MATHEMATICS OF COMPUTATION
Volume 74, Number 251, Pages 1531–1543
S 0025-5718(05)01729-1
Article electronically published on January 27, 2005

SOLVING QUADRATIC EQUATIONS USING REDUCED UNIMODULAR QUADRATIC FORMS

DENIS SIMON

ABSTRACT. Let Q be an $n \times n$ symmetric matrix with integral entries and with $\det Q \neq 0$, but not necesarily positive definite. We describe a generalized LLL algorithm to reduce this quadratic form. This algorithm either reduces the quadratic form or stops with some isotropic vector. It is proved to run in polynomial time. We also describe an algorithm for the minimization of a ternary quadratic form: when a quadratic equation $q(x, y, z) = 0$ is solvable over $\mathbb{Q}$, a solution can be deduced from another quadratic equation of determinant ± 1. The combination of these algorithms allows us to solve efficiently any general ternary quadratic equation over $\mathbb{Q}$, and this gives a polynomial time algorithm (as soon as the factorization of the determinant of Q is known).

There are various methods in the literature for solving homogeneous quadratic equations $q(x, y, z) = 0$ over $\mathbb{Q}$. Mathematicians seem to be unanimous in saying that the first step consists of reducing to the diagonal case, that is, to Legendre equations of the type $ax^2 + by^2 + cz^2 = 0$. As we will see in Section 4.2, this is a good idea in theory, but disastrous in practice: the determinant of the new equation (which has to be factored) can become extremely large, even if the original one has only a couple of digits. After this classical reduction, the ways differ according to the authors.

The most often described method to solve the Legendre equations is probably the method of Lagrange, which consists of a Fermat descent: we can deduce a solution for this equation from the solution of a similar equation, but with smaller coefficients (see for example [6, Ch. IV, §3] or [9, Ch. IV, §3.3]): the main drawback of this method is the need to factor many large numbers. Although it seems a theoretical necessity to factor the determinant, no other factorization is justified. Cochrane and Mitchell in [2] and Cremona and Rusin in [4] give ways to avoid all unnecessary factorizations. The corresponding algorithms are efficient, but can not be used for solving general quadratic equations without doing the disastrous reduction to the diagonal case.

According to [2], the solutions of $ax^2 + by^2 + cz^2 = 0$ lie in a lattice of covolume $2|abc|$ (defined by congruences modulo a, b and c), and a smallest vector of this lattice will give a solution. If we use an efficient algorithm for finding small vectors in a 3-dimensional lattice (for example LLL as described in [3, §2.6] or the algorithm of Vallée, [10]), it will give us a solution. However, the authors of [2] consider small vectors for a definite quadratic form, which is not the initial quadratic form.

Received by the editor February 14, 2003 and, in revised form, February 26, 2004.
2000 *Mathematics Subject Classification.* Primary 11Y50, 11E20; Secondary 11H55.
Key words and phrases. Quadratic equation, algorithm.

Although it is easy to define (consider only the absolute value of the coefficients!), it is not naturally attached to the problem. I do not see how to generalize it to the nondiagonal case.

The method of [2] is not that far from the one developed by Gauss in [5, sections 272, 274, 294], except that Gauss reduces directly the indefinite quadratic form for the same lattice. As is noted in [1, p. 98], the method of Gauss consists of two steps:

(1) Compute square roots modulo a, b and c, and build another quadratic form with determinant -1.
(2) Reduce and solve this new quadratic form.

However, Cassels ([1, p. 98]) says about it that "Gauss's proof of the existence of $h(x)$ is explicit but not very transparent, which perhaps explains why it is not often reproduced in the literature", where the notation $h(x)$ refers to the quadratic form with determinant -1. Another comment about it is given in [4]: "Without a fast method of carrying out such a reduction, Gauss's methods of solving Legendre's equation are much slower than the method we presented above."

The goal of this paper is to present a fast method for reducing unimodular quadratic forms, which is a generalization of the LLL–algorithm to indefinite quadratic forms, so step 2 of the algorithm of Gauss is now fast. We also give a process of minimization for a solvable quadratic form, which is a generalization of step 1 to a general ternary quadratic form. We prove that the complete algorithm runs in polynomial time (except for the step of factorization of the initial determinant). We give at the end some numerical examples and the corresponding timings.

Notation

$\mathcal{M}_n(\mathbb{Z})$ = set of $n \times n$ matrices with entries in the ring $\mathbb{Z}$.
$GL_n(\mathbb{Z})$ = subset of $\mathcal{M}_n(\mathbb{Z})$ defined by $\det(Q) = \pm 1$.
$Id(n) = n \times n$ identity matrix.
$\mathbb{F}_p$ = finite field with p elements.
$\bar{x}$ = projection of $x \in \mathbb{Z}$ to $\mathbb{Z}/N\mathbb{Z}$, N depending on the context.
$v_p(x) = p$-adic valuation of x.
$\lfloor x \rceil$ = nearest integer to x, so that $-\frac{1}{2} \leqslant x - \lfloor x \rceil < \frac{1}{2}$.

1. Reduction of unimodular quadratic forms of small dimension

1.1. **Reduction of positive definite quadratic forms.** Consider a positive definite quadratic form q over $\mathbb{Z}^n$. We write $\mathbf{x} \cdot \mathbf{y}$ for the underlying scalar product and $\mathbf{x}^2$ for $\mathbf{x} \cdot \mathbf{x}$. Let $Q = (\mathbf{b}_i \cdot \mathbf{b}_j) \in \mathcal{M}_n(\mathbb{R})$ be its symmetric Gram matrix according to a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$. We have $\det(Q) \neq 0$ and $q(\mathbf{x}) = X^t Q X$, where X contains the coefficients of $\mathbf{x} \in \mathbb{Z}^n$ in the basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$.

For a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of $\mathbb{Z}^n$ and a positive definite quadratic form q, the following algorithms are classical (see Algorithms 2.5.4 and 2.6.3 in [3]).

