JOURNAL OF ALGORITHMS 12, 556–572 (1991)

Gauss' Algorithm Revisited

BRIGITTE VALLÉE

Département de Mathématiques, Université de Caen, 14032 Caen Cedex, France

Received September 1989; revised June 1990

Gauss gave an algorithm which reduces integer lattices in the two-dimensional case and finds a basis of a lattice consisting of its two successive minima. We exhibit its worst-case input configuration and then show that this worst-case input configuration generalizes the worst-case input configuration of the centered Euclidean algorithm to dimension two. © 1991 Academic Press, Inc.

INTRODUCTION

The problem of reducing integer lattice bases consists in finding "short" lattice bases from arbitrary given ones. For the dimension $n = 2$, the problem is solved completely by a polynomial-time algorithm due to Gauss [3]. Given a lattice L described by a basis, Gauss' algorithm outputs a basis $(\mathbf{u}, \mathbf{v})$ formed by two successive minima of L: $\mathbf{u}$ is a shortest non-zero vector of L, and $\mathbf{v}$ is a shortest vector among all vectors of L linearly independent from $\mathbf{u}$. Such a basis is the best possible from the Euclidean point of view and is called minimal.

In 1983, Lenstra, Lenstra, and Lovász [13] exhibited a polynomial-time algorithm, called the LLL algorithm[1] which, given a basis of an integer lattice L of rank n, builds a "good" basis of L. This algorithm uses as a subroutine a slightly modified version of Gauss' algorithm. This LLL algorithm has been widely applied and permits the resolution of other varied and essential problems, for example, in the theory of numbers [4, 8], in algebra [6, 12, 13], in integer programming [5, 14], or in cryptography [10, 17].

[1]The LLL algorithm is often also called the L^3-algorithm or Lovasz' basis reduction algorithm.

0196-6774/91 $3.00

Gauss' algorithm is fundamental for reduction theory of lattices and quadratic forms. It deserves study because insights in the behaviour of Gauss' algorithm may give rise to insights in the behaviour of the LLL algorithm, which in turn influences many other computational algorithms.

The polynomial-time complexity of Gauss' algorithm had been established by Lagarias in 1980 [9], but some natural questions still remain open. Here are the main results of this paper:

(A) We give the best possible upper bound for the number of iterations of such an algorithm.

(B) It is intuitive that Gauss' algorithm generalizes Euclid's centered algorithm to the two-dimensional case. Here, we establish clearly the link between the two algorithms and we deduce that the worst-case input configuration of Gauss' algorithm generalizes the worst-case input configuration of the centered Euclid's algorithm first exhibited by Dupré in 1846 [2].

In addition, we consider a further question concerning the LLL algorithm. The LLL algorithm uses a modified version of Gauss' algorithm that depends on a parameter t, called the Gauss(t) algorithm below. The LLL algorithm is proved to be a polynomial time algorithm when the parameter t is chosen larger than 1. The question is: Is the LLL algorithm remaining a polynomial-time algorithm with the parameter t chosen equal to 1? The choice $t = 1$ corresponds to using the original Gauss' algorithm as a subroutine in the LLL algorithm. We do not answer this question here, but put forward some arguments that bring plausibility to the conjecture that the LLL algorithm has polynomial-time complexity when the parameter t is equal to 1.

Notations. We consider the space $\mathbf{R}^2$ endowed with its Euclidean structure. For $\mathbf{u}$ and $\mathbf{v}$ in $\mathbf{R}^2$, $\mathbf{u} \cdot \mathbf{v}$ is the dot product of $\mathbf{u}$ and $\mathbf{v}$; $|\mathbf{v}|$ is the length of $\mathbf{v}$: $|\mathbf{v}|^2 = \mathbf{v}^2$.

For an element r of $\mathbf{Q}$, we define the integer nearest to r to be the unique integer m, such that $r - m$ belongs to $]-\frac{1}{2}, \frac{1}{2}]$, and the sign of r to be equal to ± 1 and equal to 1 if r is non-negative.

A lattice L of $\mathbf{Z}^2$ is a $\mathbf{Z}$-module of rank 2; it is usually given by a basis $(\mathbf{u}, \mathbf{v})$. Some quantities may be used to measure a basis; the length M of basis $(\mathbf{u}, \mathbf{v})$ is equal to $M = \max\{|\mathbf{u}|, |\mathbf{v}|\}$ but we use mainly the *inertia* l of such a basis. It is the sum of the squares of the lengths of the two vectors of this basis; so one has

$$l = \mathbf{u}^2 + \mathbf{v}^2 \geq M^2.$$

1. An Historical Survey

Here, we follow the very nice book of Scharlau and Opolka [15] which describes the historical evolution of the notion of reduction of integral quadratic forms.

Lagrange [11] was the first mathematician to study the reduction theory of binary quadratic forms, even though these words do not actually appear in his work. The connection between bases of lattices and quadratic forms was observed by Gauss [3] in the two-dimensional case and systematically exploited by Dirichlet [1], who extended the notion of reduction to the case of dimension three. It is quite clear that, among these mathematicians, Gauss was the most interested in what we call now an algorithmical point of view, and this explains why his name is given to the algorithm of reduction of binary quadratic forms.

2. Different Formulations of Gauss' Algorithm

Gauss himself described his algorithm within the general framework of binary integer quadratic forms [3]. Here, it is described in the terms of lattice basis reduction.

