynomial of degree 1 in the lattice L.

4.9. The study of l-th roots modulo n: breaking the cryptosystem of Okamoto [26]
The general problem is as follows:

> We are given two integers n and $l \geqslant 2$. We are given two points, x_0 and y_0, a neighborhood I of x_0, a neighborhood J of y_0 which contain points x and y, respectively, satisfying $x^l = y \bmod n$. Find such points x and y.

More precisely we want to discover a triple (u_1, u_2, v) of 'small integers' which is a solution of the congruence

$$(u_1 x_0 + u_2)^l = y_0 + v \bmod n.$$

y_0 can be expanded to

$$x_0^l u_1^l + C_l^1 x_1^{l-1} u_2 + \cdots + C_l^i x_0^{l-i} u_1^{l-i} u_2^i + \cdots + C_l^1 x_0 u_1 u_2^{l-1} + u_2^l - v$$

modulo n. Letting

$$w_i = u_2^i u_1^{l-i}$$

for $0 \leqslant i \leqslant l-1$ and also

$$w_l = y_0 + v - u_2^l$$

and working in the lattice L of vectors $w = (w_0, w_1, \ldots, w_l)$ in $\mathbb{Z}^{l+1}$ satisfying

$$\sum_{i=0}^{l-1} C_l^i x_0^{l-i} w_i - w_l = 0 \bmod n$$

one searches for a point w of the lattice L which is close—in a sense to be made precise—to the point $(0,0,\ldots,y_0)$. This lattice L of determinant n and rank $l+1$ has the matrix

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ x_0^l & C_l^1 x_0^{l-1} & C_l^2 x_0^{l-2} & & C_l^{l-1} x_0 & n \end{pmatrix}$$

If this lattice is sufficiently 'regular' its first minimum $\Lambda_1(L)$ will be close to the geometric mean of the successive minima, thus of order $n^{2/(l+1)}$. One can show that the majority of lattices of this type are 'regular'; in this case the uniqueness of the nearest point allows us to confirm that the point w found by algorithm 4.2 will give birth to the desired triple (u_1, u_2, v).

4.10. Some other applications

KALTOFEN [8] has used the algorithm in dimension 4 in order to give an algorithm which determines the *gcd* of two numbers in a principal, quadratic but non-euclidean field. Landau and Miller [14] have used the algorithm in order to resolve logarithmically the solvability by radicals of a polynomial equation. Vallée [24] used ideas on lattice basis reduction in the two dimensional case to describe precisely the distribution of elements whose squares mod n are less than $O(\lambda^{2/3})$. She builds an integer factoring algorithm which has the best rigourously established complexity bound for probabilistic integers factoring algorithm.

REFERENCES

1. L. BABAI (1986). On Lovász's lattice reduction and the nearest lattice point problem. *Combinatorica 6*, 1-13.
2. A. FRIEZE, J. HASTAD, R. KANNAN, J.C. LAGARIAS, A. SHAMIR (1988). Reconstructing truncated integer variables satisfying linear congruences. *SIAM Journal on Computing 17*, 262-281.
3. A. DUPRÉ (1846). *Journal de mathématiques 11*, 41-64.
4. C.F. GAUSS (1953). *Recherches Arithmétiques*, Paris 1807, reprinted by Blanchard, Paris.

