

An Algorithm for the Determination of Space Groups

By Harold Brown*

Abstract. This semiexpository paper presents in detail a numerical formulation of an algorithm for the determination of all n -dimensional, crystallographic space groups. The implementation of this algorithm on a digital computer is also described.

1. Introduction. The n -dimensional space groups are certain discrete groups of rigid motions of Euclidean n -space. Space groups, at least in three and four dimensions, play a central role in crystallography and solid state physics, as these groups describe the microscopic structure of perfect infinite crystals, (see, for example, Buerger [6]).

Our purpose here is to give in detail a numerical formulation of an algorithm for the determination, up to affine equivalence, of all n -dimensional space groups and to describe how this algorithm was implemented on a digital computer. The computational problems that are encountered in implementing the space group algorithm are typical of the difficulties that occur in doing large group-theoretical calculations.**

2. Preliminaries. Let E_n denote Euclidean n -space. For a nonsingular linear transformation a on E_n and a vector α in E_n , we denote by $(a|\alpha)$ the (nonsingular) affine transformation on E_n with action $\eta \rightarrow a\eta + \alpha$. In this notation the product (composition) of two affine transformations, $(a|\alpha)$ and $(b|\beta)$, is $(ab|a\beta + \alpha)$ and the inverse of $(a|\alpha)$ is $(a^{-1}|-a^{-1}\alpha)$. The n -dimensional Euclidean group, \mathcal{E}_n , is the group of all affine transformations $(a|\alpha)$ on E_n with a an orthogonal transformation. In particular, the set of all translations on E_n , $\mathfrak{J}_n = \{(I_n|\alpha)|\alpha \in E_n\}$ where I_n denotes the identity map on E_n , forms a subgroup of E_n .

Let H be a subgroup of \mathcal{E}_n , and let

$$t_H = \{\alpha \in E_n | (I_n|\alpha) \in H\} .$$

H is called an n -dimensional space group if and only if t_H is a discrete subset of E_n containing a basis for E_n and the subgroup $\mathfrak{J}_n \cap H$ of H is of finite index in H .

Example. Consider the infinite plane ornament determined by Fig. 1. Let H be the subgroup of affine transformations in \mathcal{E}_2 carrying this ornament onto itself. Then

$$t_H = \{z_1\omega_1 + z_2\omega_2 | z_i \in \mathbf{Z}\} .$$

(Here \mathbf{Z} denotes the ring of integers.) If a_1 and a_2 are the reflections of E_2 about the lines L_1 and L_2 respectively, then

Received October 30, 1968.

* This work was done, while on leave, as visiting member of the Courant Institute of Mathematical Sciences. The work was supported in part by NSF grant GP-8041.

** The description of many group-theoretical, computational problems is given in Coxeter and Moser [7] and Magnus [9]. Both of these books also contain extensive bibliographies.

$$H = \{(x|\frac{1}{2}\omega_1 + \eta) | x \in \{a_1, a_2a_1\}, \eta \in t_H\} \\ \cup \{(x|\eta) | x \in \{a_2, I_2\}, \eta \in t_H\} .$$

Hence, the index of $\mathfrak{S}_2 \cap H$ in H is 4, and H is a 2-dimensional space group. In general, n -dimensional space groups arise geometrically as subgroups of \mathfrak{E}_n carrying such infinite ornaments onto themselves.

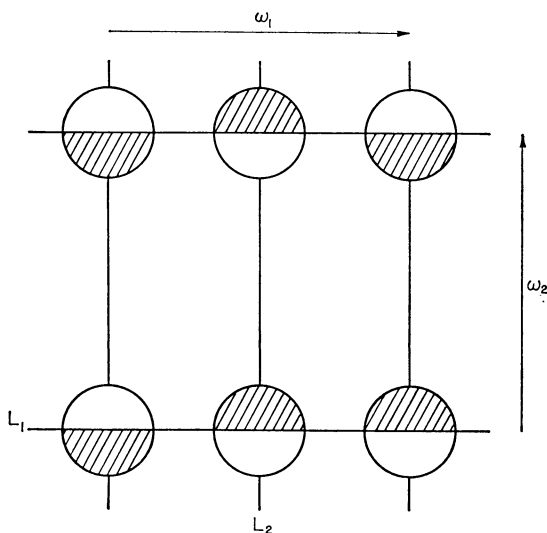


FIGURE 1

Two groups H and H' of affine transformations on E_n are said to be affinely equivalent if and only if there is an affine transformation $(x|\eta)$ on E_n such that $H' = (x|\eta)^{-1}H(x|\eta)$. In his fundamental papers on space groups, Bieberbach [2], [3], [4] proved that up to affine equivalence there are only finitely many space groups in a given dimension and, moreover, that two space groups are affinely equivalent if and only if they are abstractly isomorphic. The n -dimensional space group problem can now be stated as follows: Determine a minimal finite set, \mathfrak{K} , of groups of affine transformations on E_n with the property that each n -dimensional space group is affinely equivalent to one and only one member of \mathfrak{K} .

The solution of this problem for $n = 1, 2, 3$ is well known and was obtained using mainly geometric techniques (see, for example, Jawson [8]). There are, for example, 219 distinct space groups in three dimensions. The key work in the solution of the space group problem for general n was done by Zassenhaus [10], (see also Ascher and Janner [1]). In particular, Zassenhaus gave a completely arithmetical formulation of the problem which we now describe.

An $n \times n$ matrix $a = (a_{ij})$ is said to be unimodular if and only if the entries a_{ij} are integers and $\det(a) = \pm 1$. From this definition it follows that the product of two unimodular matrices or the inverse of a unimodular matrix is again unimodular. Thus, the collection of all $n \times n$ unimodular matrices forms a group which we denote

by $GL(n, \mathbf{Z})$. ($GL(n, \mathbf{Z})$ is often referred to as the general linear group of regular matrices of degree n with integral coefficients.)