We start with a basis $(\mathbf{u}, \mathbf{v})$ of a lattice L in $\mathbf{Z}^2$. If $\mathbf{u}$ is the shortest vector of the basis $(\mathbf{u}, \mathbf{v})$, we replace the vector $\mathbf{v}$ by the smallest vector $\chi(\mathbf{v}, \mathbf{u})$ of

$$K(\mathbf{v}, \mathbf{u}) = \{\mathbf{w} | \mathbf{w} = \varepsilon(\mathbf{v} - m\mathbf{u}),\ m \in \mathbf{Z},\ \varepsilon = \pm 1\}$$

that makes an acute angle with $\mathbf{u}$.

Note that $\chi(\mathbf{v}, \mathbf{u})$ is easy to calculate from $r = \mathbf{u} \cdot \mathbf{v}/\mathbf{u}^2$: the integer m is the integer nearest to r and ε is the sign of $r - m$.

Gauss Algorithm.

Repeat

1. If $\mathbf{u}^2 > \mathbf{v}^2$, exchange $\mathbf{u}$ and $\mathbf{v}$;
2. $\mathbf{v} := \chi(\mathbf{v}, \mathbf{u})$;

until $\mathbf{u}^2 \leq \mathbf{v}^2$.

We recover the original formulation of Gauss if we transfer the operations on the *Gram matrix* $G(\mathbf{u}, \mathbf{v})$, which is the matrix of the associated binary quadratic form. By definition, we have

$$G(\mathbf{u}, \mathbf{v}) = \begin{pmatrix} \mathbf{u}^2 & \mathbf{u} \cdot \mathbf{v} \\ \mathbf{u} \cdot \mathbf{v} & \mathbf{v}^2 \end{pmatrix}.$$

The following result is well known and describes the output configuration:

PROPOSITION 1. *Given a basis* $(\mathbf{u}, \mathbf{v})$ *of a lattice* L, *the Gauss algorithm constructs a basis of* L *that contains two successive minima of* L.

Proof. The output configuration $(\mathbf{u}, \mathbf{v})$ satisfies the two conditions

$$\mathbf{v}^2 \geq \mathbf{u}^2 \qquad \text{and} \qquad 0 \leq \mathbf{u} \cdot \mathbf{v} \leq \left(\tfrac{1}{2}\right)\mathbf{u}^2,$$

so that the length of the projection of $\mathbf{v}$ orthogonally to $\mathbf{u}$ is greater than $(\sqrt{3}/2)|\mathbf{u}|$. We deduce that all the vectors $\mathbf{w}$ of L which are neither parallel to $\mathbf{u}$ nor elements of $K(\mathbf{v}, \mathbf{u})$ satisfy

$$|\mathbf{w}| \geq \sqrt{3}|\mathbf{v}| > |\mathbf{v}| \geq |\mathbf{u}|.$$

Thus $(\mathbf{u}, \mathbf{v})$ is a minimal basis of L. □

In order to analyze the complexity of the Gauss algorithm, we describe now another algorithm that depends on a parameter t, which will be assumed to be strictly greater than 1. The new algorithm is called the *Gauss*(t) *algorithm* and the previous one arises when t is equal to 1. In Gauss(t), the loop termination condition

$$\mathbf{u}^2 \leq \mathbf{v}^2$$

is replaced by

$$\mathbf{u}^2 \leq t^2\mathbf{v}^2. \tag{1}$$

The polynomial complexity of the Gauss(t) algorithm is clear. At each loop, the length of the longer vector is decreased by a factor at least equal to $1/t$. So, we obtain an upper bound for the number k_t of iterations of this algorithm executed on an integer basis of length M and inertia l:

$$k_t \leq \log_t(M) + 1 \leq \tfrac{1}{2}\log_t(l) + 1. \tag{2}$$

We also consider a third version of Gauss' algorithm, in which the loop termination condition is replaced by

$$\mathbf{u} \cdot \mathbf{v} < \mathbf{v}^2. \tag{1bis}$$

This algorithm is called the *Gauss-acute algorithm*. This name is based on a geometrical meaning of this loop termination condition. Let us consider the situation at the end of an iteration of Gauss-acute. If the three points A, B, C are defined in the following way

$$\mathbf{u} = \mathbf{AB}, \qquad \mathbf{v} = \mathbf{AC}, \tag{3}$$

the condition (1bis) expresses that the angle at point C is acute. On the other side, Step 2 has made the two other angles at A and B acute. Hence, Gauss-acute halts when the output triangle ABC built on vectors $\mathbf{u}$ and $\mathbf{v}$ has all three of its angles acute.

3. Comparison between These Algorithms

The following result links the numbers of iterations k, $\hat{k}$, and k_t of the three algorithms—Gauss, Gauss-acute, and Gauss(t)—executed on the same basis and describes the output configurations of these algorithms.

Proposition 2. *For any $t \leq \sqrt{3}$, the two numbers k_t and k satisfy*

$$k_t \leq k \leq k_t + 1. \tag{4}$$

The two numbers k and $\hat{k}$ satisfy

$$\hat{k} \leq k \leq \hat{k} + 1. \tag{4bis}$$

The triangle built on the output configuration of Gauss(t) or on the output configuration of Gauss-acute contains two successive minima of the lattice.

Proof. The inequality $k_t \leq k$ is clear. Let us prove the inequality $\hat{k} \leq k$, by considering the end of an iteration of the algorithms to be compared.