Algorithm 1.1 (Gram-Schmidt)**.** *Starting with a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of $\mathbb{R}^n$, this algorithm computes an orthogonal basis $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ where $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$.*

`For` $i = 1, \ldots, n$ `do`
`--` `set` $\mathbf{b}_i^* = \mathbf{b}_i$.
`--` `for` $j = 1, \ldots, i-1$, `set` $\mu_{i,j} = \mathbf{b}_i \cdot \mathbf{b}_j^* / \mathbf{b}_j^* \cdot \mathbf{b}_j^*$ `and` $\mathbf{b}_i^* = \mathbf{b}_i^* - \mu_{i,j} \mathbf{b}_j^*$.

Note that we have $\prod_{i=1}^{n}(\mathbf{b}_i^*)^2 = \det(Q)$. If we exchange $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$, the vector $\mathbf{b}_{k-1}^*$ also changes and we have

$$\mathbf{b}_{k-1}^{*\ new} = \mathbf{b}_k^{*\ old} + \mu_{k,k-1}^{old}\mathbf{b}_{k-1}^{*\ old}$$

and

$$\left(\mathbf{b}_{k-1}^{*\ new}\right)^2 = \left(\mathbf{b}_k^{*\ old}\right)^2 + \left(\mu_{k,k-1}^{old}\right)^2\left(\mathbf{b}_{k-1}^{*\ old}\right)^2 .$$

Algorithm 1.2 (LLL). *Let $\frac{1}{4} < c < 1$. Starting with a basis $\mathbf{b}_1, \dots, \mathbf{b}_n$ of $\mathbb{Z}^n$, do the following transformations:*

1- Set $k = 2$.

2- Compute the $\mathbf{b}_i^*$ and the $\mu_{i,j}$ using Algorithm 1.1.

3- For $i = n, \dots, 1$, for $j = 1, \dots, i-1$ set $q = \lfloor \mu_{i,j} \rceil$, $\mathbf{b}_i = \mathbf{b}_i - q\mathbf{b}_j$ and $\mu_{i,j} = \mu_{i,j} - q$.

4- If $(\mathbf{b}_k^*)^2 + \mu_{k,k-1}^2\left(\mathbf{b}_{k-1}^*\right)^2 < c\left(\mathbf{b}_{k-1}^*\right)^2$, exchange $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$, and set $k = \max(k-1, 2)$. Otherwise, set $k = k+1$.

5- If $k \leqslant n$, go to step 2; otherwise, return the basis $(\mathbf{b}_i)$.

This version of the algorithm is absolutely not optimized, but it makes the proofs easier. It is also possible to work directly on the Gram matrix Q and not on the vectors.

Let $\gamma = \frac{1}{c - \frac{1}{4}} > \frac{4}{3}$. It is known (see [3, section 2.6]) that the result of Algorithm 1.2 terminates in polynomial time and has the following properties:

$$(\mathbf{b}_{k-1}^*)^2 < \gamma(\mathbf{b}_k^*)^2 \quad \text{for } 1 < k \leqslant n$$

and

$$(\mathbf{b}_1^2)^n \leqslant \gamma^{n(n-1)/2}\det(Q) .$$

1.2. **Reduction of indefinite quadratic forms.** We consider here a situation close to the situation of Section 1.1. We now allow q to be indefinite, but we restrict to $Q \in \mathcal{M}_n(\mathbb{Z})$. With the same notation, note that $\mathbf{x}^2$ may be nonpositive for $\mathbf{x} \neq 0$.

What happens if we apply Algorithm 1.2 for this quadratic form? As it stands, it is not clear that it finishes. However, we can replace the test in step 4 by the same one with absolute values (this is natural since the quantities involved do not need to be positive any more, and we want to make $(\mathbf{b}_{k-1}^*)^2$ decrease only in absolute value):

Algorithm 1.3 (LLL for indefinite quadratic forms). *Let $\frac{1}{4} < c < 1$. Starting with a basis $\mathbf{b}_1, \dots, \mathbf{b}_n$ of $\mathbb{Z}^n$, do the following transformations:*

1- Set $k = 2$.

2- Compute the $\mathbf{b}_i^*$ and the $\mu_{i,j}$ using Algorithm 1.1 (if this algorithm finds some $(\mathbf{b}_i^*)^2 = 0$, then return $\mathbf{b}_i^*$).

3- For $i = n, \dots, 1$, for $j = 1, \dots, i-1$ set $q = \lfloor \mu_{i,j} \rceil$, $\mathbf{b}_i = \mathbf{b}_i - q\mathbf{b}_j$ and $\mu_{i,j} = \mu_{i,j} - q$.

4- If $|(\mathbf{b}_k^*)^2 + \mu_{k,k-1}^2\left(\mathbf{b}_{k-1}^*\right)^2| < c|\left(\mathbf{b}_{k-1}^*\right)^2|$, exchange $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$ and set $k = \max(k-1, 2)$. Otherwise, set $k = k+1$.

5- If $k \leqslant n$, go to step 2; otherwise, return the basis $(\mathbf{b}_i)$.

Two situations may occur: either one of the $\mathbf{b}_i^*$ satisfies $(\mathbf{b}_i^*)^2 = 0$ at step 2 (during the execution of Algorithm 1.1) and the algorithm stops with a solution of $q(\mathbf{x}) = 0$, or this never happens and the algorithm finishes with a reduced basis.