Two subgroups G and G' of $GL(n, \mathbf{Z})$ are called conjugate if and only if there exists a unimodular matrix u in $GL(n, \mathbf{Z})$ such that $G' = u^{-1}Gu = \{u^{-1}gu | g \in G\}$. Since $GL(n, \mathbf{Z})$ is a group, the relation of conjugacy is an equivalence relation on the set of all subgroups of $GL(n, \mathbf{Z})$ and, hence, partitions this set into disjoint conjugacy classes. Let \mathfrak{G}_n be a representative set of the conjugacy classes of finite subgroups of $GL(n, \mathbf{Z})$; that is, \mathfrak{G}_n consists of precisely one member from each of the conjugacy classes of finite subgroups of $GL(n, \mathbf{Z})$. Bieberbach [4] showed that \mathfrak{G}_n is a finite set for any n . For $n = 1, 2, 3$, \mathfrak{G}_n has 2, 13, 73 elements respectively, and Zassenhaus, Neubüser and the author [5] have recently proved that \mathfrak{G}_4 has 710 elements. The orders of the groups in \mathfrak{G}_4 range from 1 to 1152. An important role was played by the computer in our determination of \mathfrak{G}_4 .

Let G be a finite $n \times n$ unimodular group in \mathfrak{G}_n , say $G = \{a, b, c, \dots\}$, and let N_G denote the normalizer of G in $GL(n, \mathbf{Z})$; that is, $N_G = \{x \in GL(n, \mathbf{Z}) | x^{-1}Gx = G\}$. A set of $n \times 1$ real vectors, $\mathfrak{J} = \{\tau_a, \tau_b, \tau_c, \dots\}$, one vector associated with each element of G , is called a *vector system* for G if and only if for any f, g in G ,

$$(2.1) \quad \tau_{fg} \equiv \tau_f + f\tau_g \pmod{\mathbf{Z}}.$$

Here the congruence means a component-wise congruence modulo \mathbf{Z} .

Example. For any fixed $n \times 1$ real vector η , let

$$\mathfrak{J}_\eta = \{\nu_g = (I_n - g)\eta | g \in G\}.$$

Then,

$$\nu_{fg} = (I_n - fg)\eta = (I_n - f)\eta + (f - fg)\eta = \nu_f + f\nu_g.$$

Hence, \mathfrak{J}_η is a vector system for G .

Let $\mathfrak{J} = \{\tau_g\}$ and $\mathfrak{J}' = \{\tau'_g\}$ be two vector systems for G . \mathfrak{J} and \mathfrak{J}' are called equivalent if and only if there exist a unimodular matrix x in N_G and an $n \times 1$ real vector η such that

$$(2.2) \quad \tau'_g \equiv x\tau_{x^{-1}gx} + (I_n - g)\eta \pmod{\Sigma}$$

for every g in G . From the property that N_G is a group, it follows that this relation is an equivalence relation on the set of all vector systems for G and, therefore, partitions this set into equivalence classes. Let \mathfrak{U}_G be a representative set for these equivalence classes, and for each $\mathfrak{J} = \{\tau_g\}$ in \mathfrak{U}_G , let $H(G, \mathfrak{J}) = \{(g | \tau_g + \gamma)\}$ where g runs over G and γ runs over all $n \times 1$ integral vectors. Zassenhaus [10] showed that each $H(G, \mathfrak{J})$ is affinely equivalent to an n -dimensional space group and that any n -dimensional space group is affinely equivalent to one and only one of the groups $H(G, \mathfrak{J})$ for some G in \mathfrak{G}_n and \mathfrak{J} in \mathfrak{U}_G . Thus, assuming that we know \mathfrak{G}_n , the space group problem can be reformulated as follows: For each G in \mathfrak{G}_n , determine \mathfrak{U}_G .

3. The Algorithm, Part I. Let W be any finite multiplicative group with identity element e . If the order of W is much over 100, it is not very practical to describe W by its multiplication table. An efficient method, particularly on a computer, for describing W is via a generator and relator presentation. A subset $\{w_1, \dots, w_k\}$ of

W is called a set of generators for W if each element in W is expressible as a (finite) product of the w_i ; and a finite set of words (formal products) in the generators, $R_1 = w_{a(1,1)}w_{a(2,1)} \cdots w_{a(t(1),1)}$, \cdots , $R_v = w_{a(1,v)}w_{a(2,v)} \cdots w_{a(t(v),v)}$, is called a set of defining relators for W with respect to the set of generators $\{w_i\}$ if the collection of all algebraic equations satisfied by the w_i consists precisely of those equations that are algebraic consequences of the particular equations $R_1 = e, \cdots, R_v = e$. Such a set of generators and defining relators for W is called a (finite) presentation for W .***

Examples. (a) The cyclic group of order m has a presentation of the form $\{w_1\}; R_1 = w_1 \cdots w_1$ (m factors).

(b) Let S be a square in E_2 with center at 0 (see Fig. 2). Let W be the group consisting of all 2×2 matrices relative to the basis $\{v_1, v_2\}$ carrying S rigidly onto itself. Abstractly, W is the group of order eight of symmetries of the square.

$$\left\{ w_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, w_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\};$$

$R_1 = w_1 w_1 w_1 w_1, R_2 = w_2 w_2, R_3 = w_1 w_2 w_1 w_2$ is a presentation for W .

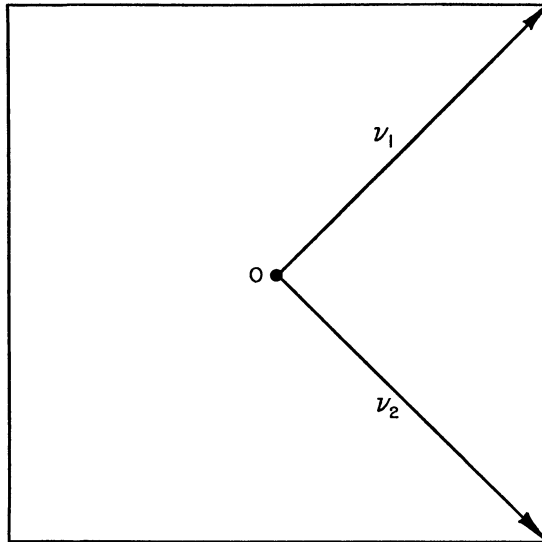


FIGURE 2

Choose a G in \mathfrak{S}_n . Since G is finite, G has a finite presentation, say $\{g_1, \cdots, g_k\}; R_1, \cdots, R_r$. Let $\mathfrak{J} = \{\tau_\theta\}$ be a vector system for G . By repeatedly applying equation (2.1), the defining equation for vector systems, we can compute modulo \mathbf{Z} any vector in \mathfrak{J} from the vectors $\tau_{\theta_1}, \cdots, \tau_{\theta_k}$ associated with the set of generators $\{g_i\}$ of G . Thus, \mathfrak{J} is completely determined modulo \mathbf{Z} by its subset $\{\tau_{\theta_1}, \cdots, \tau_{\theta_k}\}$.