Here, we have $0 \leq (\mathbf{v} \cdot \mathbf{u})/\mathbf{u}^2 \leq \frac{1}{2}$, so that the ratio between the two quantities to be tested,

$$\frac{\mathbf{v} \cdot \mathbf{u}}{\mathbf{v}^2} \text{ on one side,} \qquad \frac{\mathbf{u}^2}{\mathbf{v}^2} \text{ on the other side,}$$

is at most equal to $\frac{1}{2}$. So the loop termination condition of the original Gauss algorithm is sharper than condition (1bis).

We now prove the two other inequalities. We consider the last loop of the Gauss(t) algorithm or the last loop of the Gauss-acute algorithm and we suppose that this loop is not the last one for the Gauss algorithm. We prove that the Gauss algorithm finishes at the next loop. In both cases, we have then

$$|\mathbf{v}| < |\mathbf{u}| \qquad \text{and} \qquad 0 \leq \mathbf{v} \cdot \mathbf{u} \leq \tfrac{1}{2}\mathbf{u}^2.$$

Remark that $\mathbf{v}$ is the shortest side of the triangle built on $(\mathbf{u}, \mathbf{v})$. So the following loop of the Gauss algorithm begins by exchanging $\mathbf{u}$ and $\mathbf{v}$ and

we have

$$|\mathbf{u}| < |\mathbf{v}| \qquad \text{and} \qquad 0 \leq \mathbf{v} \cdot \mathbf{u} \leq \tfrac{1}{2}\mathbf{v}^2.$$

Now $\mathbf{u}$ is the shortest side of the triangle built on $(\mathbf{u}, \mathbf{v})$.

In the case of Gauss(t), condition (1) provides the inequality

$$0 \leq \frac{\mathbf{v} \cdot \mathbf{u}}{\mathbf{u}^2} = \left(\frac{\mathbf{v} \cdot \mathbf{u}}{\mathbf{v}^2}\right)\left(\frac{\mathbf{v}^2}{\mathbf{u}^2}\right) \leq \frac{t^2}{2} \leq \frac{3}{2}. \tag{5}$$

In the case of Gauss-acute, we use directly condition (1bis):

$$0 \leq \frac{\mathbf{v} \cdot \mathbf{u}}{\mathbf{u}^2} < 1.$$

So, in both cases, because of (5) or (1bis), following Step 2 will either leave $\mathbf{v}$ fixed or change it into $\pm(\mathbf{u} - \mathbf{v})$, according to the relative size of these two vectors. Since each of these vectors is longer than $\mathbf{u}$, the execution of the Gauss algorithm finishes at this loop, and this shows the two first assertions.

At the end of this last loop, we obtain the two vectors that were the two shortest vectors of the triangle of the end of the previous loop. Consequently, the second assertion follows from Proposition 1. □

We deduce from (2) and (4) a quite natural proof of polynomial complexity of the Gauss algorithm.

COROLLARY 1. *The number of iterations k of the Gauss algorithm executed on a basis of inertia l satisfies*

$$k(l) \leq \tfrac{1}{2} \log_{\sqrt{3}}(l) + 2. \tag{6}$$

We have obtained a polynomial complexity bound for the Gauss algorithm by comparing it to the Gauss($\sqrt{3}$) algorithm. The geometrical meaning of this value $\sqrt{3}$ is not obvious, and it is intuitively clear that the upper bound (6) is not the best one.

Now we analyze precisely the number of iterations of the Gauss-acute algorithm; we prefer this formulation of Gauss' algorithm because of the geometrical meaning of the loop termination condition. We can easily describe the worst-case input configuration of this algorithm and deduce from it a best possible upper bound for the number of iterations of this formulation of Gauss' algorithm. By using Proposition 2, we then deduce similar facts for the original version of Gauss' algorithm.

4. The Possible Antecedents of a Configuration

We say that an ordered basis $(\mathbf{u}, \mathbf{v})$ is *convenient* if it satisfies the following conditions:

$$0 \leq \frac{\mathbf{v} \cdot \mathbf{u}}{\mathbf{u}^2} \leq \frac{1}{2} \qquad \text{and} \qquad \frac{\mathbf{v} \cdot \mathbf{u}}{\mathbf{v}^2} \geq 1.$$

These convenient bases are exactly the bases that we obtain at the end of any non-final iteration of the Gauss-acute algorithm. Note that for a triangle ABC built on a convenient basis by relation (3), C is its obtuse angle and AC is its shortest side (Fig. 1).

Given a convenient basis $(\mathbf{u}, \mathbf{v})$, we wish to describe its possible *antecedents* by the Gauss-acute algorithm, i.e., the bases of the end of previous Step 2 that could give rise to it (cf. Fig. 1).

Lemma 1. *The possible antecedents of a convenient basis* $(\mathbf{u}, \mathbf{v})$ *are all the convenient bases* $(\mathbf{w}', \mathbf{u})$, *where* $\mathbf{w}'$ *is defined by the equality*

$$\mathbf{w}' = m\mathbf{u} + \varepsilon\mathbf{v},$$

with the following conditions on m *and* ε:

$$m \text{ is an integer}, m > 1, \varepsilon = \pm 1, \text{ and } \varepsilon = 1 \text{ if } m = 2 \text{ or if } \mathbf{u} \cdot \mathbf{v} = \tfrac{1}{2}\mathbf{u}^2. \tag{7}$$

Proof. Before execution of this loop, the vector $\mathbf{u}$, which is now the longer vector of the basis, was the shorter one. It remained fixed. The antecedents of vector $\mathbf{v}$ are thus among the vectors $\mathbf{w}'$ defined by

$$\mathbf{w}' = m\mathbf{u} + \varepsilon\mathbf{v} \qquad \text{with } m \in \mathbf{Z} \text{ and } \varepsilon = \pm 1.$$