5. J. HASTAD, B. JUST, J.C. LAGARIAS, C.P. SCHNORR (1986). Polynomial time algorithms for finding integer relations among real numbers. *Proceedings of STACS*, Lecture Notes in Computer Science, to appear in SIAM. J. Comput.
6. B. HELFRICH (1985). Algorithms to construct Minkowski and Hermite reduced bases. *Theoretical Computer Science 41*, 125-139.
7. J.W.S. CASSELS (1978). *Rational Quadratic Forms*, Academic Press.
8. E. KALTOFEN, H. ROLLETSCHEK (1985). Arithmetic in quadratic fields with unique factorization. *Proceedings of EUROCAL'85*, Lecture Notes in Computer Science 204, Springer-Verlag.
9. R. KANNAN (1983). Improved algorithms for integer programming and related lattice problem. *JACM 30*, 193-206.
10. R. KANNAN, H.W. LENSTRA, L. LOVÁSZ (1988). Polynomial factorization and bits of algebraic and some transcendental numbers. *Mathematics of Computation 50*, 235-250.
11. J.C. LAGARIAS (1982). Computational complexity of simultaneous diophantine approximation problem. *23rd IEEE Symp. FOCS.*, 32-39.
12. J.C. LAGARIAS, A. ODLYZKO (1983). Solving low-density subset sum problems. *24th IEEE Symp. FOCS.*, 1-10.
13. J.C. LAGARIAS, H.W. LENSTRA, C.P. SCHNORR (1986). *Korkine-Zolotarev Bases and Successive Minima of a Lattice and Its Reciprocal Lattice*, Technical Report, MSRI 07718-86, Mathematical Sciences Research Institute, Berkeley, to appear in Combinatorica.
14. S. LANDAU, G.L. MILLER (1983). Solvability by radicals is in polynomial time. *15th Annual ACM Symposium on Theory of Computing*, 140-151.
15. A.K. LENSTRA, H.W. LENSTRA, L. LOVÁSZ (1982). Factoring polynomials with rational coefficients. *Math. Annalen 261*, 513-534.
16. H.W. LENSTRA (1983). Integer programming with a fixed number of variables. *Mathematics of Operations Research 8*, 538-548.
17. L. LOVÁSZ. *An Algorithmic Theory of Numbers, Graphs and Convexity*, CBMS-NSF, Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, Pensylvania, USA.
18. A. SHAMIR (1982). A polynomial time algorithm for breaking the Merkle-Hellmann cryptosystem. *23th IEEE Symp. FOCS.*
19. A. SCHÖNHAGE (1984). Factorization of univariate integer polynomial by diophantine approximation and by an improved basis reduction algorithm. *Proceedings of the 11th ICALP*, Antwerpen (1984), Lecture Notes in Computer Science 172, Springer Verlag.
20. C.P. SCHNORR (1987). A more efficient algorithm for lattice basis reduction. *Proceedings of the 13th ICALP*, Rennes (1986), Lecture Notes in Computer Science 226, Springer Verlag, to appear in Journal of Algorithms.
21. C.P. SCHNORR (1987). A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science 53*, 201-224.
22. J. STERN (1986). *Lecture Notes*, University of Singapore.

23. J. STERN (1987). Secret linear congruential generators are not cryptographically secure. *28th IEEE Symp. FOCS.*
24. B. VALLÉE (1989). *Generation of Elements with Small Modular Squares and Provably Fast Integer Factoring Algorithms*, to appear in Math. of Comp., preliminary version in 21st Annual ACM Symposium on Theory of Computing (1989), 98-106.
25. B. VALLÉE (1986). *Une Approche Géométrique de la Réduction des Réseaux en Petite Dimension*, Thèse de doctorat de l'Université de Caen (1986), Rapport de recherche 1989-7 de l'équipe A3L de l'Université de Caen, 'Gauss' algorithm revisited', and 'An affine algorithm for minima finding in integer lattices of lower dimensions'.
26. B. VALLÉE, M. GIRAULT, PH. TOFFIN (1988). How to guess *l*-th roots modulo *n* by reducing lattice bases. *Proceedings of AAECC-6*, Roma, Lecture Notes in Computer Science 357, 427-442.
27. P. VAN EMDE BOAS (1981). *Another NP-Complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice*, Rep. MI, UVA 81-104, Amsterdam.
28. P.M. GRUBER, C.G. LEKKERKERKER (1987). *Geometry of Numbers*, North Holland Mathematical Library, 2nd edition.