Computing isometries of lattices

W. Plesken     B. Souvignier

Introduction. We report on some new ideas to calculate the group of automorphisms of a lattice with a positive definite quadratic form, which lead to an improvement of an algorithm described in [PlP 85] and [Sou 91]. To illustrate the reach of the algorithm e.g. 2 matrices generating the automorphism group of the Leech lattice can be calculated in less than 3 hours on a HP 9000/730 workstation. Moreover, the automorphism group of the densest known lattice $Q_{32}$ in dimension 32 (cf. [Que 84]) is for the first time determined in this paper. The algorithm has successfully been used in various situation cf. for instance [Sch 92], [PIN 93], [PIP 93], or [Sou 94].

Notation. Let $L$ be a $\mathbb{Z}$-lattice of rank $n$, $\Phi : L \times L \to \mathbb{Z}$ a positive definite bilinear form on $L$, $B = (b_1 \ldots b_n)$ a $\mathbb{Z}$-basis of $L$ and $F$ the Gram matrix of $\Phi$ with respect to $B$, i.e. $F_{ij} := \Phi(b_i, b_j)$. Denote by $G$ the automorphism group of $L$ and by $G_i$ the pointwise stabilizer of $b_1, \ldots, b_{i-1}$ in $G$. Let $S$ be the (finite) set of vectors $v \in L$ of norm $\Phi(v,v)$ up to $\max(F_{ii} | 1 \leq i \leq n)$.

The basic idea. For $k \leq n$ call $(v_1, \ldots, v_k)$ a $k$-partial automorphism if $\Phi(v_i, v_j) = F_{ij}$ for $1 \leq i, j \leq k$. Obviously the $n$-partial automorphisms represent the automorphisms of $L$. Whereas Witt’s Theorem on extending partial isometries guarantees that every $k$-partial automorphism extends to an automorphism in case of vector spaces over fields, here it can happen that a $k$-partial automorphism can not even be extended to a $(k+1)$-partial automorphism. Because of the finiteness of $S$ testing whether for $k < n$ a $k$-partial automorphism extends to a $(k+1)$-partial automorphism can be checked in reasonable time. However, it is highly desirable in a backtrack search to reject as early as possible $k$-partial automorphisms not extending to automorphisms. One therefore has to find in addition to the scalar products further testable properties of $k$-partial automorphisms, which are restrictions of $n$-partial automorphisms.

The Fingerprint. One property, which was already used in the algorithm in [PlP 85], is that the number of extensions of a $k$-partial automorphism to a $(k+1)$-partial automorphism must be preserved by automorphisms. These numbers can also be used to obtain a better ordering for the $b_i$. This information is stored in a matrix, called fingerprint, which is calculated as follows: Suppose that the indices $j_1 \ldots j_{i-1}$ are already chosen, then define $f_{ik}$ by $f_{ik} := 0$ if $k \in \{j_1 \ldots j_{i-1}\}$ and $f_{ik} := |\{v \in S | \Phi(v,v) = F_{kk} \text{ and } \Phi(v, b_j) = F_{kj} \text{ for } l = 1 \ldots i-1\}|$ else. Now choose $j_i$ such that $f_{ij_i}$ is minimal among the $f_{ik} \neq 0$. Then
$b_i \rightarrow b_{j_i}$ is a permutation of the basis vectors such that in each step the possible number of continuations of a partial automorphism is minimal. To avoid a mess of indices we assume for the rest of this paper that the permutation is the identity, i.e. $j_i = i$ for all $i$. Define $f_i := f_{i_t}$, then one has $|G| \leq \prod_{i=1}^{n} f_i$, since $f_i$ is an upper bound for the length of the orbit $b_i G_i$. A measure how ill-conditioned a lattice (and its basis) is for this algorithm is the product $\prod_{i=k+1}^{n} f_i$, where $k$ is the minimal index such that $f_k \neq |b_k G_k|$, since this is a (realistic) upper bound for the number of dead ends in the backtrack search after choosing a wrong image for $b_k$. Clearly this can only be analyzed a posteriori.