Theorem 1.4. *Let q be a quadratic form over $\mathbb{Z}^n$ defined by $q(\mathbf{x}) = \mathbf{x}^t Q\mathbf{x}$ with a symmetric matrix $Q \in \mathcal{M}_n(\mathbb{Z})$ such that $\det(Q) \neq 0$. Apply Algorithm 1.3 with $\frac{1}{4} < c < 1$ to a basis $\mathbf{b}_1, \dots, \mathbf{b}_n$ of $\mathbb{Z}^n$. Then*

– either it finds some $\mathbf{x} \in \mathbb{Z}^n$ such that $q(\mathbf{x}) = 0$,

– or it finishes (after a polynomial number of steps) with a reduced basis $\mathbf{b}_1, \dots, \mathbf{b}_n$ such that

$$|(\mathbf{b}_{k-1}^*)^2| \leqslant \gamma |(\mathbf{b}_k^*)^2| \quad \text{for } 1 < k \leqslant n$$

and

$$1 \leqslant |(\mathbf{b}_1)^2|^n \leqslant \gamma^{n(n-1)/2} |\det(Q)| ,$$

where $\gamma = \left(c - \frac{1}{4}\right)^{-1} > \frac{4}{3}$.

If furthermore q is indefinite, we have

$$1 \leqslant |(\mathbf{b}_1)^2|^n \leqslant \frac{3}{4} \gamma^{n(n-1)/2} |\det(Q)| .$$

Proof. We only have to consider the case where it never finds a solution of $q(\mathbf{x}) = 0$, so that all the steps of the algorithm are well defined. We shall first show why this algorithm finishes after a polynomial number of steps. The reason is exactly the same as for the usual LLL, and we reproduce the proofs given in [3, §2.6]. Set $B_i = |(\mathbf{b}_i^*)^2|$. We observe that $d_k = \prod_{i=1}^k B_i$ is the determinant (up to sign) of the minor of Q defined by its k first rows and columns; hence d_k is an integer. Each time we make an exchange at step 4, d_{k-1} strictly diminishes by a factor at least $c^{-1} > 1$, whereas the other d_j do not change. This proves that the algorithm terminates polynomially.

It is clear that at the end, we have $|\left(\mathbf{b}_k^*\right)^2 + \mu_{k,k-1}^2 \left(\mathbf{b}_{k-1}^*\right)^2| \geqslant c |\left(\mathbf{b}_{k-1}^*\right)^2|$ for all $1 < k \leqslant n$. Assume first that $\left(\mathbf{b}_k^*\right)^2$ and $\left(\mathbf{b}_{k-1}^*\right)^2$ have the same sign. We have in this case $|\left(\mathbf{b}_k^*\right)^2| \geqslant (c - \mu_{k,k-1}^2) |\left(\mathbf{b}_{k-1}^*\right)^2|$, but $|\mu_{k,k-1}| \leqslant \frac{1}{2}$; hence $B_{k-1} \leqslant \gamma B_k$. Assume now that $\left(\mathbf{b}_k^*\right)^2$ and $\left(\mathbf{b}_{k-1}^*\right)^2$ have opposite signs. In this case $\left(\mathbf{b}_k^*\right)^2 + \mu_{k,k-1}^2 \left(\mathbf{b}_{k-1}^*\right)^2$ must have the same sign as $\left(\mathbf{b}_k^*\right)^2$, and we have $|\left(\mathbf{b}_k^*\right)^2| \geqslant (c + \mu_{k,k-1}^2) |\left(\mathbf{b}_{k-1}^*\right)^2| \geqslant c |\left(\mathbf{b}_{k-1}^*\right)^2|$. Hence $B_{k-1} \leqslant c^{-1} B_k \leqslant \gamma B_k$.

It remains to prove the last inequality. Since we have assumed that the algorithm does not find any solution of $q(\mathbf{x}) = 0$, the integer $|(\mathbf{b}_1)^2|$ must be at least 1. We have $|(\mathbf{b}_1)^2| = B_1 \leqslant \gamma B_2 \leqslant \cdots \leqslant \gamma^{n-1} B_n$. The product of these inequalities gives $|(\mathbf{b}_1)^2|^n \leqslant \gamma^{1+2+\cdots+(n-1)} d_n$. Recall that $d_n = |\det(Q)|$, and we get the result for a general q.

Consider now the particular case of q indefinite. Since the quadratic form is indefinite, the sequence $(\mathbf{b}_1^*)^2$,$(\mathbf{b}_2^*)^2$,...,$(\mathbf{b}_n^*)^2$ contains a sign change. Assume for example $(\mathbf{b}_{k-1}^*)^2 (\mathbf{b}_k^*)^2 < 0$, with $1 < k \leqslant n$. As we have seen earlier, we have $B_{k-1} \leqslant c^{-1} B_k$ and $B_{i-1} \leqslant \gamma B_i$ for $i \neq k$. From this slightly better inequality, we get $|(\mathbf{b}_1)^2|^n \leqslant \gamma^{1+2+\cdots+(n-1)-(n+1-k)} c^{-(n+1-k)} d_n$. The difference with the general case is therefore the factor $(\gamma c)^{-(n+1-k)}$. We have $n + 1 - k \geqslant 1$, and the upper bound $c < 1$ gives $(\gamma c)^{-(n+1-k)} \leqslant (\gamma c)^{-1} < \frac{3}{4}$: we have the conclusion. □

Remark 1.5. From this result, we see that the quality of the reduction is better when the quadratic form is indefinite. In fact, the bound for $|(\mathbf{b}_1)^2|$ can be even smaller if the sign change occurs between $(\mathbf{b}_{k-1}^*)^2$ and $(\mathbf{b}_k^*)^2$ for a small k (the worst case being $k = n$) or if there are several such sign changes.