If $\{\eta_{\theta_1}, \cdots, \eta_{\theta_k}\}$ is any ordered set of k $n \times 1$ real vectors, we can attempt to construct a vector system for G by recursive application of the rule $\eta_{\theta_i \theta_t} = \eta_{\theta_i} + g_i \eta_{\theta_t}$ and the associative law in G . For example, if $g = g_1 g_2 g_1 g_2$,

*** See, for example, Coxeter and Moser [7] for a more complete discussion of presentations.

$$\begin{aligned} \eta_{\sigma} &= \eta_{\sigma_1(\sigma_2\sigma_1\sigma_2)} = \eta_{\sigma_1} + g_1\eta_{\sigma_2(\sigma_1\sigma_2)} \\ &= \eta_{\sigma_1} + g_1(\eta_{\sigma_2} + g_2\eta_{\sigma_1\sigma_2}) \\ &= \eta_{\sigma_1} + g_1(\eta_{\sigma_2} + g_2(\eta_{\sigma_1} + g_1\eta_{\sigma_2})) \\ &= (I_n + g_1g_2)\eta_{\sigma_1} + (g_1 + g_1g_2g_1)\eta_{\sigma_2}. \end{aligned}$$

The difficulty here is that the so constructed vectors may not be well defined since a given element in G has, in general, many distinct representations as a product of the generators $\{g_i\}$. However, if the η_{σ_i} satisfy the condition that when the vectors $\eta_{R_1}, \dots, \eta_{R_v}$ corresponding to the defining relators for G are constructed as above we have

$$\eta_{R_j} \equiv 0 \pmod{\mathbf{Z}}, \quad j = 1, \dots, v,$$

then $\{\eta_{\sigma_1}, \dots, \eta_{\sigma_k}\}$ does determine a well-defined vector system for G , [10, pp. 123–125]. (Here, of course, the relator words R_j are treated as formal products in the symbols g_1, \dots, g_k when performing these constructions.) Also, if $\{\tau_{\sigma}\}$ is a vector system for G , then by the defining Eq. (2.1),

$$\tau_{I_n} = \tau_{I_n I_n} \equiv \tau_{I_n} + \tau_{I_n} \pmod{\mathbf{Z}}.$$

That is, $\tau_{I_n} \equiv 0 \pmod{\mathbf{Z}}$, and hence if τ_{R_j} is formally evaluated as above, $\tau_{R_j} \equiv 0 \pmod{\mathbf{Z}}, j = 1, \dots, v$. Thus, a necessary and sufficient condition for an ordered set of k $n \times 1$ real vectors $\{\eta_{\sigma_1}, \dots, \eta_{\sigma_k}\}$ to determine a vector system for G is that when η_{R_j} is formally evaluated using the rule $\eta_{\sigma_i\sigma_l} = \eta_{\sigma_i} + g_i \eta_{\sigma_l}$, then $\eta_{R_j} \equiv 0 \pmod{\mathbf{Z}}, j = 1, \dots, v$. We note that as in the above example, the expansion of η_{R_j} can be expressed as a linear combination of the η_{σ_i} with $n \times n$ integral matrices as coefficients, say

$$\eta_{R_j} = \sum_{i=1}^k A(j, i)\eta_{\sigma_i}, \quad j = 1, \dots, v.$$

Moreover, the coefficients $A(j, i)$ can be computed algorithmically from the given presentation for G . Therefore, if we form the $nv \times nk$ integral relation matrix $A = (A(j, i)), 1 \leq j \leq v, 1 \leq i \leq k$, and the $nk \times 1$ vector

$$\eta = \begin{bmatrix} \eta_{\sigma_1} \\ \eta_{\sigma_2} \\ \vdots \\ \eta_{\sigma_k} \end{bmatrix},$$

a necessary and sufficient condition that $\{\eta_{\sigma_i}\}$ determines a vector system for G is $A\eta \equiv 0 \pmod{\mathbf{Z}}$.

The first step of our algorithm is to construct the relation matrix $A = (A(j, i))$ from the given presentation for $G, \{g_1, \dots, g_k\}; R_1 = g_{a(1,1)} \dots g_{a(t(1),1)}, \dots, R_r = g_{a(1,r)} \dots g_{a(t(r),r)}$. (Here we may assume that the generators g_i are all distinct from the identity matrix I_n .) This computation is performed on the computer using the following procedure:

P_0, P_1 and $A(j, i), 1 \leq j \leq v, 1 \leq i \leq k, n \times n$ arrays.

- (1) Set $j = 1$.
 - (2) Set $A(j, a(1, j)) = I_n$ and set $A(j, t) = 0_n$ for $t \neq a(1, j)$. (Here 0_n denotes the $n \times n$ zero matrix.)
 - (3) $g_{a(1, j)} + A(j, a(2, j)) \rightarrow A(j, a(2, j))$.
 - (4) If $t(j) = 2$, go to step 9, otherwise set $y = 0, z = 1, x = 3$.
 - (5) $g_{a(1, j)} \rightarrow P_y$.
 - (6) $P_y \cdot g_{a(x-1, j)} \rightarrow P_z$.
 - (7) $P_z + A(j, a(x, j)) \rightarrow A(j, a(x, j))$.
 - (8) If $x < t(j), x + 1 \rightarrow x, y \leftrightarrow z$ and go to step 6.
 - (9) If $j < v, j + 1 \rightarrow j$ and go to step 2.
- END.

The next task is to find all the solutions of the equation $A\eta \equiv 0 \pmod{\mathbf{Z}}$. To solve this equation, we use a unimodular diagonalization procedure. This procedure is quite similar to Gaussian elimination with the exception that the only allowable elementary row and column operations are those which are invertible over the integers; that is, those whose corresponding elementary matrices are unimodular. Hence, we are allowed the following elementary operations:

- (a) Interchange two rows (columns).
- (b) Add an integral multiple of a row (column) to a distinct row (column).
- (c) Multiply a row (column) by -1 .