**Vector sums.** Especially for lattices, which do not have a big symmetry, the fingerprint alone does not suffice to detect dead ends early enough. Instead of counting the number of vectors with correct scalar products only, one can also take into account arbitrary combinations of scalar products and not only count the vectors having this combination of scalar products but use their sum. More precisely:

For $s = (s_1, \ldots, s_l) \in \mathbb{Z}^l$ and an $l$-partial automorphism $\varphi = (v_1, \ldots, v_l)$ define $X_s(\varphi) := \{w \in S | \Phi(w, v_i) = s_i \text{ for } i = 1, \ldots, l\}$ and $\sum_s X_s(\varphi) := \sum_{w \in X_s(\varphi)} w$. Then for an automorphism $\varphi$ of $L$ one has $X_s(b_1, \ldots, b_l) \varphi = X_s(b_1 \varphi, \ldots, b_l \varphi)$.

The preprocessing of the algorithm consists of the computation of the fingerprint and of the vector sums. What is stored after the preprocessing is:

1) The fingerprint.
2) For each $1 \leq l \leq n$:
   a) a $\mathbb{Z}$-basis $B_l$ of the lattice generated by the $X_s(b_1, \ldots, b_l)$, expressed as a linear combination of certain of the $X_s(b_1, \ldots, b_l)$,
   b) the scalar products of the vectors in $B_l$,
   c) the coordinates of all $X_s(b_1, \ldots, b_l)$ with respect to $B_l$.

Since the number of vectors $X_s$ explodes for growing $l$, it turned out to be efficient to replace the sets $X_s$ by sets $X'_s(v_1, \ldots, v_l) := \{w \in S | \Phi(w, v_i) = s_i \text{ for } i = l_0, \ldots, l\}$ with $l_0 := \max(1, l + 1 - d)$, where $d$ is some chosen constant, which we call the depth parameter. The value $d = 0$ corresponds to the old method using only the fingerprint, the value $d = 1$ more or less to the old method using “types” for the elements of $S$ in addition.

**The backtrack search.** Suppose that $(v_1 \ldots v_{i-1})$ is an $(i-1)$-partial automorphism and let $C_i$ be the list of candidates for the image of $b_i$ which have not yet been tested as a continuation of $(v_1 \ldots v_{i-1})$. Choose a vector $v \in C_i$, replace $C_i$ by $C_i - \{v\}$ and check whether the number of possible continuations of $(v_1, \ldots, v_{i-1}, v)$ is $f_{i+1}$. Then calculate the vector sums $X_s(v_1, \ldots, v_{i-1}, v)$ and with the help of the coefficients stored in 2a) the “pseudo image” $B'_i$ of the basis $B_i$, Now check, whether the scalar products of the vectors in $B'_i$ coincide with those stored in 2b) and whether the linear combinations of the vectors in $B'_i$ with the coefficients stored in 2c) coincide with the vector sums $X_s(v_1, \ldots, v_{i-1}, v)$. If all conditions are fulfilled, set $v_i := v$ and proceed to the next step, else choose the next vector from $C_i$. If
$C_i = \emptyset$ decrease $i$ until $C_i \neq \emptyset$.

**The stabilizer chain.** What has been said so far, refers to the calculation of just one automorphism. To obtain the full group of automorphisms one follows the concept of strong generating sets as introduced in [Sim 71] with $B$ as base for the action of $G$, i.e. one calculates a generating set $G$ for $G$ such that $G_i$ is generated by $G \cap G_i$ for $i = 1, \ldots, n$. For the construction of the set $G$ one has two methods: Schreier generators lying in $G_{i+1}$ are obtained as quotients of two elements in $G_i$ mapping $b_i$ to the same vector $v$. These are cheap, however do not increase the group $H \leq G$ constructed so far. On the other hand one obtains generators from the backtrack search. Here the fingerprint gives a good guideline:

Let $k$ be minimal such that $|b_k H_k| < f_k$, where $H_k := \langle G \cap G_k \rangle$ and let $C_k$ be the set of candidates for the image of $b_k$ not lying in $b_k H_k$, then check for representatives $v$ of the orbits of $H_k$ on $C_k$, whether there exists $\gamma \in G_k$ with $b_k \gamma = v$. If such a $\gamma$ exists, replace $G$ by $G \cup \{\gamma\}$, $H_k$ by $\langle H_k, \gamma \rangle$ and the orbits by their fusions under $\gamma$. If not, replace $f_k$ by $f_k - |v H_k|$. Then remove $v H_k$ from $C_k$.