TABLE 1.

n	lower bound
3	$\frac{3}{4} = 0.750 < c$
4	$\frac{1}{4} + 2^{-2/3} < 0.880 < c$
5	$\frac{1}{4} + 2^{-1/2} < 0.958 < c$

1.3. **Solving unimodular quadratic equations of small dimension.**

Theorem 1.6. *Let $n \leqslant 5$ and let q be a unimodular quadratic form over $\mathbb{Z}^n$ defined by $q(\mathbf{x}) = \mathbf{x}^t Q\mathbf{x}$ with a symmetric matrix $Q \in \mathcal{M}_n(\mathbb{Z})$ such that $\det(Q) = \pm 1$. Apply Algorithm 1.3 with $\frac{1}{4} + 2^{-2/(n-1)} < c < 1$, and assume it does not find a solution of $q(\mathbf{x}) = 0$. Then the Gram matrix of the reduced basis is diagonal with only ± 1 coefficients on the diagonal.*

Proof. The lower bound given for c gives us exactly $\gamma^{n(n-1)/2} < 2^n$. Using Theorem 1.4, we see that when the algorithm does not find a solution, it finishes with a reduced basis such that $1 \leqslant |(\mathbf{b}_1)^2| < 2$. Since $(\mathbf{b}_1)^2 \in \mathbb{Z}$, we have $(\mathbf{b}_1)^2 = \pm 1$. Now, for $1 < i \leqslant n$, we have $\mu_{i,1} = \mathbf{b}_i.\mathbf{b}_1^*/\mathbf{b}_1^*.\mathbf{b}_1^*$, but $\mathbf{b}_1 = \mathbf{b}_1^*$ and $\mathbf{b}_1^*.\mathbf{b}_1^* = \pm 1$; hence $\mu_{i,1} = \pm \mathbf{b}_i.\mathbf{b}_1 \in \mathbb{Z}$. At the end of the algorithm, we have $|\mu_{i,1}| \leqslant \frac{1}{2}$, which implies that $\mu_{i,1} = 0$ and that $\mathbf{b}_i.\mathbf{b}_1 = 0$. By an easy induction we obtain the result for the other coefficients of the Gram matrix. □

Remark 1.7. Table 1 shows the computations for the numerical values given in this theorem for the lower bound for c.

Theorem 1.8. *Let $n \leqslant 6$ and let q be a unimodular indefinite quadratic form over $\mathbb{Z}^n$ defined by $q(\mathbf{x}) = \mathbf{x}^t Q\mathbf{x}$ with a symmetric matrix $Q \in \mathcal{M}_n(\mathbb{Z})$ such that $\det(Q) = \pm 1$. Apply Algorithm 1.3 with $\frac{1}{4} + 2^{-2/(n-1)} \left(\frac{3}{4}\right)^{2/(n^2-n)} < c < 1$, and assume it does not find a solution of $q(\mathbf{x}) = 0$. Then the Gram matrix of the reduced basis is diagonal with only ± 1 coefficients on the diagonal.*

Proof. The proof is similar to the proof of Theorem 1.6, using the inequality given in Theorem 1.4 for q indefinite. □

Remark 1.9. The improved lower bounds for c are given numerically in Table 2.

Remark 1.10. We can really see Algorithm 1.3 as an equation solver for unimodular indefinite quadratic equations of dimension $n \leqslant 6$. Since even it does not directly return a solution, we can certainly find such a solution among the $\mathbf{b}_i + \mathbf{b}_j$.

TABLE 2.

n	lower bound
3	$0.705 < c$
4	$0.851 < c$
5	$0.938 < c$
6	$0.994 < c$

2. Minimization of solvable ternary quadratic forms

Let $q(X,Y,Z)$ be a quadratic form, defined by a 3×3 symmetric matrix Q with integral entries. We assume in this section that the determinant of Q is not 0. Our goal is to find a nontrivial integral solution (or rational, but this is clearly equivalent) for $q(X,Y,Z)=0$. The existence of such a solution is classically equivalent to the existence of a solution in each p-adic field $\mathbb{Q}_p$ and in $\mathbb{R}$ (see for example [6]). It is also well known that for an odd prime p not dividing $\det Q$, a solution always exists. This existence is given by several Hilbert symbols which are easy to compute if we know the factorization of $\det Q$. Hence, we shall assume in this section that this factorization is known and that a nontrivial p-adic solution always exists. This last assumption is not very important, since it can be deduced from our algorithm; however, it is not an efficient way to prove (or disprove) it.

The aim of this section is to minimize the solvable ternary quadratic form q. This means that we will build another quadratic form q', equivalent to q, but with determinant ± 1, such that a solution of q can be deduced from a solution of q'. The strategy is to work with one prime divisor p of $\det Q$ at a time and to divide successively the determinant by p, until it is ± 1.

We shall prove

Theorem 2.1. *Let $Q\in\mathcal{M}_3(\mathbb{Z})$ be a symmetric matrix with $\det Q\neq 0$ and such that the quadratic equation $\mathbf{x}^tQ\mathbf{x}=0$ has a nontrivial local solution in $\mathbb{Q}_p$ for all primes p dividing $\det(Q)$. Then, there is a matrix $V\in\mathcal{M}_3(\mathbb{Z})$ with the following properties:*

$$\det(V)=|\det(Q)|,$$
$$Q'=\frac{1}{\det(Q)}V^tQV\in\mathcal{M}_3(\mathbb{Z}),$$
$$\det(Q')=\pm 1\ .$$

As soon as the factorization of $\det(Q)$ is known, there is an algorithm for finding V in at most $O(\ln^4(|\det(Q)|))$ operations. There is a constant $\kappa>0$ such that the coefficients of V are $O(|\det(Q)|^\kappa)$.

We will repeatedly use the over–line to denote the reduction mod p.

2.1. Computing the kernel mod p.

In this section, we want to find the kernel of a matrix mod p and lift the corresponding base change (given by a matrix in $GL_n(\mathbb{F}_p)$) to $GL_n(\mathbb{Z})$. This is done in a single algorithm.

Consider the following algorithm:

In this algorithm, we write $x \mod p$ for the integer $x'\in\mathbb{Z}$ such that $p\mid(x-x')$ and $|x'|\leqslant\frac{p}{2}$ (for $p=2$, we choose x' to be 0 or 1). We also write M_k for the kth column of M.

Algorithm 2.2. *Let p be a prime number and $M\in\mathcal{M}_n(\mathbb{Z})$. This algorithm computes a matrix $U\in GL_n(\mathbb{Z})$ and $d\geqslant 0$:*

1- Set $i=n$, $d=0$ and $U=Id(n)$.