The implementation of this diagonalization procedure that is used on the computer goes as follows:

(1) Search $A = (a_{ij})$ for an entry of minimal nonzero magnitude, say a_{xy} . If in the process of this search an entry of magnitude 1, say a_{st} , is encountered, set $x = s, y = t$ and go immediately to step 2. If no entry of minimal nonzero magnitude exists, that is, if A is a zero matrix, go to END.

(2) Use a_{xy} as a row pivot. That is, for $1 \leq i \leq nv, i \neq x$, compute the integer q_i satisfying

$$a_{iy} = q_i a_{xy} + r_i, \quad |r_i| < |a_{xy}|,$$

and add $-q_i$ times the x th row of A to the i th row of A . Similarly, use a_{xy} as a column pivot.

(3) If any remainder r_i is not zero, go to step 1. Otherwise, replace a_{xy} by $|a_{xy}|$, shift a_{xy} to the 1-1 position, truncate the first row and the first column of A and then go to step 1. If such a truncation is impossible, that is, if A has only one row or one column, go to END.

END.

That this procedure must terminate after finitely many steps is well known from number theory, for example, Euclid's algorithm; for as the program loops through steps 1 and 2, the magnitudes of the pivot elements a_{xy} form a strictly decreasing sequence of positive integers which must eventually terminate.

At the termination of this procedure, the matrix A has been transformed into an $nv \times nk$ diagonal matrix, say

$$D = \text{diag}(d_1, \dots, d_p, 0, \dots, 0)$$

where the d_i are positive integers. Moreover, $D = SAQ$ where S and Q are unimodular matrices. Since both the matrix Q and its inverse, Q^{-1} , are needed later in the pro-

gram, as we perform a column operation on A , we also perform the same column operation on the columns of an $nk \times nk$ matrix Q and perform the inverse operation on the rows of an $nk \times nk$ matrix Q^{-1} where initially $Q = Q^{-1} = I_{nk}$.

It is now easy to solve the equation $A\eta \equiv 0 \pmod{\mathbf{Z}}$; for an $nk \times 1$ real vector η satisfies this equation if and only if $\eta = Q\chi$ with χ of the form

$$\chi = \begin{bmatrix} z_1/d_1 \\ \vdots \\ z_p/d_p \\ * \\ \vdots \\ * \end{bmatrix}$$

where the z_i are integers and the $*$'s are any real numbers. This result is an immediate consequence of the fact that both S and Q are integral matrices with integral inverses.

Even in the case $n = 4$, the matrices A that occur in the algorithm can be fairly large, for example 80×32 . Therefore, the unimodular diagonalization routine used must be efficient. From empirical evidence it seems that the procedure just described, that is, always choosing the pivot as an entry of minimal nonzero magnitude relative to the entire matrix, is statistically more efficient than choosing the pivot from a predetermined row or column. An interesting experimental routine in which the pivot element was chosen essentially randomly was also tested. This random routine was even faster than the routine used; however, I was unable to prove that the random routine even terminated in general.

4. The Algorithm, Part II. Having found all vector systems for G , we must now determine a representative set for the equivalence classes of these vector systems. This determination is done in two stages. Firstly, if in Eq. (2.2) we require that $x = I_n$, we obtain a more restrictive equivalence relation on the set of vector systems for G . Namely, $\mathfrak{J} = \{\tau_g\}$ is *strongly* equivalent to $\mathfrak{J}' = \{\tau_{g'}\}$ if and only if there exists an $n \times 1$ real vector η such that

$$(4.1) \quad \tau_{g'} \equiv \tau_g + (I_n - g)\eta \pmod{\mathbf{Z}}$$

for all g in G . It is clear that if \mathfrak{J} and \mathfrak{J}' are strongly equivalent, then they are also equivalent in the original sense. Thus, a representative set of the strong equivalence classes of vector systems for G contains a representative set of the equivalence classes of vector systems for G . As we shall now show, there are only finitely many strong equivalence classes of vector systems for G and, moreover, a representative set for these strong equivalence classes can be easily determined directly from the diagonal form D of the relation matrix A .

If $\mathfrak{J} = \{\tau_g\}$ and $\mathfrak{J}' = \{\tau_{g'}\}$ are two vector systems for G , then, from the defining Eq. (2.1), it follows by a direct computation that \mathfrak{J} is equivalent (strongly equivalent) to \mathfrak{J}' if and only if Eq. (2.2) (Eq. (4.1)) is satisfied for the generators $\{g_i\}$ of G . Hence, in determining the representative sets of vector systems for G , we can work directly with the $nk \times 1$ vectors

$$\tau = \begin{bmatrix} \tau_{\theta_1} \\ \vdots \\ \tau_{\theta_k} \end{bmatrix}$$

which satisfy $A\tau \equiv 0 \pmod{\mathbf{Z}}$. In terms of the associated $nk \times 1$ vectors τ and τ' of \mathfrak{J} and \mathfrak{J}' respectively, \mathfrak{J} is strongly equivalent to \mathfrak{J}' if and only if

$$(4.2) \quad \tau' - \tau = \begin{bmatrix} (I_n - g_1)\eta \\ \vdots \\ (I_n - g_k)\eta \end{bmatrix} + \mu$$

for some $n \times 1$ real vector η and some $nk \times 1$ integral vector μ .

For any $n \times 1$ real vector η , consider the vector system $\{\nu_g = (I_n - g)\eta \mid g \in G\}$ given in the example in Section 2. Since η was an arbitrary real vector, $\{\lambda\nu_g\}$ must also be a vector system for G for any real λ . Thus, if

$$\nu = \begin{bmatrix} \nu_{\theta_1} \\ \vdots \\ \nu_{\theta_k} \end{bmatrix},$$

then $A(\lambda\nu) = \lambda(A\nu) \equiv 0 \pmod{\mathbf{Z}}$ for any real λ . Hence, $A\nu$ is the $nk \times 1$ zero vector, $0_{nk \times 1}$. From this result and equation (4.2) it follows that if \mathfrak{J} is strongly equivalent to \mathfrak{J}' , then $A(\tau' - \tau - \mu) = 0_{nk \times 1}$ for some $nk \times 1$ integral vector μ .

