Theoretical Computer Science 125 (1994) 229–241
Elsevier


# An algorithm for finding a shortest vector in a two-dimensional modular lattice*

Mody Lempel and Azaria Paz

*Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel*

Communicated by A. Salomaa
Received October 1991
Revised September 1992

*Abstract*

Lempel, M. and A. Paz, An algorithm for finding a shortest vector in a two-dimensional modular lattice, Theoretical Computer Science 125 (1994) 229–241.

Let $0<a,b<d$ be integers with $a\neq b$. The lattice $L_d(a,b)$ is the set of all multiples of the vector $(a,b)$ modulo $d$. An algorithm is presented for finding a shortest vector in $L_d(a,b)$. The complexity of the algorithm is shown to be logarithmic in the size of $d$ when the number of arithmetical operations is counted.

## 1. Introduction

A classical algorithm, due to Gauss, for finding a shortest vector in a two-dimensional lattice has been used as one of the main building blocks in the recent $L^3$ (Lenstra, A.K., Lenstra, H.W. Jr. and Lovasz, L.) basis reduction algorithm for general lattices [2]. The complexity of the Gauss algorithm has been shown to be logarithmic in the maximal integer among the entries of the vectors forming the basis of the lattice at input (when counting the number of arithmetical operations involved) [1].

Let $0<a,b<d$ be integers such that $a\neq b$. We define $L_d(a,b)$ to be the modular lattice generated by the vector $(a,b)$ modulo $d$, i.e. the (finite) set of all vectors of the form $(ia(\text{mod}\, d), ib(\text{mod}\, d)), 0\leqslant i<d$, which is closed under addition modulo $d$.

*Correspondence to*: A. Paz, Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel.

*The results shown in this paper are part of the M.Sc. Thesis of the first author done under the supervision of the second author, submitted to the Senate of the Technion.


*SSDI* 0304-3975(92)00021-M

We present, in this paper, an algorithm for finding a shortest vector in a lattice $L_d(a,b)$ as above, and we prove that the complexity of the algorithm is logarithmic in the size of $d$ when the number of arithmetical operations is counted.

While our algorithm bears certain similarities to the algorithm of Gauss, the two algorithms are different and cannot be reduced one to the other when the Gauss algorithm is considered over vectors with integer entries only. Thus, e.g. in the modular lattice generated by the vector (4, 1) modulo 5, a shortest (nonzero) vector is the vector $(2,3)=(3\cdot 4 \,(\mathrm{mod}\, 5), 3\cdot 1\,(\mathrm{mod}\, 5))$ (or the vector (3, 2) which has the same length). The shortest vector in the corresponding general (nonmodular) lattice containing the vectors (4, 1) and (3, 2) is $(-1,1)=(3-4, 2-1)$.

Conversely, consider the general (nonmodular) lattice with base vectors (7, 11) and (5, 8). The determinant $\left|\begin{smallmatrix}7 & 11\\ 5 & 8\end{smallmatrix}\right|$ is equal to 1. It can be shown that under these circumstances no $d>1$ exists such that $(5,8)=(i7\,(\mathrm{mod}\, d), i11\,(\mathrm{mod}\, d))$, $0\leqslant i<d$, since the existence of such a $d$ would imply that the above determinant has a value $\geqslant d$ (see Section 2).

It is hoped that this algorithm will enable generalizations for general $n$-dimensional modular lattices and will have applications to other areas of study (e.g. cryptology, coding theory, geometry of numbers, etc.).

## 2. Preliminaries

Given the integers $0<a,b<d$ and $i$, the notation $i(a,b)\,(\mathrm{mod}\, d)$ stands for the vector $(ia\,(\mathrm{mod}\, d), ib\,(\mathrm{mod}\, d))$.

We shall denote by $L_d(a,b)$ the modular lattice $L_d(a,b)=\{i(a,b)\,(\mathrm{mod}\, d)\colon 0\leqslant i\leqslant d-1\}$. We start with a few simple remarks:

(1) If $gcd(a,b,d)=g>1$ then the lattice $L_{d/g}(a/g, b/g)$ is an isomorphic contraction of the lattice $L_d(a,b)$. The shortest vector of the original lattice is equal to the shortest vector of the $L_{d/g}$ lattice multiplied by $g$. We shall assume therefore that $gcd(a,b,d)=1$.