2- Set $j=i+d$. While $j>0$ and $p\mid M_{i,j}$, do $j=j-1$. If $j=0$, set $d=d+1$ and go to 6.

3- If $j<i+d$, exchange M_j and M_{i+d} and exchange U_j and U_{i+d}.

4- Let $u\in\mathbb{Z}$ such that $uM_{i,i+d}\equiv 1 \mod p$.

5- For all $1 \leqslant k \leqslant j-1$ set $\alpha = uM_{i,k} \mod p$, $M_k = M_k - \alpha M_{i+d}$ and $U_k = U_k - \alpha U_{i+d}$. Reduce $M \mod p$.

6- Set $i = i-1$. If $i > 0$, go to 2.

7- Return U and d.

Proposition 2.3. *The result of Algorithm* 2.2 *is such that the kernel of* $\bar{M}$ *(mod* p*) is the span of the first* d *columns of* $\bar{U}$*. The coefficients of* U_{n-k} *are bounded by* $\left(1+\frac{p}{2}\right)^k$ *and by* $\left(1+\frac{p}{2}\right)^{n-d}$*. If the coefficients of* M *are bounded by* $\frac{p}{2}$*, then the algorithm runs in* $O(n^3 \ln^2(p))$ *bit operations.*

Proof. It is standard that this algorithm gives the kernel of $M \mod p$ (see Algorithm 2.3.1 in [3]). The inequalities are easy since $|\alpha| \leqslant \frac{p}{2}$ in step 5. Step 4 requires $O(\ln^2(p))$ bit operations, and step 5 requires $O(n^2 \ln^2(p))$ bit operations, so that the full algorithm runs in $O(n^3 \ln^2(p))$ bit operations. □

2.2. **Minimization.**

Lemma 2.4. *Let* $Q \in M_n(\mathbb{Z})$ *and let* p *be a prime number. Let* $d = \dim_{\mathbb{F}_p}(\ker \bar{Q})$*. We have* p^d *divides* $\det Q$*.*

Proof. This is a corollary of Proposition 2.3. □

Now Q will be a symmetric matrix in $\mathcal{M}_3(\mathbb{Z})$, such that the quadratic form $\mathbf{x}^t Q\mathbf{x}$ is solvable over $\mathbb{Q}_p$.

Proposition 2.5. *Let* p *be a prime such that* $v_p(\det(Q)) = 1$ *and such that the quadratic equation* $\mathbf{x}^t Q\mathbf{x}$ *has a nontrivial solution in* $\mathbb{Q}_p$*. There is a matrix* $V \in \mathcal{M}_3(\mathbb{Z})$ *such that* $\det(V) = p$ *and* $V^t QV = pQ'$ *where* $Q' \in M_3(\mathbb{Z})$ *is symmetric with* $\det(Q') = p^{-1}\det(Q)$*.*

Proof. From Lemma 2.4, we have $\dim_{\mathbb{F}_p}(\ker \bar{Q}) = 1$. Let U be given by Algorithm 2.2 and $Q'' = U^t QU$. Let $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ be the canonical basis of $\mathbb{Z}^n$. We know that $Q''\mathbf{x}_1$ is divisible by p and in particular $Q''_{1,1} = \mathbf{x}_1^t Q''\mathbf{x}_1$ is divisible by p. But this last quantity cannot be divisible by p^2 since $v_p(\det(Q'')) = 1$. Let $\mathbf{x} = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \alpha_3\mathbf{x}_3$ be a nontrivial p-adic solution of $\mathbf{x}^t Q''\mathbf{x} = 0$. After rescaling, we can assume that $\min_{i=1,2,3}(v_p(\alpha_i)) = 0$. We have $\min_{i=2,3}(v_p(\alpha_i)) = 0$. Indeed, if we had $p \nmid \alpha_1$, $p \mid \alpha_2$ and $p \mid \alpha_3$, then we would have $0 = \alpha_1^2 \mathbf{x}_1^t Q''\mathbf{x}_1 + 2\alpha_1(\mathbf{r}^t Q''\mathbf{x}_1) + \mathbf{r}^t Q''\mathbf{r}$ (we have set $\mathbf{r} = \alpha_2\mathbf{x}_2 + \alpha_3\mathbf{x}_3$) with $v_p(\alpha_1^2 \mathbf{x}_1^t Q''\mathbf{x}_1) = 1$, $v_p(2\alpha_1(\mathbf{r}^t Q''\mathbf{x}_1)) \geqslant 2$ and $v_p(\mathbf{r}^t Q''\mathbf{r}) \geqslant 2$. But this is impossible, so we have $\min_{i=2,3}(v_p(\alpha_i)) = 0$. By symmetry, we can assume that $v_p(\alpha_2) = 0$. Let $x \in \mathbb{Z}$ with $x \equiv \alpha_3\alpha_2^{-1} \mod p$. Let $N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & p & x \\ 0 & 0 & 1 \end{pmatrix}$. Set $V = UN$. We have $V \in M_3(\mathbb{Z})$ with $\det(V) = p$, and $V^t QV$ is divisible by p by construction, so we have the conclusion. □

This proposition corresponds to the following algorithm:

Algorithm 2.6. *Assume a symmetric matrix* $Q \in M_3(\mathbb{Z})$ *and a prime number* p *satisfying the conditions of Proposition* 2.5*. This algorithm returns the matrix* $V \in M_3(\mathbb{Z})$ *described in this proposition.*

1- Let U be given by Algorithm 2.2. Set $Q'' = U^t QU$.

2- If $p \mid Q''_{2,2}$, set $N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & p \end{pmatrix}$ and go to 6.

3- Let $u \in \mathbb{Z}$ such that $uQ''_{2,2} \equiv 1 \mod p$.

4- Set $\Delta = (Q''_{3,2})^2 - Q''_{2,2}Q''_{3,3}$. Using Algorithm 1.5.1 of [3], compute a square root δ of $\Delta \mod p$.