Conversely, if $A(\tau' - \tau - \mu) = 0_{nk \times 1}$ for some $nk \times 1$ integral vector μ , then for any real λ , the $nk \times 1$ vector $\lambda(\tau' - \tau - \mu)$ determines a vector system for G . Let $\{\nu_g\}$ be the vector system for G determined by $\tau' - \tau - \mu$. For any real λ and any g, h in G , we have

$$\lambda\nu_{gh} \equiv \lambda\nu_g + g\lambda\nu_h \pmod{\mathbf{Z}};$$

that is, $\lambda(\nu_{gh} - \nu_g - g\nu_h)$ is integral. Hence, $\nu_{gh} = \nu_g + g\nu_h$ for any g, h in G . Let

$$\eta = (1/d) \sum_{g \in G} \nu_g$$

where d is the order of G . Then,

$$(4.3) \quad (1/d) \sum_{g \in G} (\nu_{gig} - \nu_{gi} - g_i\nu_g) = \eta - \nu_{g_i} - g_i\eta = 0,$$

$i = 1, \dots, k$; for as g runs through G , so does $g_i g$. From (4.3) it follows that

$$\tau' = \tau + \begin{bmatrix} (I_n - g_1)\eta \\ \vdots \\ (I_n - g_k)\eta \end{bmatrix} + \mu;$$

that is, \mathfrak{J} is strongly equivalent to \mathfrak{J}' . Thus, a necessary and sufficient condition for \mathfrak{J} and \mathfrak{J}' to be strongly equivalent is that $A(\tau' - \tau - \mu) = 0_{nk \times 1}$ for some $nk \times 1$ integral vector μ .

From the diagonal form D of the relation matrix A , we see that $A\nu = 0_{nk \times 1}$ for a real $nk \times 1$ vector ν if and only if $\nu = Q\chi$ where the first p entries of χ are 0. Hence, since Q^{-1} is an integral matrix, we have that \mathfrak{J} is strongly equivalent to \mathfrak{J}' if and only if $\tau' - \tau = Q\gamma$ with γ of the form

$$\gamma = \begin{bmatrix} y_1 \\ \vdots \\ y_p \\ * \\ \vdots \\ * \end{bmatrix}, \quad y_i \in \mathbf{Z},$$

with the *'s any real numbers.

Form the $q = d_1 \cdots d_p$ $nk \times 1$ vectors $QP_{e_1 \dots e_p}$ where

$$P_{e_1 \dots e_p} = \begin{bmatrix} e_1/d_1 \\ \vdots \\ e_p/d_p \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad 0 \leq e_i < d_i.$$

The vector systems for G determined by these q vectors form a representative set for the strong equivalence classes of vector systems for G . For if $\mathfrak{J} = \{\tau_\sigma\}$ is any vector system for G , say

$$\tau = \begin{bmatrix} \tau_{\sigma 1} \\ \vdots \\ \tau_{\sigma k} \end{bmatrix} = Q \begin{bmatrix} t_1/d_1 \\ \vdots \\ t_p/d_p \\ * \\ \vdots \\ * \end{bmatrix}, \quad t_i \in \mathbf{Z},$$

let e_i' be the unique nonnegative integer satisfying

$$t_i = q_i d_i + e_i', \quad q_i \in \mathbf{Z}, \quad 0 \leq e_i' < d_i, \quad i = 1, \dots, p.$$

Then

$$\tau - QP_{e_1' \dots e_p'} = Q \begin{bmatrix} q_1 \\ \vdots \\ q_p \\ * \\ \vdots \\ * \end{bmatrix};$$

that is, \mathfrak{J} is strongly equivalent to the vector system determined by $QP_{e_1' \dots e_p'}$.

Moreover, if $QP_{e_1 \dots e_p} - QP_{e'_1 \dots e'_p}$ is of the form $Q\gamma$ where the first p entries of γ are integral, we must have $e_i = e'_i, i = 1, \dots, p$.

After transforming the relation matrix A into its diagonal form D , the computer program determines from the diagonal entries of D the q $nk \times 1$ vectors $P_{e_1 \dots e_p}$ in lexicographical order, say

$$P_1' = P_{0 \dots 0}, P_2' = P_{0 \dots 01}, \dots, P_q' = P_{(d_1-1) \dots (d_p-1)},$$

and outputs the vectors QP_i' in sequential order. These latter vectors are not saved in memory as in the balance of the computation we work directly with the vectors $\{P_{e_1 \dots e_p}\}$.

5. The Algorithm, Part III. From the set of vectors $\{P_{e_1 \dots e_p}\}$, we must now select a subset which determines a representative set for the equivalence classes of vector systems for G .

Let $\{\tau_\theta\}$ be the vector system for G determined by $QP_{e_1 \dots e_p}$, and let x be an element of N_G ; that is, x is an $n \times n$ unimodular matrix and $x^{-1}Gx = G$. Then, the indexed collection of vectors $\{\tau'_\theta = x\tau_{x^{-1}g\theta} | g \in G\}$ is again a vector system for G ; for

$$\begin{aligned} \tau'_{\theta h} &= x\tau_{x^{-1}g\theta h} = x\tau_{(x^{-1}gx)(x^{-1}hx)} \\ &\equiv x(\tau_{x^{-1}gx} + x^{-1}g\tau_{x^{-1}hx}) \\ &\equiv x\tau_{x^{-1}gx} + g(x\tau_{x^{-1}hx}) \\ &\equiv \tau'_\theta + g\tau'_h \pmod{\mathbf{Z}}. \end{aligned}$$

Since $\{\tau'_\theta\}$ is a vector system for G , it must be strongly equivalent to precisely one of the vector systems determined by the q vectors $\{QP_{e_1 \dots e_p}\}$. Say $\{\tau'_\theta\}$ is strongly equivalent to the vector system determined by $QP_{e'_1 \dots e'_p}$. In this manner the matrix x determines a mapping π_x from the finite set $\mathcal{O} = \{P_{e_1 \dots e_p} | 0 \leq e_i < d_i\}$ into itself defined by $\pi_x: P_{e_1 \dots e_p} \rightarrow P_{e'_1 \dots e'_p}$. Now, if $\{\bar{\tau}_\theta\}$ is the vector system for G determined by $QP_{\bar{e}_1 \dots \bar{e}_p}$ and $\pi_x(P_{e_1 \dots e_p}) = P_{\bar{e}_1 \dots \bar{e}_p}$, then $\{\tau'_\theta = x\tau_{x^{-1}g\theta}\}$ and $\{\bar{\tau}'_\theta = x\bar{\tau}_{x^{-1}g\theta}\}$ must be strongly equivalent vector systems. Therefore, there is an $n \times 1$ real vector η such that