(2) If $gcd(a,b,d)=1$ but $gcd(a,b)=g_1>1$ then $L_d(a/g_1, b/g_1)=L_d(a,b)$. This follows from the fact that $g_1$ is invertible modulo $d$, given that $gcd(a,b,d)=1$. Thus,

$$i(a,b)\,(\mathrm{mod}\, d)=j\left(\frac{a}{g_1}, \frac{b}{g_1}\right)(\mathrm{mod}\, d),$$

where

$$j=ig_1\,(\mathrm{mod}\, d) \text{ if } i \text{ is given} \quad\text{and}\quad i=jg_1^{-1}\,(\mathrm{mod}\, d) \text{ if } j \text{ is given.}$$

(3) Any two vectors in $L_d$ whose determinant is equal to $\pm d$ will be called a geometrical basis for $L_d$. It will be shown in the next section that any vector $(a,b)\neq(1,1)$ (with $gcd(a,b)=1$) belongs to a geometrical basis. Vectors in $L_d$ will be considered both as vectors and as points in two-dimensional space. Let $(a,b)$ and $(c,e)$ be a geometrical basis in $L_d$. Consider the topological torus formed from the square

$0 \leqslant x, y \leqslant d$ when its edges $x=0$, $y=0$ are identified with the edges $x=d$, $y=d$, respectively. The area of the face of this torus is $d^2$ and it is covered by $d$ nonoverlapping translates of the parallelogram whose vertices are $(0,0), (a,b), (c,e), (a+c, b+e)$ and whose area is $d$. It follows that the determinant of any 2 points in $L_d$ which are not colinear is equal to $\pm kd$, where $k$ is an integer such that $0<k<d$.

## 3. Some properties of $L_d$

**Lemma 3.1.** *Let $(c,e)$ be a point vector in $L_d(a,b)$ such that $gcd(c,e)=1$. If $c>1$ then there is a vector $(c_1, e_1)$ in $L_d(a,b)$ such that $\left|\begin{smallmatrix} c & e \\ c_1 & e_1 \end{smallmatrix}\right| = d$. If $e>1$ then there is a vector $(c_2, e_2)$ in $L_d(a,b)$ such that $\left|\begin{smallmatrix} c & e \\ c_2 & e_2 \end{smallmatrix}\right| = -d$.*

**Proof.** Assume that $c>1$. $gcd(c,e)=1$ implies that there are integers $u, v$ such that $cu-ev=1$. Multiplying by $d$ we get $cud-evd=d$. This equality induces the set of equalities

$$c(ud-ke)-e(vd-kc)=d$$

for any integer $k$.

Let $k_0$ be the maximal $k$ such that both $(ud-k_0e)=e_1 \geqslant 0$ and $(vd-k_0c)=c_1 \geqslant 0$. If both $e_1$ and $c_1$ are smaller than $d$, then from $(c_1, e_1)=(v,u)d-(c,e)k_0$ we get that $(c_1, e_1)$ is a modular multiple of a vector in $L_d(a,b)$ and satisfies therefore the requirement of Lemma 3.1. To complete the proof of the first part of the lemma we must show that $c_1, e_1 < d$.

From the choice of $k_0$ we know that either $c_1 < c$ or $e_1 < e$. Assume, by way of contradiction, that

$$ce_1 - ec_1 = d$$

and either

$$c_1 < c < d \text{ together with } e_1 \geqslant d$$

or

$$c_1 \geqslant d \text{ together with } e_1 < e < d.$$

In the first case we have that $c_1 \leqslant c-1$, $e<d$ and $e_1 \geqslant d$. Also, since $c>1$ (by assumption), $c-1>0$. Therefore,

$$d = ce_1 - ec_1 > cd - d(c-1) = d,$$

a contradiction.

In the second case, we have that $e_1 \leqslant e-1, c<d$ and $c_1 \geqslant d$. This implies that

$$d = ce_1 - ec_1 < d(e-1) - ed = -d,$$

which is impossible.

It follows that $c_1, e_1 < d$ and the proof of the first part of the lemma is complete. The proof of the second part is similar. □

**Remark.** The excluded point vector $(1,1)$ can never belong to a geometrical basis since the value of the determinant $|\begin{smallmatrix}1 & 1\\ c & e\end{smallmatrix}|$ is always less than $d$, in absolute value, given that $0 \leqslant c, e < d$. Moreover, if the vector $(1,1)$ belongs to a modular lattice $L_d$, then

$$L_d = \{(k,k);\ 0 \leqslant k < d\}.$$

No other vector $(c,e) \neq (k,k)$ can belong to $L_d$. Any such vector forms a determinant with $(1,1)$ in $L_d$ whose value is less than $d$, which cannot happen for vectors in $L_d$ (see Section 2).

The next few lemmas provide a characterization of the set of points forming a geometrical basis with a given vector.