**Examples.** An example showing the quality of the fingerprint is the Leech lattice. We choose a basis of norm 4 vectors. For our choice the values for $f_k$ resp. $|b_k G_k|$ are:

<table>
<thead>
<tr>
<th>$k$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_k$</td>
<td>196560</td>
<td>4600</td>
<td>891</td>
<td>336</td>
<td>32</td>
<td>40</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>b_k G_k</td>
<td>$</td>
<td>196560</td>
<td>4600</td>
<td>891</td>
<td>336</td>
<td>32</td>
<td>40</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The full automorphism group $2Co_1$ of order $2^{22} \cdot 3^9 \cdot 5^4 \cdot 7^2 \cdot 11 \cdot 13 \cdot 23$ can be calculated in less than 3 hours CPU-time on a HP 9000/730.

In contrast to that one has for a lattice in the genus of the 16-dimensional Barnes-Wall lattice (cf. [ScV 93] lattice $L_4$ with root system $8C_2$), generated by vectors of norm 2 and 4:

<table>
<thead>
<tr>
<th>$k$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_k$</td>
<td>32</td>
<td>30</td>
<td>28</td>
<td>26</td>
<td>16</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>16</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>$</td>
<td>b_k G_k</td>
<td>$</td>
<td>32</td>
<td>28</td>
<td>24</td>
<td>4</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Choosing the depth $d = 5$ the automorphism group of order $2^{30} \cdot 3 \cdot 7$ can be computed in 16 seconds. For $d = 4$ it takes already more than 5 minutes. Using only the fingerprint ($d = 0$) there was no result even after 50 hours.

A really involved example is the densest known lattice $Q_{32}$ in dimension 32, which is described in [Que 84]. With respect to some basis of norm 6 vectors one has:

| $k$ | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  | 19  | 20  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_k$ | 261120 | 1520 | 192  | 2   | 189 | 4   | 3   | 1   |     |     |     |     |     |     |     |     |     |     |     |     |     |
| $|b_k G_k|$ | 1920 | 2   | 1   | 1   | 18  | 1   | 3   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |

With depth $d = 4$ it takes already 1 hour CPU-time to calculate $G_3$ and then another 12 hours to calculate $G_2$, which would be impossible using only the fingerprint. For larger values of $d$ the number of vector sums becomes too large to be stored. The
full group $G$ of automorphisms has order $2^9 \cdot 3^4 \cdot 5$ and is absolutely irreducible as matrix group. It has the structure of the central product $2.A_6.Y.H$ extended by $V_4$, where $H$ is the subdirect product of $C_3^2 : V_4$ and $Q_{16}$ amalgamated over the common factor group $V_4$. The largest perfect subgroup of $G$ is $G^{(3)} \cong 2.A_6$ and restricted to this subgroup the natural representation has two absolutely irreducible constituents of degree 4, each with multiplicity 4. The automorphism group leaves invariant a sublattice of $Q_{32}$ of index $3^9$, which decomposes into two copies of a 16-dimensional 6-modular lattice $L$ with $|\text{Aut}(L)| = 2^{10} \cdot 3^6 \cdot 5$ and 960 shortest vectors (of norm 6), cf. [PIN 93].

**Variations.** With minor modifications this algorithm can also be applied to other problems, e.g.:

1) Isometries between two lattices $L$ and $L'$: the fingerprint and vector sums are calculated with respect to a basis $B$ of $L$, then a basis of $L'$ having the same scalar products as the vectors in $B$ is searched.

2) Bravais groups: the single Gram matrix, which is fixed by automorphisms can clearly be replaced by a collection of Gram matrices.

3) Automorphisms of lattices over number fields with a totally positive bilinear or hermitian form taking values in the ring of integers of the number field, cf. e.g. [Sch 92]: these lattices can be dealt with as lattices over $\mathbb{Z}$ with several $\mathbb{Z}$-valued bilinear forms, at least one of which is symmetric and positive definite.

**Bibliography**