The possible antecedents of basis $(\mathbf{u}, \mathbf{v})$ can only be bases $(\mathbf{w}', \mathbf{u})$ with

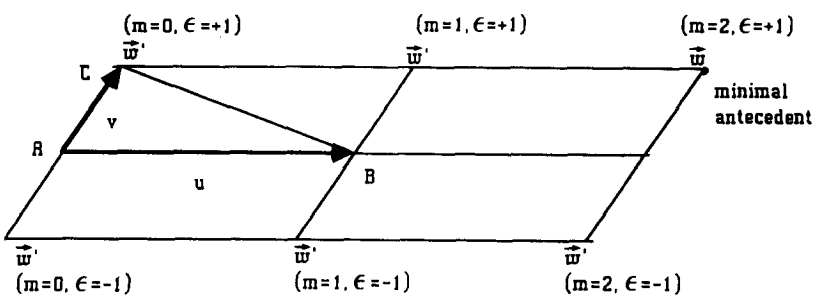

Figure 1

$\mathbf{w}' \cdot \mathbf{u} \geq 0$: this condition excludes the negative values of m and the pair $(m, \varepsilon) = (0, -1)$ (see fig. 1). Moreover, the vector $\mathbf{u}$ must be the shortest side of triangle built on basis $(\mathbf{w}', \mathbf{u})$: this excludes the value $m = 1$ and also the pair $(m, \varepsilon) = (2, -1)$. Finally, if the triangle built on basis $(\mathbf{u}, \mathbf{v})$ is isosceles, i.e.,

$$\mathbf{u} \cdot \mathbf{v} = \left(\tfrac{1}{2}\right)\mathbf{u}^2,$$

all the vectors $\mathbf{w}'$ associated to $\varepsilon = -1$ are excluded because they give, after reduction, the vector $\mathbf{v}'$ symmetrical to $\mathbf{v}$ with respect to the line of vector $\mathbf{u}$. All the other values of the pair (m, ε) lead to convenient bases $(\mathbf{w}', \mathbf{u})$. □

The vector $\mathbf{w}$ defined by the equality

$$\mathbf{w} = 2\mathbf{u} + \mathbf{v}$$

is called the *minimal antecedent* of the basis $(\mathbf{u}, \mathbf{v})$. This definition is justified by the next two lemmas.

For comparing different configurations built on pairs of vectors $(\mathbf{u}, \mathbf{v})$, we use Gram matrices $G(\mathbf{u}, \mathbf{v})$ and we use a partial order on them induced by the order on the coefficients. More precisely, we say that a matrix is *strictly less* than another one if each coefficient of the first one is strictly less than its associate of the second one, except perhaps for the coefficients at the right bottom, where equality is allowed. We also say that a matrix G is *strictly minimal* among a subset $\mathscr{S}$ if G is strictly less than all the other elements of $\mathscr{S}$.

We compare now the Gram matrices built with all the possible antecedents $(\mathbf{w}', \mathbf{u})$ of a convenient basis $(\mathbf{u}, \mathbf{v})$. We return to quadratic forms!

LEMMA 2. *Let* $(\mathbf{u}, \mathbf{v})$ *be a convenient basis. Then the Gram matrix* $G(\mathbf{w}, \mathbf{u})$ *built with its minimal antecedent* $(\mathbf{w}, \mathbf{u})$ *is strictly minimal among all the Gram matrices* $G(\mathbf{w}', \mathbf{u})$ *built with all its possible antecedents* $(\mathbf{w}', \mathbf{u})$.

Proof. Since all points D' defined by $\mathbf{w}' = \mathbf{AD}'$ are at the same distance from the line AB, it is sufficient to show the following inequalities:

$$0 \leq \mathbf{w} \cdot \mathbf{u} < \mathbf{w}' \cdot \mathbf{u} \qquad \text{if } \mathbf{w}' \neq \mathbf{w}.$$

Since ABC is acute at A, one has $\mathbf{v} \cdot \mathbf{u} \geq 0$ and the left inequality is clear. On the other hand, for $\mathbf{w}' \neq \mathbf{w}$, we have to evaluate the sign of the difference

$$\mathbf{w}' \cdot \mathbf{u} - \mathbf{w} \cdot \mathbf{u} = (m - 2)\mathbf{u}^2 + (\varepsilon - 1)\mathbf{u} \cdot \mathbf{v}.$$

This difference is strictly positive under the conditions $\varepsilon = 1$ and $m > 2$.

If $\varepsilon = -1$, the triangle built on $(\mathbf{u}, \mathbf{v})$ cannot be isosceles: we have $2\mathbf{u} \cdot \mathbf{v} < \mathbf{u}^2$ and the difference is also strictly positive under the conditions $\varepsilon = -1$ and $m \geq 3$. □

5. The Worst-Case Configuration of Gauss' Algorithm

We start now with a convenient basis $(\mathbf{u}_0, \mathbf{u}_{-1})$, which is the last convenient basis obtained during an execution of the Gauss-acute algorithm and we consider all the sequences $\mathbf{u}'_n$ made with the nth possible successive antecedents of the basis $(\mathbf{u}_0, \mathbf{u}_{-1})$. Among them, there is a particular one, the sequence $\mathbf{u}_n$ made with the successive minimal antecedents. This sequence follows the linear recurrence (cf. fig. 2)

$$\mathbf{u}_n = 2\mathbf{u}_{n-1} + \mathbf{u}_{n-2} \qquad \text{for } n \geq 1. \tag{8}$$

We compare the Gram matrices G_n and G'_n respectively built with the particular pair $(\mathbf{u}_n, \mathbf{u}_{n-1})$ and all the pairs $(\mathbf{u}'_n, \mathbf{u}'_{n-1})$.