**Lemma 3.2.** *Let $(a,b)$ be a vector in a lattice $L_d$. Let $(c,e)$ be another vector in $L_d$ such that $(a,b)$ and $(c,e)$ form a geometrical basis. If $gcd(a,b) = g > 1$ with $(a,b) = g(a',b')$ then any vector of the form $k(a',b')$, $1 \leqslant k < g$, is not in $L_d$.*

**Proof.** Since $(a,b)$ and $(c,e)$ form a basis we have that $|\begin{smallmatrix}a & b\\ c & e\end{smallmatrix}| = |\begin{smallmatrix}ga' & gb'\\ c & e\end{smallmatrix}| = \pm d$. Assume that the determinant equals $+d$ (the other case is similar). This implies that $0 < |\begin{smallmatrix}ka' & kb'\\ c & e\end{smallmatrix}| < |\begin{smallmatrix}ga' & gb'\\ c & e\end{smallmatrix}| = d$. Given that $(c,e)$ is in $L_d$, if $k(a',b')$ is in $L_d$ then $|\begin{smallmatrix}ka' & kb'\\ c & e\end{smallmatrix}|$ must be equal to 0 or a nonzero multiple of $d$, a contradiction. □

**Lemma 3.3.** *Let $(a,b)$ be a vector in a lattice $L_d$ and let $(c,e)$ and $(c',e')$ be two vectors in $L_d$ such that both form a basis with $(a,b)$. Then $(c,e)$ can be written in the form*

$$(c,e) = (c',e') + i(a,b) \quad \textit{or} \quad (c,e) = -(c',e') + i(a,b)$$

*for some $1 \leqslant i < d$.*

**Proof.** It follows from the assumptions that $ae - bc = \pm(ae' - bc')$. Let $gcd(a,b) = g$ with $(a,b) = g(a',b')$. Then

$$ga'(e \pm e') = gb'(c \pm c'),$$

implying that

$$e \pm e' = kb' \quad \text{and} \quad c \pm c' = ka'$$

(since $gcd(a',b') = 1$) for some integer $k$.

Thus,

$$(c,e)=(c',e')+k(a',b') \quad \text{or} \quad (c,e)=-(c',e')+k(a',b').$$

If $g=1$ or $g|k$ then we are done. To complete the proof we show that this is the only possible case. Otherwise, let $g>1$ and $k=gs+r$, $0<r<g$. Then

$$(c,e)=(c',e')+s(a,b)+r(a',b')$$

or

$$(c,e)=-(c',e')+s(a,b)+r(a',b').$$

In both cases $r(a',b')$ must be in $L_d$ since $(c,e),(c',e')$ and $s(a,b)$ are in $L_d$ and all the entries of all the vectors involved are nonnegative. But this contradicts Lemma 3.2 since $0<r<g$. □

**Corollary 3.4.** *Let $(a,b)$, $(c,e)$ be two vectors in $L_d$ which are a geometrical basis. The set of all vectors forming a basis with $(a,b)$ in $L_d$ is the set* (∗):

$$\{(p,q)=\pm(c,e)+i(a,b):\ -d\leqslant i\leqslant d,\ 0\leqslant p,q<d\}. \qquad (*)$$

**Lemma 3.5.** *Let $i_0$ be the maximal $i$ such that $(c,e)-i_0(a,b)$ is nonnegative and let $i_1$ be the minimal $i$ such that $-(c,e)+i_1(a,b)$ is nonnegative in the set* (∗). *Then the shortest vector in the set* (∗) *is the shortest of $(p',q')$ and $(p'',q'')$, where*

$$(p',q')=(c,e)-i_0(a,b),$$

$$(p'',q'')=-(c,e)+i_1(a,b).$$

**Proof.** Left to the reader. □

**Remark.** Note that $i_0$ can be defined as

$$i_0=\begin{cases} \text{if } a,b>0 \text{ then } \min\{\lfloor \frac{c}{a}\rfloor,\lfloor \frac{e}{b}\rfloor\},\\ \text{if } b=0 \text{ then } \lfloor \frac{c}{a}\rfloor,\\ \text{if } a=0 \text{ then } \lfloor \frac{e}{b}\rfloor,\end{cases}$$

and $i_1$ can be defined in a similar way. It follows that the number of operations involved in the computation of the shortest vector in the set (∗) is constant.

The algorithm to be presented in the sequel bears some resemblance to the classical algorithm of Gauss for finding a minimal vector in a general two-dimensional lattice. As in the classical algorithm, after a geometrical basis is found (Lemma 3.1), a sequence of decreasing vectors in the lattice is generated until the minimal vector is found. A particular case in our algorithm needs special attention in order to keep the complexity of the general algorithm linear (in the magnitude of $d$). A procedure for handling this particular case (defined below) is provided in Section 4.

## 4. Crossing vector procedure

**Definition 4.1.** Let $v_1=(v_{11},v_{12})$ and $v_2=(v_{21},v_{22})$ be nonnegative vectors. Let $\delta=(\delta_1,\delta_2)=(v_{11}-v_{21},v_{12}-v_{22})$ be their difference vector. $v_1$ and $v_2$ are *crossing* if $\delta_1\delta_2<0$. They are left crossing if $\delta_1>0$ ($\delta_2<0$) and are right crossing if $\delta_1<0$ ($\delta_2>0$). $|v|$ denotes the length of a vector $v$.