5- Set $x = u(-Q''_{2,3} + \delta) \mod p$ and $N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & p & x \\ 0 & 0 & 1 \end{pmatrix}$.

6- Return $V = UN$.

Remark 2.7. During the proof, we have used the local solubility to build a nontrivial solution mod p of the quadratic equation $\mathbf{x}^t Q'' \mathbf{x} = 0$ for $\mathbf{x}$ of the form $\mathbf{x} = a\mathbf{x}_2 + b\mathbf{x}_3$. This algorithm can be used as a test for the local solubility of Q. Indeed, one can prove that the square root involved in step 4 exists if and only if Q has a nontrivial solution in $\mathbb{Q}_p$.

Remark 2.8. At step 4, we have $\Delta \neq 0 \mod p$. For $p = 2$, this implies that Δ is a square and that the algorithm still works, even if we do not assume the existence of a nontrivial solution in $\mathbb{Q}_2$. I thank John Cremona who pointed out this fact.

Remark 2.9. Since computing that a square root mod p is achieved in $O(\ln^4(p))$ bit operations, this algorithm runs in $O(\ln^4(p))$ bit operations. During step 5, if x is chosen in the interval $[-\frac{p}{2}, \frac{p}{2}]$, we deduce from Proposition 2.3 that the coefficients of V are $O(p^3)$.

Proposition 2.10. *Let p be a prime such that $v_p(\det(Q)) \geqslant 2$. Assume further that $\dim_{\mathbb{F}_p}(\ker \bar{Q}) = 1$. There is a matrix $V \in \mathcal{M}_3(\mathbb{Z})$ such that $\det(V) = p^2$ and $V^t Q V = p^2 Q'$ where $Q' \in \mathcal{M}_3(\mathbb{Z})$ is a symmetric matrix with $\det(Q') = p^{-2}\det(Q)$.*

Proof. Let U be given by Algorithm 2.2 and let $Q'' = U^t Q U$. We know that $Q''_{1,1}$ is divisible by p. It is not difficult to prove that in this situation it is in fact divisible by p^2. Let $N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{pmatrix}$. The matrix $V = UN$ has the required property. □

This proposition corresponds to the following very simple algorithm:

Algorithm 2.11. *Assume a symmetric matrix $Q \in M_3(\mathbb{Z})$ and a prime number p satisfying the conditions of Proposition* 2.10. *This algorithm returns the matrix $V \in M_3(\mathbb{Z})$ described in this proposition.*

1- Let U be given by Algorithm 2.2. Set $Q'' = U^t Q U$.

2- Set $N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{pmatrix}$.

3- Return $V = UN$.

Remark 2.12. This algorithm clearly runs in $O(\ln^2(p))$ bit operations. We deduce from Proposition 2.3 that the coefficients of V are $O(p^3)$.

Proposition 2.13. *Let p be a prime such that $v_p(\det(Q)) \geqslant 2$. Assume further that $\dim_{\mathbb{F}_p}(\ker \bar{Q}) \geqslant 2$. There is a matrix $V \in \mathcal{M}_3(\mathbb{Z})$ such that $\det(V) = p$ and $V^t Q V = pQ'$ where $Q' \in \mathcal{M}_3(\mathbb{Z})$ is a symmetric matrix with $\det(Q') = p^{-1}\det(Q)$.*

Proof. Let U be given by Algorithm 2.2 and let $Q'' = U^tQU$. The first two columns and rows of Q'' are divisible by p. Let $N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & p \end{pmatrix}$. The matrix $V = UN$ has the required property. □

This proposition corresponds to the following very simple algorithm:

Algorithm 2.14. *Assume a symmetric matrix $Q \in M_3(\mathbb{Z})$ and a prime number p satisfying the conditions of Proposition* 2.13. *This algorithm returns the matrix $V \in M_3(\mathbb{Z})$ described in this proposition.*

1- Let U be given by Algorithm 2.2. Set $Q'' = U^tQU$.

2- Set $N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & p \end{pmatrix}$.

3- Return $V = UN$.

Remark 2.15. This algorithm clearly runs in $O(\ln^2(p))$ bit operations. We deduce from Proposition 2.3 that the coefficients of V are $O(p^2)$.

Now we get the proof of Theorem 2.1, just by putting together all the previous results and applying these algorithms to each prime p dividing $\det Q$ and by noting (again !) that the factorization of $\det Q$ must be given because we do not know how to compute it nearly so quickly. This algorithm is part of Algorithm 3.1 given subsequently.

Remark 2.16. The constant κ involved in the bound for V is not explicit. In order to get an explicit bound, we should do a careful analysis of the bounds in Algorithms 2.6, 2.11 and 2.14 and be able to derive an explicit bound when we multiply all the different V together. However, this bound for V should not be too far from $O(|\det Q|^3)$ which gives the bound $O(M|\det Q|^5)$ for Q' where M is a bound for the coefficients of Q. In any case, we have $\det Q = O(M^3)$, so that the algorithm for finding V is polynomial time, and Q' is always bounded by a power of M.

3. The complete solution of ternary quadratic equations

3.1. **The general case.** Putting together Theorem 1.8 and Theorem 2.1 and the corresponding algorithms, we get the following algorithm for solving general ternary quadratic equations:

Algorithm 3.1. *Assume a symmetric matrix Q with $\det Q \neq 0$, such that a non-trivial rational solution of $\mathbf{x}^tQ\mathbf{x}$ exists. This algorithm returns one such solution.*

1- Factor $\det Q$. Set $W = Id(3)$.

2- If $\det Q = \pm 1$, go to 6.

3- Let $p \mid \det Q$. Let U and d be given by Algorithm 2.2 applied with $M = Q$.

4- If $v_p(\det Q) = 1$, compute V by Algorithm 2.6. Otherwise, compute V by Algorithm 2.11 if $d = 1$ or by Algorithm 2.14 if $d > 1$.

5- Set $W = WV$ and $Q = \frac{1}{\det V} V^tQV$. Go to 2.