Lemma 3. *The Gram matrix G_n is strictly minimal among all the Gram matrices G'_n.*

Proof. The proof is by induction on n. The assertion is true for $n = 2$, by applying Lemma 2 to basis $(\mathbf{u}_0, \mathbf{u}_{-1})$. For all $n \geq 0$, we consider the matrix U_n built with column vectors $\mathbf{u}_n$ and $\mathbf{u}_{n-1}$ expressed in the canonical basis,

$$U_n = (\mathbf{u}_n \quad \mathbf{u}_{n-1}).$$

The same quantities with a quote denote the associated quantities for the vectors $\mathbf{u}'_n$. We have

$$G_n = {}^tU_nU_n \qquad \text{and} \qquad G'_n = {}^tU'_nU'_n.$$

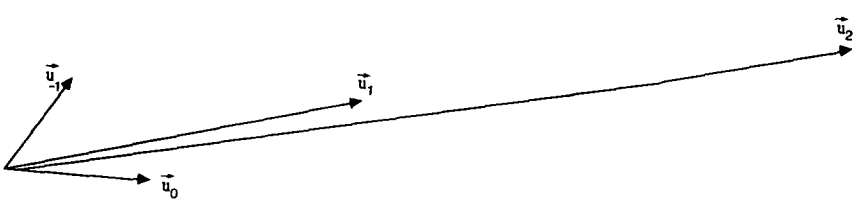

Figure 2

If we let

$$Q = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad Q' = \begin{pmatrix} m & \varepsilon \\ 1 & 0 \end{pmatrix},$$

where the pair (m, ε) satisfies the conditions (7), we have

$${}^tU_n = Q\,{}^tU_{n-1} \quad \text{and} \quad {}^tU'_n = Q'\,{}^tU'_{n-1},$$

so that

$$G_n = QG_{n-1}\,{}^tQ \quad \text{and} \quad G'_n = Q'G'_{n-1}\,{}^tQ'.$$

If we apply now Lemma 2 to possible antecedents of the pair $(\mathbf{u}'_{n-1}, \mathbf{u}'_{n-2})$, we obtain

$$G'_n \geq QG'_{n-1}\,{}^tQ. \tag{9}$$

Since the matrix Q is strictly positive, we can apply the induction hypothesis to obtain

$$QG'_{n-1}\,{}^tQ \geq QG_{n-1}\,{}^tQ.$$

This last term is by definition equal to G_n and, using (9), completes the induction step. Furthermore, we remark that the last two inequalities can become equalities only if the sequence $\mathbf{u}'_n$ is made with all minimal antecedents. □

We consider now the inertiae of these bases. We denote by l_n —resp. l'_n—the inertia of basis $(\mathbf{u}_n, \mathbf{u}_{n-1})$—resp.$(\mathbf{u}'_n, \mathbf{u}'_{n-1})$. These inertiae are in fact traces of the associated quadratic forms: We have

$$l_n = \text{Trace}(G_n) \quad \text{and} \quad l'_n = \text{Trace}(G'_n). \tag{10}$$

We deduce from Lemma 3 that l'_n is greater than l_n and can be equal to l_n only if the sequence $\mathbf{u}'_n$ is made with all minimal antecedents. Now, we calculate l_n exactly.

LEMMA 4. *We have*

$$\frac{l_n}{l_0} \geq \frac{1}{2\sqrt{2}}\left[(1+\sqrt{2})^{2n+1} - (1-\sqrt{2})^{2n+1}\right]. \tag{11}$$

The equality holds only when triangle AB_0B_{-1} defined by $\mathbf{AB}_0 = \mathbf{u}_0$ and $\mathbf{AB}_{-1} = \mathbf{u}_{-1}$ is right-angled at B_{-1}.

Proof. Using $G_n = Q^n G_0 Q^n$ and (10), we link easily l_n to G_0 via matrix Q:

$$l_n = \text{Trace}(Q^n G_0 Q^n).$$

Since *Trace* is invariant by a circular permutation, we obtain

$$l_n = \text{Trace}(Q^{2n} G_0).$$

The symmetrical matrix Q has its eigenvalues equal to $1 + \sqrt{2}$ and $1 - \sqrt{2}$ and can be diagonalized in an orthonormal basis associated to a matrix P, with coefficients in $\mathbf{Q}(\sqrt{2})$. So we have

$$D = {}^tPQP \qquad \text{with } D = \begin{pmatrix} \sqrt{2}+1 & 0 \\ 0 & 1-\sqrt{2} \end{pmatrix},$$

and we deduce

$$l_n = \text{Trace}\left({}^tPQ^{2n}P\,{}^tPG_0P\right) = \text{Trace}\left(D^{2n}\,{}^tPG_0P\right)$$
$$= \gamma(\sqrt{2}+1)^{2n} + \delta(1-\sqrt{2})^{2n},$$

where γ and δ are the diagonal coefficients of matrix ${}^tPG_0P$. Since l_n must be an integer, the coefficients γ and δ are conjugates in $\mathbf{Q}(\sqrt{2})$, so that

$$l_n = (\alpha + \beta\sqrt{2})(\sqrt{2}+1)^{2n} + (\alpha - \beta\sqrt{2})(1-\sqrt{2})^{2n}.$$