**Procedure Min-Cross** (*finding a minimal-length vector in a lattice defined by a crossing basis*).
*Input*: A basis $v_2, v_1$ for a lattice $L_d$ such that $|v_2|<|v_1|$, $v_2$ and $v_1$ are crossing. Let $v_{2j}$ and $v_{1j}$ be the entries in $v_2$ and $v_1$ such that $v_{1j}-v_{2j}=\delta_j>0$. Denote by $u_s$ the vector $u_s=v_2-s(\delta_1,\delta_2)$ with $u_{-1}=v_1$.

1. If $v_{2j}=0$ return $v_2$, halt.
2. Repeat until
   $|v_2|>|v_1|$ or $v_{2j}=0$.
   begin
   2.1. Find $k:=\left\lceil \frac{v_{1j}}{v_{2j}} \right\rceil$
   {Remark: $v_{1j}>v_{2j}$ implies that $k\geqslant 2$}
   2.2. If $k=2$ do begin
   Set $m:=\left\lfloor \frac{v_{2j}}{\delta_j} \right\rfloor$;
   {Remark: $k=2$ implies $m\geqslant 1$}
   Set $p_1=\left\lfloor \frac{v_{21}\delta_1+v_{22}\delta_2}{\delta_1^2+\delta_2^2} \right\rfloor$
   {Remark: $p_1\leqslant m$}
   2.3. If $p_1<0$ return $v_2$, halt;
   define $p=\begin{cases} p_1 & \text{if } |u_{p_1}|\leqslant|u_{p_1+1}| \text{ or } p_1=m \\ p_1+1 & \text{otherwise} \end{cases}$
   2.4. If $p\leqslant m-1$ return $u_p$, halt;
   2.5. Set $v_2:=u_p$, $v_1:=u_{p-1}$, $k:=\left\lceil \frac{v_{1j}}{v_{2j}} \right\rceil$
   end;
   {Remark: Now $k>2$ and $\delta_j>0$}
   2.6. Set $v_3:=v_2$, $v_2:=-v_1+kv_2$, $v_1:=v_3$
   end (repeat);

3. If $|v_2|>|v_1|$ return $v_1$, else return $v_2$
end of algorithm.

## 5. Properties of the Min-Cross procedure

The vectors at input $v_1$ and $v_2$ are assumed to satisfy the following properties:
(a) $|v_2|<|v_1|$,
(b) $v_2$ and $v_1$ are cross vectors,
(c) $v_2$ and $v_1$ are a geometrical basis for a lattice $L_d$.
Consider the sequence of vectors

$$v_1, v_2, v_3, \dots, v_t \tag{1}$$

such that for all $i>2$ the following properties hold:
(d) $|v_i|<|v_{i-1}|$, $i\geq 2$,
(e) $v_i$ is the shortest vector in $L_d$ which forms a basis for $L_d$ with $v_{i-1}$.
We proceed to prove the following theorem.

**Theorem 5.1.** (f) *$v_i$ and $v_{i-1}$ are cross vectors, of the same type (left or right) as $v_2$ and $v_1$, for all $i\geq 2$.*

(g) *The vectors generated at steps 2.5 and 2.6 by the procedure Min-Cross are a subsequence of the sequence* (1), *starting from $v_2$ and on.*

(h) *$v_t$, the last vector in* (1), *is the last vector generated by the procedure, at one of the steps* 1, 2.3, 2.4, 3.

(i) *The number of iterations of the procedure is logarithmic in the magnitude of d.*

**Proof.** (f): It is proved by induction. By assumption $v_2$ and $v_1$ are cross vectors. Assume that $v_{i-1}$ and $v_{i-2}$, $i\geq 3$, are left cross vectors with $v_{i-1,1}<v_{i-2,1}$ and $v_{i-1,2}>v_{i-2,2}$ (the right cross case is similar). If $v_{i-1}$ is not the last vector in the sequence then, by the properties of $v_i$ ((d) and (e)) and by Lemma 3.5, $v_i$ must be one of the vectors

$$v_i=v_{i-2}-i_0 v_{i-1} \quad \text{or} \quad v_i=-v_{i-2}+i_1 v_{i-1},$$

where $i_0$ and $i_1$ are as defined in Lemma 3.5. The first of the two choices requires $i_0=0$, otherwise $v_i$ is a vector with negative entries since $v_{i-1}$ and $v_{i-2}$ are cross. The only possible choice which can result in a shorter vector is therefore $v_i=-v_{i-2}+i_1 v_{i-1}$. If $v_{i-1,1}=0$ (steps 1 and 2 in the procedure) then no shorter vector in $L_d$ can form a basis with $v_{i-1}$. This follows from the fact that $v_{i-2,1}>v_{i-1,1}=0$ (our assumption) so that $v_i=-v_{i-2}+i_1 v_{i-1}$ is a vector with negative entries. Therefore, if $v_{i-1,1}=0$ then $v_{i-1}=v_t$ is the last vector in the sequence (1). Otherwise, with $i_1=\lceil v_{i-2,1}/v_{i-1,1}\rceil\geq 2$ (since $v_{i-2,1}>v_{i-1,1}$) and $v_i=-v_{i-2}+i_1 v_{i-1}$ we have that $v_{i,1}$ is $v_{i-1,1}$ minus the