$$x\tau_{x^{-1}g\theta} \equiv x\bar{\tau}_{x^{-1}g\theta} + (I_n - g)\eta \pmod{\mathbf{Z}}$$

for all g in G . Thus,

$$\tau_{x^{-1}g\theta} \equiv \bar{\tau}_{x^{-1}g\theta} + (I_n - x^{-1}gx)(x^{-1}\eta) \pmod{\mathbf{Z}},$$

and, since as g runs through G so does $x^{-1}gx$, $\{\tau_\theta\}$ and $\{\bar{\tau}_\theta\}$ are strongly equivalent. But, because the vectors $\{QP_{e_1 \dots e_p}\}$ determine a representative set of the strong equivalence classes of vector systems for G , $\{\tau_\theta\}$ is strongly equivalent to $\{\bar{\tau}_\theta\}$ if and only if $P_{e_1 \dots e_p} = P_{\bar{e}_1 \dots \bar{e}_p}$. Hence, the map π_x is a one-to-one map of \mathcal{O} onto \mathcal{O} ; that is, π_x is a permutation of \mathcal{O} for each x in N_G . The set of these permutations, $R = \{\pi_x | x \in N_G\}$, forms a permutation group on \mathcal{O} of order at most $q!$.

Let $\mathcal{O}_1, \dots, \mathcal{O}_w$ be the orbits of \mathcal{O} with respect to R . That is, the \mathcal{O}_i are the distinct equivalence classes of \mathcal{O} with respect to the relation $P_{e_1 \dots e_p}$ is equivalent to $P_{e'_1 \dots e'_p}$ if and only if there is an x in N_G such that $\pi_x(P_{e_1 \dots e_p}) = P_{e'_1 \dots e'_p}$. Choose

exactly one member, say P_i , from each orbit \mathcal{O}_i . Then, the set of vector systems for G determined by the vectors $QP_i, i = 1, \dots, w$, forms a representative set for the equivalence classes of vector systems for G . For let $\mathfrak{J} = \{\tau_\theta\}$ be any vector system for G . By the choice of the q vectors $QP_{e_1 \dots e_p}$, \mathfrak{J} is strongly equivalent to one and only one of the q vector systems determined by these vectors. Say \mathfrak{J} is strongly equivalent to the vector system determined by

$$QP_{e_1 \dots e_p'} = \begin{bmatrix} \tau'_{\theta_1} \\ \vdots \\ \tau'_{\theta_k} \end{bmatrix}.$$

Hence, there is an $n \times 1$ real vector η such that

$$(5.1) \quad \tau_{\theta_i} \equiv \tau'_{\theta_i} + (I_n - g_i)\eta \pmod{\mathbf{Z}}, \quad i = 1, \dots, k.$$

Since the orbits of \mathcal{O} with respect to R partition the set \mathcal{O} , $P_{e_1 \dots e_p'}$ lies in one and only one orbit, say \mathcal{O}_c . This means that there is a π_x in R such that $\pi_x(P_{e_1 \dots e_p'}) = P_c$; that is, the vector system for G determined by $\{\tau''_{\theta_i} = x\tau'_{x^{-1}g_ix}\mid i = 1, \dots, k\}$ is strongly equivalent to the vector system determined by QP_c . Substituting $x^{-1}\tau''_{\theta_ix^{-1}}$ for τ'_{θ_i} in Eq. (5.1) yields

$$\tau_{\theta_i} \equiv x^{-1}\tau''_{\theta_ix^{-1}} + (I_n - g_i)\eta \pmod{\mathbf{Z}}, \quad i = 1, \dots, k.$$

Thus, since x^{-1} is also in N_G , \mathfrak{J} is equivalent to $\{\tau_{\theta''}\}$ which, in turn, is equivalent to the vector system determined by QP_c . Also, since the vector P_c was uniquely determined by \mathfrak{J} , \mathfrak{J} is not equivalent to any of the vector systems determined by $QP_j, j \neq c$.

Therefore, the final computational task is to determine the orbits of the set $\mathcal{O} = \{P_{e_1 \dots e_p} \mid 0 \leq e_i < d_i\}$ with respect to the group of permutations R induced by N_G . We note that from Eqs. (2.2) and (4.1) it follows that the strong equivalence class of vector systems for G associated with the vector $QP_{0 \dots 0} = 0_{nk \times 1}$ is also an equivalence class. Thus, if the number of strong equivalence classes of vector systems for some G in \mathfrak{G}_n does not exceed 2, then this orbit computation can be entirely bypassed.

For an x in N_G , the elements $x^{-1}g_jx, j = 1, \dots, k$, are again in G and, hence, can be expressed as products of the generators g_i . Say,

$$(5.2) \quad x^{-1}g_jx = g_{a(x,1,j)} \cdots g_{a(x,s(x,j),j)}, \quad j = 1, \dots, k.$$

Let $\{\tau_\theta\}$ be the vector system for G determined by $QP_{e_1 \dots e_p}$. In exactly the same manner as for the defining relators for G in Section 3, from the right-hand side of (5.2) we can formally expand the vectors $\tau_{x^{-1}g_jx}$ as linear combinations of the τ_{θ_i} with $n \times n$ integral matrices as coefficients. Say,

$$\tau_{x^{-1}g_jx} = \sum_{i=1}^k M(x, j, i)\tau_{\theta_i}, \quad j = 1, \dots, k.$$

Form the $nk \times nk$ matrix $M_x = (xM(x, j, i)), 1 \leq j, i \leq k$. Then, the vector system $\{\tau_{\theta'} = x\tau_{x^{-1}g_jx}\}$ is strongly equivalent to the vector system \mathfrak{J}'' determined by the $nk \times 1$ vector

$$M_x Q P_{e_1 \dots e_p} = Q \begin{bmatrix} t_1/d_1 \\ \vdots \\ t_p/d_p \\ * \\ \vdots \\ * \end{bmatrix}, \quad t_i \in \mathbf{Z}.$$

Moreover, the vector system \mathfrak{Z}'' is strongly equivalent to the vector system determined by $Q P_{e'_1 \dots e'_p}$ where the e'_i are the uniquely determined integers satisfying

$$e'_i \equiv t_i \pmod{d_i} \quad \text{and} \quad 0 \leq e'_i < d_i, \quad i = 1, \dots, p$$

for

$$M_x Q P_{e_1 \dots e_p} - Q P_{e'_1 \dots e'_p} = Q \begin{bmatrix} z_1 \\ \vdots \\ z_p \\ * \\ \vdots \\ * \end{bmatrix},$$

where the z_i are integers. Thus, the permutation π_x is given by $\pi_x: P_{e_1 \dots e_p} \rightarrow P_{e'_1 \dots e'_p}$.