6- Apply Algorithm 1.3 to Q (and to the canonical basis of $\mathbb{Z}^n$) with $c = \frac{3}{4}$. If the answer is a solution of $\mathbf{x}^tQ\mathbf{x} = 0$, return $W\mathbf{x}$.

Otherwise let B be the matrix of the reduced basis, and let $Q'=B^tQB$ (this matrix is diagonal with coefficients ±1). Find a solution of $\mathbf{x}^tQ'\mathbf{x}=0$ and return $WB\mathbf{x}$.

As we have seen, if the coefficients of Q are bounded by M, the size of the minimized matrix $\frac{1}{\det Q}W^tQW$ is bounded by a polynomial in M. The analysis of Algorithm 1.3 shows therefore that the complete algorithm runs in polynomial time as soon as the factorization of the determinant is known.

3.2. **An important special case: Legendre equations.** We specialize in this section to the case where the symmetric matrix $Q \in \mathcal{M}_3(\mathbb{Z})$ is diagonal. The quadratic form is now $q(x,y,z)=ax^2+by^2+cz^2$. The equation $q=0$ is known as a *Legendre equation*. We can assume that a, b, and c are integral and pairwise coprime, and, if we know their factorizations, that they are squarefree. Assume $q=0$ has a local solution in $\mathbb{Q}_p$ for all primes p and in $\mathbb{R}$. It is known that this condition is equivalent to the existence of a solution in $\mathbb{Q}$. In this situation, it is possible to do all the minimization steps described in Section 2.2 for all primes in a single step. The only remaining thing to do for solving $q=0$ is to use Algorithm 1.3.

Theorem 3.2. *Let a, b and c be pairwise coprime integers. Assume that the quadratic equation $ax^2+by^2+cz^2=0$ has a local solution in $\mathbb{Q}_p$ for all primes p dividing abc. Let $Q=\begin{pmatrix}a&0&0\\0&b&0\\0&0&c\end{pmatrix}$. Then, there is a matrix $U\in\mathcal{M}_3(\mathbb{Z})$ with the following properties:*

$$\begin{aligned}&\det(U)=abc,\\ Q'=\frac{1}{abc}U^tQU&\in\mathcal{M}_3(\mathbb{Z}),\\ &\det(Q')=\pm1.\end{aligned}$$

Proof. This theorem is essentially a reformulation of [5, p. 294]. It is also a special case of Theorem 2.1. □

Remark 3.3. As we have seen for Proposition 2.5, the 2-adic solubility is not necessary in this theorem.

The full algorithm for solving the Legendre equation $aX^2+bY^2+cZ^2=0$ corresponds to the following:

Algorithm 3.4. *Given a, b and c, three squarefree and pairwise coprime integers, this algorithm assumes that the Legendre equation has a rational solution and returns one such solution.*

1- For all prime divisors p of a, compute a square root of $-c/b$ modulo p (using the algorithm of Shanks ([3, Alg.1.5.1]), and use the Chinese Remainder Theorem to deduce a square root X_a of $-c/b$ modulo a.

2- Compute also a square root X_b of $-c/a$ modulo b and a square root X_c of $-b/a$ modulo c.

3- Using the Extended Euclid Algorithm, compute two integers u and v such that $bu+cv=1$.

4- Let $U = \begin{pmatrix} bc & abuX_c & X_abuX_c + X_bcv \\ 0 & a & X_a \\ 0 & 0 & 1 \end{pmatrix}$ and $Q' = \frac{1}{abc}U^t \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} U$.

5- Apply Algorithm 1.3 to Q' with $c = \frac{3}{4}$. If the answer is a solution of $\mathbf{x}^tQ'\mathbf{x} = 0$, return $U\mathbf{x}$. Otherwise let B be the matrix of the reduced basis, and let $Q'' = B^tQ'B$ (this matrix is diagonal with coefficients ± 1). Find a solution of $\mathbf{x}^tQ''\mathbf{x} = 0$ and return $UB\mathbf{x}$.

4. Numerical examples

4.1. **Legendre equations.** We test our algorithm on the values given in [4], and we reproduce the notation: the coefficients are of the form $10^k + \varepsilon$. There are 100 test values for each $k \leqslant 200$ and only 5 for $k = 500$. In these examples, the equation is rationally solvable, and the coefficients are known to be primes, a fact which is explicitly used in the algorithm so that no factorization is needed. Our implementation is written in GP and runs with an Athlon 900 MHz processor. In Table 3 the time indicated for the certificate (first column) is the time for the computation of the square roots modulo p (steps 1 and 2 of Algorithm 3.4). The time indicated for the reduction (second column) is the time for the second part of Algorithm 3.4 (steps 3–5), that is, essentially for Algorithm 1.3. The times are expressed in seconds.

4.2. **A huge example from 2-descent on elliptic curves.** In [8] we have seen that the algorithm of 2-descent on elliptic curves over a number field K uses the solution of several quadratic equations over K. It was suggested there to solve them using the standard algorithm, which starts by a diagonalization and then uses either a standard algorithm for Legendre equations if the equation is over $\mathbb{Q}$, or an algorithm for solving norm equations (for example as described in [7]) if we are over a number field. However, if we work over $\mathbb{Q}$, this is not a good idea, since the diagonalization step multiplies the determinant of the equation by its first coefficient, which may be impossible to factorize, compared to the determinant. If we use our new method, the factorization of the determinant is known in advance,

Table 3.

k	Certificate	Reduction
10	0.030	0.630
15	0.030	0.800
20	0.040	0.930
25	0.050	1.030
30	0.090	1.150
50	0.120	1.680
75	0.400	2.270
100	0.800	2.990
125	1.740	3.720
150	1.640	4.470
175	3.700	5.400
200	4.800	6.200
500	3.900	1.000

and no other factorization is needed. Usually, the factorization of the determinant is very easy (typically a few digits), since it is given by the norm of a unit or an S-unit. On the other hand, the coefficients of the quadratic form themselves are impossible to factor (typically hundreds of digits), since they correspond to the coefficients of the same unit or S-unit.