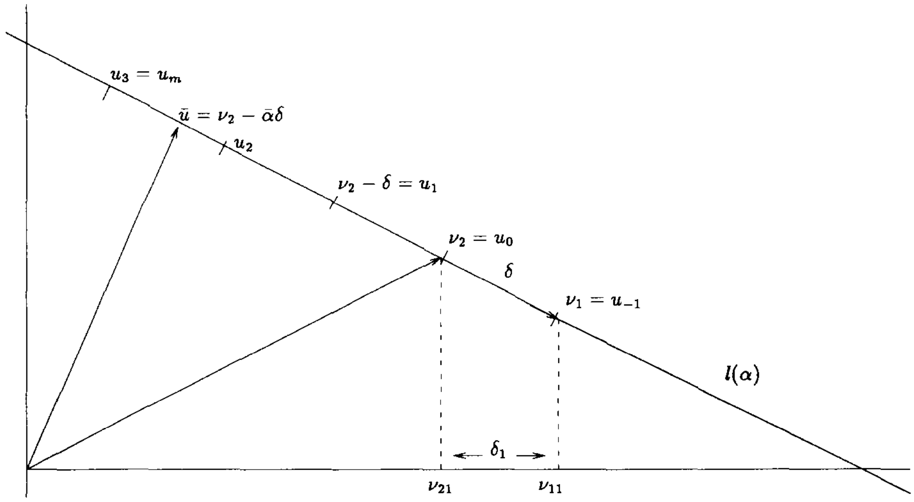


Fig. 1.

remainder from the division of $v_{i-2,1}$ by $v_{i-1,1}$, so $v_{i,1} < v_{i-1,1}$. The second entry in $v_i$ satisfies

$$v_{i,2} = -v_{i-2,2} + i_1 v_{i-1,2} \geqslant -v_{i-2,2} + 2v_{i-1,2}$$
$$= (-v_{i-2,2} + v_{i-1,2}) + v_{i-1,2} > v_{i-1,2}$$

(since $v_{i-1,2} > v_{i-2,2}$). It follows that $v_i$ and $v_{i-1}$ are left cross vectors as required.

(g) *and* (h): We shall follow the procedure step by step to verify that the generated sequence satisfies the required properties, using an inductive argument. Assume that the procedure has gone through several iterations and that the subsequence of vectors, ending in the currently defined vectors $v_1, v_2$, generated so far, satisfies the required properties (this is vacuously true at the input stage). For the sake of simplicity, we shall assume w.l.o.g. that $j$, as defined in the procedure, is given by $j = 1$ (i.e. $v_2$ and $v_1$ are left cross vectors).

*Step* 1: If $v_{21} = 0$ then, as explained in the proof of property (f), $v_2$ is the last vector in the sequence (1) and is the last vector generated by the procedure, as required.

Before considering steps 2 and 3, we must prove some additional properties of the lattice $L_d$. Consider Fig. 1, where the line passing through $v_1$ and $v_2$ is

$$l(\alpha) = v_2 - \alpha(v_1 - v_2) = v_2 - \alpha\delta.$$

Let $m = \lfloor v_{21}/\delta_1 \rfloor$ and assume $k = \lceil v_{11}/v_{21} \rceil = 2$.

**Claim 1.** *$k = 2$ implies that $m \geqslant 1$.*

**Proof of Claim 1.** $k=2$ implies that $v_{11} \leqslant 2v_{21}$ by the definition of $k$. This implies that $\delta_1 = v_{11} - v_{21} \leqslant v_{21}$ or $v_{21}/\delta_1 \geqslant 1$, resulting in $m = \lfloor v_{21}/\delta_1 \rfloor \geqslant 1$. □

Define, as before, $u_i = v_2 - i\delta$ ($u_{-1} = v_1$, $\delta = v_1 - v_2$), $k_i = \lceil u_{i-1,1}/u_{i,1} \rceil$ ($k = k_0$), and assume the following: $k_0 = 2$, $u_0$ is the shortest vector in $L_d$ forming a basis with $u_{-1}$ and $u_0, u_{-1}$ are left cross vectors.