If we define the integers α_n and β_n by the relation

$$(\sqrt{2}+1)^n = \alpha_n + \beta_n\sqrt{2}, \tag{12}$$

we obtain

$$l_n = 2(\alpha\alpha_{2n} + 2\beta\beta_{2n}),$$

with initial values $l_0 = 2\alpha$ and $l_1 = 2(3\alpha + 4\beta)$. We evaluate the quantity

$$l_1 - 5l_0 = 5\mathbf{u}_0^2 + \mathbf{u}_{-1}^2 + 4\mathbf{u}_0 \cdot \mathbf{u}_{-1} - 5\mathbf{u}_0^2 - 5\mathbf{u}_{-1}^2 = 4(\mathbf{u}_0 \cdot \mathbf{u}_{-1} - \mathbf{u}_{-1}^2).$$

Since the basis $(\mathbf{u}_0, \mathbf{u}_{-1})$ is convenient, $l_1 - 5l_0$ is always non-negative and is equal to 0 only if the triangle AB_0B_{-1} is right-angled at B_{-1}. Thus

$$8\beta = l_1 - 3l_0 \geq 2l_0 = 4\alpha,$$

and, finally,

$$\frac{l_n}{l_0} = \frac{2(\alpha\alpha_{2n} + 2\beta\beta_{2n})}{\alpha + \beta} \geq \alpha_{2n} + \beta_{2n} = \beta_{2n+1}.$$

Finally, from (12) we obtain the lower bound (11) which is the best possible since it is reached when the first triangle is right-angled. □

We deduce now an upper bound for the number $\hat{k}$ of iterations of the Gauss-acute algorithm.

THEOREM 1. *Let* $(\mathbf{u}, \mathbf{v})$ *be a basis of inertia* l *of an integer lattice* L. *Then the number* $\hat{k}(l)$ *of iterations executed by the Gauss-acute algorithm on basis* $(\mathbf{u}, \mathbf{v})$ *is upper-bounded by* $K(l)$ *with*

$$K(l) = \frac{1}{2}\left[\log_{1+\sqrt{2}}\left(\frac{2\sqrt{2}}{3}l\right) + 1\right]. \tag{13}$$

This upper bound is asymptotically best possible in the following sense: *There exists a sequence of bases of lattice* $\mathbf{Z}^2$ *with strictly increasing inertiae* l *for which the ratio* $\hat{k}(l)/K(l)$ *tends to* 1 *when* l *tends to* ∞.

Proof. Consider a convenient basis $(\mathbf{u}, \mathbf{v})$ of an integer lattice L. Let n be the number of loops executed by the Gauss-acute algorithm on this triangle, while the basis remains convenient. We have $n = \hat{k}(l) - 1$ and, since the inertia l_0 of the last convenient basis $(\mathbf{u}_0, \mathbf{u}_{-1})$ is at least equal to 3, we have

$$\frac{l}{3} \geq \frac{l}{l_0} \geq \frac{l_n}{l_0} \geq \beta_{2n+1} \geq \left(\frac{1}{2\sqrt{2}}\right)(1 + \sqrt{2})^{2n+1}. \tag{14}$$

So, we obtain the announced bound (13).

In the first and third inequality of (14), equality is reached only if triangle AB_0B_{-1} defined in Lemma 4 enjoys the two following properties: it is right-angled at B_{-1} and the inertia $l_0 = \mathbf{AB}_0^2 + \mathbf{AB}_{-1}^2$ is equal to 3. Thus lattice L is the square lattice $\mathbf{Z}^2$, and, after a possible rotation of a right angle, we can choose

$$\mathbf{u}_{-1} = \mathbf{AB}_{-1} = (0, 1) \qquad \text{and} \qquad \mathbf{u}_0 = \mathbf{AB}_0 = (1, 1). \tag{15}$$

In the second inequality of (14), equality is reached only if ABC is the nth minimal antecedent of triangle AB_0B_{-1}. So we obtain the equality

$$\frac{l}{3} = \beta_{2n+1}$$

if and only if, after a possible rotation of a right angle, vectors $\mathbf{u}$ and $\mathbf{v}$ are two successive elements of the sequence $\mathbf{u}_n$ defined by the linear recurrence (8) together with initial conditions (15). More precisely, we must have $\mathbf{u} = \mathbf{u}_n$ and $\mathbf{v} = \mathbf{u}_{n-1}$.

Since β_{2n+1} is equivalent to $(1/2\sqrt{2})(1 + \sqrt{2})^{2n+1}$ for $n \to \infty$, we obtain the stated asymptotic bound. □

In this proof, we also exhibited the worst-case configuration of the Gauss-acute algorithm. The components (x_n, y_n) of vectors $\mathbf{u}_n$ satisfying conditions (8) follow the same linear recurrence as sequences α_n and β_n defined in (12). By comparing initial values defined by (15), we obtain

$$x_n = \beta_{n+1} \qquad \text{and} \qquad y_n = \alpha_{n+1}$$

and we can now describe precisely the worst-case configuration of the Gauss-acute algorithm:

THEOREM 2. *For $k \geq 1$, let $(\mathbf{u}, \mathbf{v})$ be a convenient basis such that the Gauss-acute algorithm requires exactly k loops and such that basis $(\mathbf{u}, \mathbf{v})$ has an inertia I which is minimal among all bases satisfying these conditions. Then, up to a possible rotation of a right angle, the input configuration is defined from sequences α_n and β_n of (12) by*

$$\mathbf{u} = (\beta_k, \alpha_k) \qquad \textit{and} \qquad \mathbf{v} = (\beta_{k-1}, \alpha_{k-1}).$$

6. COMPARING GAUSS' ALGORITHM AND EUCLID'S ALGORITHM

It is clear that Gauss' algorithm generalizes Euclid's algorithm, more precisely the centered Euclidean algorithm. A basic step of the centered Euclidean algorithm between two positive integers a and b $(a \geq b)$ is written

$$a = bq + r \qquad \text{with } -\frac{b}{2} < r \leq \frac{b}{2}$$

and replaces the pair (b, a) by the pair $(|r|, b)$. An iteration of Gauss' algorithm clearly generalizes this basic step to the two-dimensional case.