From the input $\{g_1, \dots, g_k\}, x \in N_G$ and the sequences of indices, $(a(x, 1, j), \dots, a(x, s(x, j), j)), j = 1, \dots, k$, of (5.2), the program determines the permutation π_x using the following procedure:

(1) Using exactly the same routine as that used to determine the relation matrix A in Section 3, compute the k^2 $n \times n$ matrices $M(x, j, i), 1 \leq j, i \leq k$.

(2) Form the matrix $M = Q^{-1}(xM(x, j, i))Q$.

(3) For $r = 1, \dots, q$: Compute the first p entries of the vector MP_r' , say $t_1/d_1, \dots, t_p/d_p$. Determine the unique integers e'_m satisfying $e'_m \equiv t_m \pmod{d_m}$ and $0 \leq e'_m < d_m, m = 1, \dots, p$. Determine by direct comparison with the P_s' the index $z(r, x)$ satisfying $P'_{z(r, x)} = P_{e'_1 \dots e'_p}$. (Here P_1', \dots, P_q' is the lexicographical ordering of the $P_{e_1 \dots e_p}$ given in Section 4.)

(4) Form the $1 \times q$ vector $p_x = (z(1, x), \dots, z(q, x))$.

The permutation π_x is then given by $\pi_x: P_s' \rightarrow P'_{z(s, x)}$.

Although it may be that the group $N_G = \{x \in \text{GL}(n, \mathbf{Z}) | x^{-1}Gx = G\}$ is infinite, N_G always has a finite set of generators, [10, pp. 130–131]. Moreover, we note that if $x \in G \cap N_G$, then for any vector system $\{\tau_g\}$,

$$\begin{aligned} (5.3) \quad x\tau_{x^{-1}gx} &\equiv x(\tau_{x^{-1}} + x^{-1}\tau_g + x^{-1}g\tau_x) \\ &\equiv x\tau_{x^{-1}} + \tau_g + g\tau_x. \end{aligned}$$

Since $\tau_{I_n} \equiv 0 \pmod{\mathbf{Z}}$ implies that $\tau_x \equiv -x\tau_{x^{-1}} \pmod{\mathbf{Z}}$, it follows from (5.3) that

$$x\tau_{x^{-1}gx} \equiv \tau_g + (I_n - g)(-\tau_x) \pmod{\mathbf{Z}}$$

for every g in G . Thus, $\{\tau'_g = x\tau_{x^{-1}gx}\}$ is strongly equivalent to $\{\tau_g\}$, and π_x is the identity permutation. Hence the generators of N_G lying in G can be ignored in

computing the orbits of \mathcal{O} with respect to the group $R = \{\pi_x | x \in N_G\}$.

Let x_1, \dots, x_t be elements of N_G such that $\{G, x_1, \dots, x_t\}$ generates N_G , and let $p_{x_j} = (z(1, x_j), \dots, z(q, x_j)), j = 1, \dots, t$, be the associated permutation vectors computed as above. The order of the permutation group R cannot exceed $q!$; however, even in the case $n = 4$, q can be greater than 700. Therefore, the computation of the orbits of \mathcal{O} with respect to R must be done without actually constructing the group R .

The computer routine used to perform the orbit calculation goes as follows:

$V = (v_r)$ a $1 \times q$ vector and SW a switch.

- (1) $0 \rightarrow v_r, r = 1, \dots, q$.
 - (2) Replace the first zero entry in V , say v_{i_0} , by 1, set $s = i_0 + 1$ and set $i = i_0$.
 - (3) Turn SW off.
 - (4) $1 \rightarrow j$.
 - (5) If $v_{z(i, x_j)} = 0$, turn SW on and set $v_{z(i, x_j)} = 1$.
 - (6) If $j < t, j + 1 \rightarrow j$ and go to step 5.
 - (7) If $i = q$, go to step 11.
 - (8) $i + 1 \rightarrow i$.
 - (9) If $v_i = 1$, go to step 4.
 - (10) If $i < q, i + 1 \rightarrow i$ and go to step 9.
 - (11) If SW is off, go to step 15.
 - (12) $s \rightarrow i$.
 - (13) If $v_i = 1$, go to step 3.
 - (14) If $i < q, i + 1 \rightarrow i$ and go to step 13.
 - (15) Output the subset S of $\{1, \dots, q\}$ consisting of those indices r such that $v_r = 1$, and replace each v_r such that $v_r = 1$ by 2.
 - (16) If any $v_r = 0$ for $s \leq r \leq q$, go to step 2.
- END.

If S_1, \dots, S_m are the subsets of $\{1, \dots, q\}$ outputted by the above routine, then the sets $\mathcal{O}_i = \{P_j' | j \in S_i\}, i = 1, \dots, m$, are the orbits of the set \mathcal{O} with respect to the permutation group R . Thus, if we select precisely one index, say $c(j)$, from each of the subsets S_j and take the m vectors

$$QP'_{c(j)} = \begin{bmatrix} \tau_{\theta_1}^{(j)} \\ \vdots \\ \tau_{\theta_k}^{(j)} \end{bmatrix}, \quad j = 1, \dots, m,$$

then these m vectors determine a representative set for the equivalence classes of vector systems for G . The groups of affine transformations $H(G, \mathfrak{J}_1), \dots, H(G, \mathfrak{J}_m)$ where $H(G, \mathfrak{J}_j)$ is generated by the set $\{(g_i | \tau_{\theta_i}^{(j)}) | 1 \leq i \leq k\} \cup \{(I_n | \gamma) | \gamma \text{ an } n \times 1 \text{ integral vector}\}, j = 1, \dots, m$, represent precisely the nonisomorphic n -dimensional space groups associated with G . Hence, if for each unimodular group G in the finite set \mathcal{G}_n we determine via the above algorithm the finitely many associated groups of affine transformations, $H(G, \mathfrak{J}_j)$, then the totality of these groups, $\mathfrak{C} = \{H(G, \mathfrak{J}_j) | G \in \mathcal{G}_n\}$, forms a minimal finite set of groups of affine transformations with the property that any n -dimensional space group is affinely equivalent to one and only one member of \mathfrak{C} .