**Claim 2.** *Under the above assumptions, for all* $0 \leqslant i < m$, $k_i = 2$, *the vector* $u_i$ *is the shortest vector forming a basis in* $L_d$ *with* $u_{i-1}$, $u_i$ *and* $u_{i-1}$ *are left cross, and* $u_{i-1} - u_i = \delta$.

**Proof of Claim 2.** By induction. For $i=0$, the properties follow from the definitions and assumptions.

Assume now that, for $i>0$, $k_{i-1}=2$, $u_{i-1}$ is the shortest vector forming a basis in $L_d$ with $u_{i-2}$. $u_{i-1}$ and $u_{i-2}$ are left cross vectors and $u_{i-2} - u_{i-1} = \delta$. The shortest vector forming a basis in $L_d$ with $u_{i-1}$ is

$$2u_{i-1} - u_{i-2} = u_{i-1} - (u_{i-2} - u_{i-1}) = u_{i-1} - \delta$$

and

$$u_{i-1} - \delta = v_2 - (i-1)\delta - \delta = v_2 - i\delta = u_i.$$

It follows from the above equalities that $u_{i-1} - u_i = \delta$.

To show that $u_i$ and $u_{i-1}$ are left cross, one can use an argument similar to the argument used in the proof of property (f).

Finally, since $i<m$, we have that $u_{m,1} = v_{21} - m\delta_1$ is positive and closer to the origin than $u_{i1} = v_{21} - i\delta_1$ by a multiple of $\delta_1$. Thus, $u_{i1} \geqslant \delta_1$. Therefore, $u_{i1} \geqslant u_{i-1,1} - u_{i,1} = \delta_1$ or $2u_{i1} \geqslant u_{i-1,1}$. But $u_{i,1} < u_{i-1,1}$ since $u_i$ and $u_{i-1}$ are left cross. Therefore, $k_i = \lceil u_{i-1,1}/u_{i,1} \rceil = 2$ and the proof is now complete. □

**Claim 3.** *Under the same assumptions as in Claim 2,* $u_m$ *is the shortest vector forming a basis for* $L_d$ *with* $u_{m-1}$, $u_{m-1} - u_m = \delta$. $u_m$ *and* $u_{m-1}$ *are left cross but* $k_m \geqslant 3$.

**Proof of Claim 3.** The proofs of the first three properties are similar to the corresponding proofs in Claim 2. Now, by the definition of $m$, $u_{m,1} < \delta_1$ (since $u_m - \delta$ has a negative $x$-coordinate), while $u_{m-1,1} = \delta_1 + u_{m,1}$. Thus, $u_{m-1,1} > 2u_{m,1}$ and $k_m = \lceil u_{m-1,1}/u_{m,1} \rceil > 2$. □

Claim 3 proves the remark after step 2.5.

To find the shortest vector on the line $l(\alpha)$ we can differentiate the value $\beta^2(\alpha) = (v_{21} - \alpha\delta_1)^2 + (v_{22} - \alpha\delta_2)^2$:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}(\beta^2(\alpha)) = -2((v_{21} - \alpha\delta_1)\delta_1 + (v_{22} - \alpha\delta_2)\delta_2) = 0.$$

The solution is $\bar{\alpha}=(v_{21}\delta_1+v_{22}\delta_2)/(\delta_1^2+\delta_2^2)$, resulting in $\bar{u}=v_2-\bar{\alpha}\delta$.

We can proceed now with the analysis of our procedure.

Assume $k=2$. If $\bar{\alpha}<0$ (step 2.3, $p_1=\lfloor\bar{\alpha}\rfloor$) then there is no vector shorter than $v_2$ forming a basis for $L_d$ with $v_1$ (the shortest such vector is the vector $v_2-\delta$ which must be longer than $v_2$ since $v_2$ is longer than $v_2+(-\bar{\alpha})\delta$). $v_2$ must therefore be the final vector in the sequence (1) and the procedure halts.

Let $u_r$ be the shortest vector in the sequence $(u_i)$ on $l(\alpha)$.

If $r=m$: this happens if $p_1=\lfloor\bar{\alpha}\rfloor=m$ or $p_1=\lfloor\bar{\alpha}\rfloor=m-1$ but $p=m(|u_m|<|u_{m-1}|)$. Then the sequence $u_{-1},u_0,\ldots,u_m$ is the subsequence $v_1,v_2,\ldots,v_{m+2}$ of (1), by Claim 2 with $m\geqslant 1$ (Claim 1). This case corresponds to step 2.5 in the procedure. The vectors $v_1$ and $v_2$ are reset and the procedure continues with step 3.

If $r<m$, then $u_r$ is one of the vectors $v_2-\lfloor\bar{\alpha}\rfloor\delta$, $v_2-\lceil\bar{\alpha}\rceil\delta$. $u_{r+1}$with $r+1\leqslant m$ is the shortest vector in $L_d$ which forms a base with $u_r$. But in this case $|u_{r+1}|>|u_r|$ and therefore the sequence (1) terminates with $u_r=v_{r+2}$. If this is the case the procedure halts with $u_r$ at output.