We show now that, in $\mathbf{Z}^2$, the two algorithms, Gauss' algorithm and the centered continued fraction algorithm, are the same. This fact is deduced from a more general result.

PROPOSITION 3. *Let* $(\mathbf{w}, \mathbf{w}')$ *and* $(\mathbf{u}, \mathbf{v})$ *be two bases of the same lattice* L. *Suppose that the two following conditions hold*:

(i) *the two vectors* $\mathbf{u}$ *and* $\mathbf{v}$ *form an acute angle*,

(ii) *the two vectors* $\mathbf{w}$ *and* $\mathbf{w}'$ *can be expressed in the basis* $(\mathbf{u}, \mathbf{v})$

$$\mathbf{w} = p\mathbf{u} + q\mathbf{v} \qquad \text{and} \qquad \mathbf{w}' = p'\mathbf{u} + q'\mathbf{v},$$

with strictly positive integers p, p', q, q' *satisfying* $q < q'$.

Then, a step of Gauss' algorithm, executed on the pair $(\mathbf{w}, \mathbf{w}')$, *replaces the vector* $\mathbf{w}'$ *by the vector* $\mathbf{w}_0 = p_0\mathbf{u} + q_0\mathbf{v}$ *built from the rational* p_0/q_0 *that is the last convergent of the centered continued fraction expansion of* p/q.

Proof. Since $(\mathbf{u}, \mathbf{v})$ and $(\mathbf{w}, \mathbf{w}')$ are two bases of the same lattice L, we have

$$pq' - p'q = \pm 1. \tag{16}$$

From this equality and hypothesis $q \leq q'$, we deduce that $\mathbf{w}'$ is the longer vector between the two vectors $\mathbf{w}$ and $\mathbf{w}'$. Thus, a step of the Gauss algorithm working on the pair $(\mathbf{w}, \mathbf{w}')$ replaces $\mathbf{w}'$ by another vector—a shorter one.

We consider the last convergent p_0/q_0 of p/q. Then, the pair (p_0, q_0) satisfies the conditions

$$pq_0 - p_0q = \pm 1, \tag{17}$$

$$0 \leq p_0 \leq \frac{p}{2}, \qquad 0 \leq q_0 \leq \frac{q}{2}. \tag{18}$$

Comparing the two equalities (16) and (17), we deduce that there exists an integer m of $\mathbf{Z}$ such that

$$p' = \varepsilon p_0 + mp,\ q' = \varepsilon q_0 + mq \qquad \text{with } \varepsilon = \pm 1.$$

Therefore, $\mathbf{w}_0$ is an element of $K(\mathbf{w}, \mathbf{w}')$. Using (18) and the fact that $\mathbf{u}$ and $\mathbf{v}$ form an acute angle, we deduce that the scalar product $\mathbf{w}_0 \cdot \mathbf{w}$, equal to

$$p_0p\mathbf{u}^2 + q_0q\mathbf{v}^2 + (p_0q + pq_0)\mathbf{u} \cdot \mathbf{v},$$

satisfies

$$0 \leq \mathbf{w}_0 \cdot \mathbf{w} \leq \tfrac{1}{2}\mathbf{w}^2.$$

Thus the vector $\mathbf{w}_0$ is just the vector built by the Gauss algorithm from the pair $(\mathbf{w}, \mathbf{w}')$. □

We can apply this proposition in the particular case when the basis $(\mathbf{u}, \mathbf{v})$ is made with two sides of an acute basic triangle of lattice L. Then, after a possible modification of the sign of the two vectors $\mathbf{u}, \mathbf{v}$, we can assume that the hypotheses of Proposition 3 are fulfilled. So, we prove that the whole Gauss algorithm working on basis $(\mathbf{w}', \mathbf{w})$ coincides with the centered Euclidean algorithm working on the coordinates of $\mathbf{w}$ in this basis $(\mathbf{u}, \mathbf{v})$.

It is usual to assert that the Gauss algorithm and the centered Euclidean algorithm are the same; but it is true only a posteriori, when one already knows the minimal basis of the lattice. And, generally speaking, the Euclidean algorithm is unable to find such a basis while the Gauss algorithm is built for finding it!

But there is a particular case when one knows a priori the minimal basis of the lattice; this is the case of lattice $\mathbf{Z}^2$, where this basis is the canonical one. In this case, it is actually true that the two algorithms coincide.

The worst-case of Gauss' algorithm arises precisely in this case. So, our proof enables us, first, to recover the description of the worst-case configuration of the centered Euclidean algorithm, which was first given by Dupré [2] and, second, to show that the generalization of Euclid's centered algorithm to Gauss' algorithm does not worsen the worst-case.