This space group algorithm has been programed for an IBM 7094 in the assembly language, and the program has been run at the Ohio State University Computer Center for most of the 710 four-dimensional unimodular groups. The running time per unimodular group is between one and fifteen seconds.

6. Example. Let G be the 2-dimensional unimodular group with presentation

$$\left\{ g_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, g_2 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \right\}; \quad R_1 = g_1g_1, R_2 = g_2g_2, R_3 = g_1g_2g_1g_2.$$

Abstractly, G is the Klein four group. G arises geometrically as the group of symmetries of a rectangle.

The 2×2 coefficient matrices $A(j,i)$ are

$$\begin{aligned} A(1, 1) &= (I_2 + g_1) & A(1, 2) &= 0_2 \\ A(2, 1) &= 0_2 & A(2, 2) &= (I_2 + g_2) \\ A(3, 1) &= (I_2 + g_1g_2) & A(3, 2) &= (g_1 + g_1g_2g_1). \end{aligned}$$

The relation matrix A is

$$A = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & -2 \end{bmatrix}.$$

A has the diagonal form $D = \text{diag}(2, 2, 0, 0)$, and the column transformation matrix is

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{with } Q^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

There are four strong equivalence classes of vector systems for G . These are represented by the vectors

$$\begin{aligned} P_{0,0} = QP_{0,0} = P_1' &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & P_{0,1} = QP_{0,1} = P_2' &= \begin{bmatrix} 0 \\ \frac{1}{2} \\ 0 \\ 0 \end{bmatrix}, \\ P_{1,0} = QP_{1,0} = P_3' &= \begin{bmatrix} \frac{1}{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}, & P_{1,1} = QP_{1,1} = P_4' &= \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \\ 0 \end{bmatrix}. \end{aligned}$$

N_G is generated by

$$\left\{ G, x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\}$$

with $x^{-1}g_1x = g_1g_2$, $x^{-1}g_2x = g_2$. The 2×2 coefficient matrices associated with π_x are

$$\begin{aligned} M(x, 1, 1) &= I_2 & M(x, 1, 2) &= g_1 \\ M(x, 2, 1) &= 0_2 & M(x, 2, 2) &= I_2, \end{aligned}$$

and

$$M = Q^{-1}(xM(x, j, i))Q = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Also,

$$MP_1' = P_1', MP_2' = P_3', MP_3' = P_2'$$

and

$$MP_4' = P_4'.$$

The 1×4 vector p_x associated with the permutation π_x is $p_x = (1 \ 3 \ 2 \ 4)$.

The orbits of the set $\{1, 2, 3, 4\}$ under the group of permutations induced by p_x are $S_1 = \{1\}$, $S_2 = \{2, 3\}$ and $S_3 = \{4\}$.

Thus, there are three distinct 2-dimensional space groups associated with G . They are represented by the affine transformation groups $H(G, \mathfrak{J}_1)$, $H(G, \mathfrak{J}_2)$, $H(G, \mathfrak{J}_3)$ generated by the sets

$$\begin{aligned} &\left\{ \begin{pmatrix} 1 & 0|0 \\ 0 & -1|0 \end{pmatrix}, \begin{pmatrix} -1 & 0|0 \\ 0 & -1|0 \end{pmatrix}, \begin{pmatrix} 1 & 0|z_1 \\ 0 & 1|z_2 \end{pmatrix} \Big|_{z_i \in \mathbf{Z}} \right\}, \\ &\left\{ \begin{pmatrix} 1 & 0|0 \\ 0 & -1|\frac{1}{2} \end{pmatrix}, \begin{pmatrix} -1 & 0|0 \\ 0 & -1|0 \end{pmatrix}, \begin{pmatrix} 1 & 0|z_1 \\ 0 & 1|z_2 \end{pmatrix} \Big|_{z_i \in \mathbf{Z}} \right\}, \\ &\left\{ \begin{pmatrix} 1 & 0|\frac{1}{2} \\ 0 & -1|\frac{1}{2} \end{pmatrix}, \begin{pmatrix} -1 & 0|0 \\ 0 & -1|0 \end{pmatrix}, \begin{pmatrix} 1 & 0|z_1 \\ 0 & 1|z_2 \end{pmatrix} \Big|_{z_i \in \mathbf{Z}} \right\}, \end{aligned}$$

respectively.

The Ohio State University
Columbus, Ohio 43210

1. E. ASCHER & A. JANNER, "Algebraic aspects of crystallography," *Helv. Phys. Acta.*, v. 38, 1965, pp. 551-572. MR 33 #7420.

2. L. BIEBERBACH, "Über die Bewegungsgruppen der Euklidischen Räume. I," *Math. Ann.*, v. 70, 1910, pp. 297-336.

3. L. BIEBERBACH, "Die Gruppen mit endlichem Fundamentalbereich," *Math. Ann.*, v. 72, 1912, pp. 400-412.

4. L. BIEBERBACH, "Über die Minkowskische Reduktion der positiven quadratischen Formen und die endlichen Gruppen linear ganzzahliger Substitutionen," *Nachr. Akad. Wiss. Göttingen*, 1912, pp. 207-216.

5. H. BROWN, J. NEUBÜSER & H. ZASSENHAUS, "The 4-dimensional finite unimodular groups." (To appear.)
6. M. BUEGER, *Elementary Crystallography*, Wiley, New York, 1956.
7. H. S. M. COXETER & W. O. J. MOSER, *Generators and Relations for Discrete Groups*, 2nd ed., *Ergebnisse der Math. und ihrer Grenzgebiete*, Bd. 14, Springer-Verlag, New York, 1965. MR 30 #4818.
8. M. JAWSON, *An Introduction to Mathematical Crystallography*, American Elsevier, New York, 1965.
9. W. MAGNUS, A. KARRASS & D. SOLITAR, *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations*, *Pure and Appl. Math.*, vol. 13, Interscience, New York, 1966. MR 34 #7617.
10. H. ZASSENHAUS, "Über einen Algorithmus zur Bestimmung der Raumgruppen," *Comment. Math. Helv.*, v. 21, 1948, pp. 117–141. MR 9, 491.