If $k\geqslant 3$, then the procedure proceeds directly to step 2.6 and it either halts with $v_1$ at output (if the new $v_2$ (the shortest vector forming a basis with $v_1$) is longer than $v_1$) or it halts with the new $v_2$ at output (if the new $v_2$ is terminal) or it proceeds with a new iteration. The proof of properties (g) and (h) is thus complete.

**Proof of Theorem 5.1** (*conclusion*). Let $v_{j+2}$ be the new $v_2$ vector created at step 2.6 at iteration $j$. The application of step 2.6 is based on $k>2$. Therefore, $k=\lceil v_{j+1,1}/v_j+2,1\rceil>2$ or $v_{j+1,1}>2v_{j+2,1}$. Thus, the new first coordinate of $v$ is decreased by a factor of at least 2. The number of iterations is therefore logarithmic in the magnitude of the coordinates of the vectors at input which are bounded by $d$.

All properties of the procedure are now proved. □

## 6. The main algorithm

To find the shortest vector in a modular lattice $L_d$ generated by a vector $v_1=(a,b)$, modulo $d$, $a\neq b$, apply the following algorithm.

1. Assume $gcd(a,b,d)=1$
2. If $gcd(a,b)=g>1$ then reset $(a,b):=(1/g)(a,b)$. Now $(a,b)\neq(1,1)$
3. Based on Lemmas 3.1 and 3.5 find the shortest vector $v_2$ forming a basis with $v_1$
4. While $|v_2|<|v_1|$
   - 4.1. If $v_2$ and $v_1$ are crossing
     return (Min-Cross $(v_1,v_2)$)
   - 4.2. $v_1:=v_2$
   - 4.3. Based on Lemma 3.5 find the shortest vector $v_2$ forming a basis with $v_1$
5. Return $v_1$

We conclude now by showing that the algorithm is correct and that its complexity is logarithmic in the size of $d$ (when counting the number of arithmetical operations).

## 7. Proof of correctness

When the algorithm terminates, either via the Min-Cross procedure or at step 5, it produces a vector $v$ such that no vector in $L_d$ forming a basis with $v$ is shorter than $v$. We claim that such a vector $v$ is the shortest vector in $L_d$. We first need the following.

**Lemma 7.1.** *Let* ABC *be a triangle in the plane such that the vertices* A, B, C *correspond to vectors in* $L_d$. *If the area of* ABC *is greater than* $d/2$ *then there must be a point of* $L_d$ *different from* A, B *and* C *on the border of or inside the triangle.*

**Proof.** Consider the torus formed by identifying the edges $x=d$, $y=d$ with the edges $x=0$, $y=0$, respectively, of the square $\{(x,y): 0\leqslant x, y\leqslant d\}$. The area of the face of this torus is $d^2$. If no lattice point exists inside or on the border of ABC then the parallelogram formed by the edges AB and AC has no lattice points inside or on its border, except its vertices. Under the assumption of Lemma 7.1, the area of the parallelogram is greater than $d$. Thus, $d$ translates of this parallelogram will cover the whole torus with no overlap, implying that the area of the torus is greater than $d^2$, a contradiction. □

**Theorem 7.2.** *If a vector* $v$ *in* $L_d$ *has the property that no vector in* $L_d$ *forming a basis with* $v$ *is shorter than* $v$, *then* $v$ *is the shortest vector in* $L_d$.

**Proof.** Assume to the contrary that there is a vector $v_1$ shorter than $v$ in $L_d$. $v_1$ cannot form a basis with $v$ by the properties of $v_1$. Therefore, the triangle whose vertices are $0, v, v_1$ (0 is the origin) must have an area which is greater than $d/2$. Both vectors $v$ and $v_1$ belong to some basis and therefore, by Lemma 3.2, no vector in $L_d$ can subdivide $v$ or $v_1$. By Lemma 7.1 there must be a point in $L_d$ inside the triangle or on the line joining $v$ to $v_1$. Let $v_2$ be such a point; then obviously $|v_2|<|v|$ (since $|v_1|<|v|$) and the area of the triangle whose vertices are $0, v_2, v$ is smaller than the area of the original triangle. Choose $v_2$ to be a point as above and such that the area of the triangle whose vertices are $0, v_2, v$ is minimal. It follows from the choice of $v_2$ that no vector in $L_d$ can be inside the minimal triangle or on its $(0, v_2)$ or $(v, v_2)$ boundaries. Now $v_2$ cannot form a basis with $v$ since $|v_2|<|v|$. Therefore, the area of the minimal triangle must be greater than $d/2$. But this contradicts, by Lemma 7.1, the fact that no points of $L_d$ exist inside or on the boundary of this minimal triangle. The algorithm is thus shown to be correct. □

## 8. Complexity analysis

If, at step 4, $|v_2| \geqslant |v_1|$, the algorithm halts. If, at step 4.1, $v_2$ and $v_1$ are crossing, then the algorithm enters procedure Min-Cross and will eventually halt, while executing this procedure, in at most $O(\log_2 d)$ steps.