COROLLARY 2. *For $k \geq 1$, let u and v be integers with $v > u > 0$ such that the centered Euclidean algorithm requires exactly k division steps and such that v is as small as possible satisfying these conditions. Then $v = \beta_{k+1}$ and $u = \beta_k$.*

7. THE COMPLEXITY OF THE LLL ALGORITHM

We recall that the LLL algorithm [13] reduces bases of integer lattices: when given a basis $b = (b_1, b_2, \ldots, b_n)$ of a lattice L, it builds a new basis of this lattice which enjoys good Euclidean properties. This algorithm considers the box B_i which is made with the projections of b_i and b_{i+1} orthogonally to the subspace generated by the first $i - 1$ vectors of basis b and aims at building a new basis b for which all associated boxes B_i are reduced in the sense of the Gauss(t) algorithm.

From both the points of view of complexity and output configuration, it seems that this algorithm depends on the parameter t, which is assumed, as previously, to be strictly greater than 1. During its execution, it performs steps of the Gauss(t) algorithm on boxes B_i, and the total number of steps $s(t)$ of Gauss(t) performed can be upper-bounded as a function of

dimension n and of the length M of the input basis; one obtains

$$s(t) \leq \tfrac{1}{2}n(n-1)\log_t(M)$$

by considering the geometrical decrease (by a factor of $1/t^2$) of the quantity

$$D(b) = \prod_{i=1}^{n-1} D_i(b),$$

where $D_i(b)$ is the determinant of the Gram matrix built with the first i vectors $(b_1, \ldots, b_i)$ of basis b.

This argument fails to prove the polynomial-time complexity of the LLL algorithm for the value $t = 1$ of the parameter t. However, we must remark that this argument already fails when applied in the two-dimensional case and does not prove the polynomial-time complexity of the Gauss algorithm itself!

In Section 2, we have proved the polynomial complexity of the Gauss algorithm in an indirect way: we first established the polynomial-time complexity of the Gauss(t) algorithm; then we proved that the number of iterations of the Gauss(t) is almost independent from the parameter t. For this, we compared the output configurations of both algorithms Gauss(t) and Gauss.

We think that the same indirect method can help to study the complexity of the LLL algorithm for $t = 1$. One needs to answer the question:

> *Does the output configuration of the* LLL *algorithm associated to parameter t essentially depend on t?*

We remark that Lagarias and Odlyzko [10] performed extensive computations and conjectured that the LLL algorithm runs in polynomial-time even for the particular value $t = 1$ of the parameter.

Acknowledgments

This work is a part of my Ph.D. thesis [16]. I thank Jacques Stern, who was my thesis advisor, for many helpful discussions. Many thanks to a referee who gave me many hints to improve the paper.

References

1. G. L. Dirichlet, *J. Reine Angew Math.* **40** (1850), 216–219.
2. A. Dupré, Sur le nombre des divisions à effectuer pour obtenir le plus grand commun diviseur entre deux nombres entiers, *J. Math.* **11** (1846), 41–64, quoted in [7].

3. C. F. GAUSS, Recherches arithmétiques, French translation of "Disquisitiones Arithmeticae," Blanchard, Paris, 1953.
4. B. JUST, Integer relations among algebraic numbers, *in* "Proceedings, MFCS'89," Lectures Notes in Computer Science, Vol. 379, pp. 314–320, Springer-Verlag, New York/Berlin, 1989.
5. R. KANNAN, Improved algorithms for integer programming and related lattice problem, *in* "15th Annual ACM Sympos. Theory of Computing, (1983)", pp. 193–206.
6. R. KANNAN, H. W. LENSTRA AND L. LOVÁSZ, Polynomial factorization and bits of algebraic and some transcendental numbers, *Math. Comput.* **50**, 181 (1988), 235–250.
7. D. E. KNUTH, "The Art of Computer Programming," Vol. II, pp. 361, 605, Addison–Wesley, Reading, MA, 1981.
8. J. C. LAGARIAS, Computational complexity of simultaneous diophantine approximation problem, *in* "23rd IEEE Symp. FOCS, 1982."
9. J. C. LAGARIAS, Worst-Case complexity bounds for algorithms in the theory of integral quadratic forms, *J. Algorithms* **1** (1980), 142–186.
10. J. C. LAGARIAS AND A. M. ODLYZKO, Solving low-density subset sum problem, *in* "24th IEEE Symposium FOCS, 1983," pp. 1–10.
11. J. B. LAGRANGE, *in* "Oeuvres III," pp. 695–795.
12. S. LANDAU, Factoring polynomials over algebraic number fields, *SIAM J. Comput.* **14**, No. 1 (1985), 184–195.
13. A. K. LENSTRA, H. W. LENSTRA, AND L. LOVÁSZ, Factoring polynomials with rational coefficients, *Math. Ann.* **261** (1982), 513–534.
14. H. W. LENSTRA, Integer programming with a fixed number of variables, *Math. Oper. Res.* **8**, No. 4 (1983).
15. W. SCHARLAU AND H. OPOLKA, "From Fermat to Minkowski, Lectures on the Theory of Numbers and Its Historical Development," Undergraduate Texts in Mathematics, Springer-Verlag, New York/Berlin, 1985.
16. B. VALLÉE, "Une approche géométrique de la réduction des réseaux en petite dimension," Thèse de Doctorat, Université de Caen, 1986.
17. B. VALLÉE, M. GIRAULT, AND PH. TOFFIN, How to guess lth roots modulo n by reducing lattice bases, *in* "Proceedings of AAECC-6, Roma, 1988," Lecture Notes in Computer Science, Vol. 357, pp. 427–442, Springer-Verlag, New York/Berlin, 1988.