We give here a striking example, occuring for the elliptic curve $y^2 = x^3 + 7823$.

We recall quickly how the quadratic equations are built in the context of 2-descent on elliptic curves (for more details, see [8]). Let $K = \mathbb{Q}(\theta)$ be a cubic field, where θ is a root of $\theta^3 + A\theta^2 + B\theta + C = 0$. Let $\delta = a - b\theta + c\theta^2$ be a unit of K (or an S-unit of K, for some set S containing only a few small primes), such that its norm $\mathcal{N}_{K/\mathbb{Q}}(\delta)$ is a square r^2. We want to find some nonzero $z = u + v\theta + w\theta^2 \in K$, such that the coefficient on θ^2 of δz^2 vanishes. This gives a quadratic equation $q_2(u, v, w) = 0$ with determinant $\det(q_2) = \mathcal{N}_{K/\mathbb{Q}}(\delta) = r^2$. If we follow the process of diagonalization suggested in [8] or [4], we have to solve the new equation $V^2 - \alpha U^2 - cW^2$, where $\alpha = Abc + Bc^2 - ac + b^2$. Hence, we have to factor both c and α.

In our example, θ is a root of $\theta^3 + 7823 = 0$, and δ is the fundamental unit of K: its coefficients a, b, $c \in \mathbb{Z}$ have about 1370 decimal digits, and its norm is only 1 (the regulator is about 6306.9). If we want to solve the diagonal Legendre equation, we have to factor c (about 1370 decimal digits) and α (about twice as many!), which is out of reach. If we use our new method instead, we only have to factor $r^2 = 1$, so there is no factorization to do (and also no minimization) and it only remains to solve the unimodular quadratic form q_2, whose coefficients have the size of a, b, and c: this takes only 1 second with Algorithm 1.3. We record here only the value of c:

$c =$47355 47642 38342 24877 99072 68459 54397 37493 79449 77195 50937 15271 96023 78702 35986 15693 32318 03073 96962 24642 59795 53147 92312 40881 72610 70891 77105 51426 13056 31285 70083 46940 10067 38064 96608 46156 58665 34864 18124 66382 09200 76958 35199 50394 77725 00014 54651 60673 20602 88379 58836 00959 25255 73399 68766 20231 05833 89860 34597 55786 83776 25628 25726 73794 89942 85394 32918 33006 88608 19014 78465 25124 59692 61429 88221 51680 44616 26181 13724 03145 21440 31030 84663 09890 20724 50488 61069 48766 15988 04854 64097 80681 83971 51702 28725 81522 20556 48833 45371 78786 39558 20893 85252 54441 30765 15325 30745 05560 72888 43070 87720 33760 36121 92697 00127 79708 71383 62874 41701 35637 27954 48964 48588 55219 50671 69581 56588 43037 45785 05866 09486 17728 94481 72639 24805 83508 09025 10878 20743 48511 22014 72796 98562 99812 30396 88176 86861 60576 55387 94588 37579 05779 03513 12618 46336 89855 43410 76374 53962 81655 32752 38561 64313 77612 56185 56445 27520 76296 38686 38771 70383 59844 84765 97233 53418 73696 67030 74121 22183 63083 04086 77691 83360 51810 50699 08408 52046 27144 85278 05072 76393 68278 86359 92021 21872 18920 38223 71149 53584 33110 16613 61300 64277 29041 07062 21578 81387 35328 86961 69316 83369 86647 89790 32589 24992 66880 42179 72512 60053 72967 71798 28828 38244 78910 21648 69938 04775 09451 65793 38502 78503 10149 58498 37198 85645 25202 21032 10982 98751 79521 97877 37537 17573 15753 82400 39513 81495 02461 34722 47217 97326 55105 37103 78126 80943 13206 17143 43277 00041 18122 19502 79438 38816 95155 20987 80276 46692 11549 82831 93578 08920 83133 8238 188

References

1. J.W.S. Cassels: Rational Quadratic Forms, L.M.S. Monographs, No.13. London, New York, San Francisco: Academic Press (1978). MR0522835 (80m:10019)
2. T. Cochrane and P. Mitchell: *Small solutions of the Legendre equation*, Journal of Number Theory **70** (1998), 62-66. MR1619944 (99a:11029)

3. H. Cohen: *A Course in Computational Algebraic Number Theory*, Graduate Texts in Math. **138**, Third corrected printing, Springer–Verlag (1996). MR1228206 (94i:11105)
4. J.E. Cremona, D. Rusin: *Efficient solution of rational conics*, Math. Comp. **72** (2003), 1417–1441. MR1972744 (2004a:11137)
5. C.F. Gauss: *Disquisitiones Arithmeticae*, Springer Verlag (1986). MR0837656 (87f:01105)
6. J.-P. Serre: *Cours d'arithmétique*, P.U.F. 3d edition (1988). MR0498338 (58:16473)
7. D. Simon: *Solving norm equations in relative number fields using S-units*, Math. Comp., vol. **71** No. 239 (2002), 1287 – 1305. MR1898758 (2003d:11044)
8. D. Simon: *Computing the rank of elliptic curves over number fields*, London Math. Soc. Journal of Computation and Mathematics, vol. **5** (2002), 7–17. MR1916919 (2003g:11060)
9. N.P. Smart: *The algorithmic resolution of Diophantine equations*, London Math. Soc. Student Texts **41**, Cambridge University Press, 1998. MR1689189 (2000c:11208)
10. B. Vallée: *An affine point of view on minima finding in integer lattices of lower dimensions*, Proc. of EUROCAL '87 (Leipzig, 1987), Lecture Notes in Comput. Sci. **378**, Springer, Berlin, 1989, 376–378. MR1033317 (92d:11069)

LMNO–UMR 6139, Université de Caen–Campus II, Bd du Maréchal Juin, BP 5186–14032 Caen Cedex, France

E-mail address: simon@math.unicaen.fr