Let $v_i, v_{i-1}, v_{i-2}$ be the vectors generated at step 4.3 at the $i, i-1$ and $i-2$ iterations, respectively, with $i \geqslant 2$. Since $|v_{i-1}| < |v_{i-2}|$ with $|v_{i-1}|^2$ and $|v_{i-2}|^2$ integers, the algorithm will eventually halt. Since the algorithm did not enter the procedure Min-cross at step 4.1, we must have that $|v_{i-1}| < |v_{i-2}|$ and $v_{i-1}, v_{i-2}$ are not crossing. Therefore, $v_{i-1,1} \leqslant v_{i-2,1}$, $v_{i-1,2} \leqslant v_{i-2,2}$, and at least one of the inequalities is strict. Let $k_i = \min(\lfloor v_{1-2,1}/v_{i-1,1} \rfloor, \lfloor v_{i-2,2}/v_{i-1,2} \rfloor)$; $k_i \geqslant 1$ (since the vectors are not cross). The vector $v_i$ generated at step 4.3 is either equal to $v_{i-2} - k_i v_{i-1}$ or a shorter vector (in case $-v_{i-2} + i_1 v_{i-1}$, as defined in Lemma 3.5, is shorter than $v_{i-2} + i_0 v_{i-1}$). Set $v_i' = v_{i-2} - k_i v_{i-1}$. It follows that

$$|v_i'| = |v_{i-2} - k_i v_{i-1}| \geqslant |v_i|.$$

Now $v_{i-2} = v_i' + k_i v_{i-1}$, which implies that

$$|v_{i-2}| = |v_i' + k_i v_{i-1}| \geqslant |v_i' + v_{i-1}|$$

since $k_i$ is positive and the entries of the vectors involved are nonnegative.

Consider the parallelogram whose vertices are the origin O and the points A, B, C, corresponding to $v_{i-1}, v_i'$ and $v_i' + v_{i-1}$, all in the positive quadrant. Since $v_i'$ and $v_{i-1}$ are both in the positive quadrant, the origin is an acute angle in the parallelogram and the angle between the edges OA and AC is obtuse. It follows from the law of cosines that $OC^2 \geqslant OA^2 + AC^2 = OA^2 + OB^2$, which implies that

$$|v_i' + v_{i-1}|^2 \geqslant |v_i'|^2 + |v_{i-1}|^2.$$

Combining the last three inequalities we get that

$$|v_{i-2}|^2 \geqslant |v_i' + v_{i-1}|^2 \geqslant |v_i'|^2 + |v_i - 1|^2 \geqslant |v_i|^2 + |v_{i-1}|^2.$$

Note also that the numbers involved in the above inequality are nonnegative integers.

Let $t$ be the number of iterations of the algorithm through step 4.3 and let $\phi$ be the positive solution of the equation $x^2 = x + 1$, $\phi = (1 + \sqrt{5})/2$. Then

$$|v_t|^2 \geqslant 1,$$

$$|v_{t-1}|^2 \geqslant |v_t|^2 \geqslant 1,$$

$$|v_{t-2}|^2 \geqslant |v_{t-1}|^2 + |v_t|^2 \geqslant 2 > \phi,$$

$$|v_{t-3}|^2 \geqslant |v_{t-2}|^2 + |v_{i-1}|^2 > \phi + 1 = \phi^2,$$

$$|v_{t-j}|^2 \geqslant |v_{t-j+1}|^2 + |v_{t-j+2}|^2 > \phi^{j-1},$$

$$|v_0|^2 = |v_{t-t}|^2 > \phi^{t-1}.$$

We get that

$$(t-1)\log\phi<\log|v_0|^2<4\log d$$

or

$$t<\frac{4\log d}{\log\phi}+1.$$

The complexity of the algorithm is thus shown to be logarithmic in the magnitude of $d$. If the algorithm does not enter the Min-Cross procedure then the number of iterations before it halts is bounded as above. If it enters the procedure Min-Cross then the number of iterations before entering the procedure is also bounded as above, and after entering the procedure Min-Cross the algorithm will stay in the procedure no more than a logarithmic number of iterations before halting.

## References

[1] J.C. Lagarias, Worst-case complexity bounds for algorithms in the theory of integral quadratic forms. *J. Algorithms* **1** (1980) 142–186.

[2] A.K. Lenstra, H.W. Lenstra and L. Lovasz, Factoring polynomials with rational coefficients, *Math. Ann.* **261** (1982) 513–534.