Acta Cryst. (1999). A**55**, 383–395

Algorithms for deriving crystallographic space-group information

R. W. Grosse-Kunstleve

Yale University, New Haven, Connecticut 06520-8114, USA. E-mail: rwgk@laplace.csb.yale.edu

(*Received 28 April 1998; accepted 27 July 1998*)

Abstract

Algorithms are presented for three-dimensional crystallographic space groups, handling tasks such as the generation of symmetry operations, the characterization of symmetry operations (determination of rotation-part type, axis direction, sense of rotation, screw or glide part and location part), the determination of space-group type [identified by the space-group number of the *International Tables for Crystallography* (Dordrecht: Kluwer Academic Publishers)] and the generation of structure-seminvariant vectors and moduli. The latter are an algebraic description of allowed origin shifts, which are important in crystal structure determination methods or for comparing crystal structures. The space-group type determination produces a change-of-basis matrix which transforms a given space-group representation to the standard one according to the *International Tables for Crystallography*. The algorithms were implemented and tested using the SgInfo library. The source code is free for non-commercial applications.

1. Introduction

This paper presents a set of algorithms for the application of space-group symmetry in crystallographic programs. Any method for the solution of crystal structures and for structure refinement needs at least the set of symmetry operations for the space group under consideration. In this context, the symmetry operations can be viewed as 'primary symmetry information'. Frequently, 'secondary symmetry information' is needed for specific purposes; this information can be derived from the primary information, the symmetry operations. Examples of secondary symmetry information are structure-seminvariant vectors and moduli (needed in direct methods or Patterson methods), and axis directions and locations of symmetry elements (useful when dealing with atoms on special positions or with Harker vectors and planes).

The first algorithm presented addresses the generation of the primary information. Two approaches are in common use. One approach takes a space-group symbol, usually a Hermann–Mauguin (H–M) symbol as defined in the *International Tables for Crystallography* (Hahn, 1983) (IT83), and obtains the symmetry operations from a look-up table. This approach has the disadvantage of being limited to the tabulated space-group representations. Another approach attempts to overcome this limitation by interpreting the space-group symbol. Larson (1969), Burzlaff *et al.* (1977) and Burzlaff & Hountas (1982) devised algorithms which translate H–M symbols into a set of symmetry operations (the generators) which are then used to produce the full set of operations through group multiplication. However, H–M symbols were originally designed as a convenient description of given space-group representations. While it is natural to derive a H–M symbol for a given list of symmetry operations, it is problematic to derive the symmetry operations from a H–M symbol. Burzlaff & Zimmermann (1980) define simple rules for the interpretation of short H–M symbols which make the selection of the space-group type unique, but the rules for the selection of the location of the origin with respect to the symmetry elements that complies with the *International Tables for Crystallography* are 'too complicated to give the basis of a computer program' (Burzlaff & Hountas, 1982). Therefore, the existing algorithms fall back to look-up tables (implicit or explicit) for the origin selection.

The procedures of Larson (1969) [as implemented in the *GSAS* suite of programs (Larson & von Dreele, 1995)] and Burzlaff *et al.* (1977) [as implemented in the *LAZY PULVERIX* program (Yvon *et al.*, 1977)] produce identical results for all conventional H–M symbols [as listed in Hall & Grosse-Kunstleve (1999)], except for the symbols $Pnm2_1$, $P2_1nm$ and $Pm2_1n$ (Le Page, 1997). In a more rigorous test with the 534 sensible short H–M symbols possible for the orthorhombic crystal system, 179 differences were found in the resulting space-group representations. This means H–M symbols are both difficult to handle with computer methods and prone to cause misunderstandings. Furthermore, the collection of symmetry representations that can be described by H–M symbols is limited because there is no established notation for the selection of alternative origins, and no provision for symmetry representations where the generators have axes' directions other than the reference directions of the H–M notation. These limitations are serious when working with subgroup and supergroup relations.

The limitations and ambiguities of H–M symbols prompted Menzer (1960), Hall (1981) and Shmueli (1984) to suggest new types of space-group symbols. The

notation of Menzer was motivated by his work on the classification of reciprocal lattices of special lattice complexes. The notations of Hall and Shmueli were both designed to be 'computer adapted' (Shmueli, 1993). The definition of the *Hall symbols* was further refined by Hall & Grosse-Kunstleve (1999). These Hall symbols have some similarities with Hermann–Mauguin symbols, but define the space-group representation without ambiguities. Another advantage is that *any* three-dimensional crystallographic space-group representation can be described by a Hall symbol.

This paper presents an algorithm for the generation of symmetry operations, which was developed for use with a Hall symbol interpreter. The Hall symbol interpreter itself is not described. As a consequence of the clear definition of Hall symbols, the algorithm for the interpreter is trivial. Of course, the algorithm for the generation of symmetry operations is not restricted to use with a Hall symbol interpreter. A group-generating set of symmetry operations from any source can be used.

The remaining parts of this paper describe algorithms for the derivation of secondary symmetry information. The algorithms for the characterization of symmetry operations are the foundation for the determination of the space-group type and the generation of structure-seminvariant vectors and moduli. A symmetry operation is characterized by the type of the rotation part, the axis direction, the sense of rotation, and the intrinsic (screw or glide) part and location part of the translation part. The space-group type determination produces a change-of-basis matrix which transforms a given representation to the standard one according to IT83. Combining this algorithm with that of Le Page (1987) would result in a program for the automatic space-group determination of atomic structures. It could also be used in the determination of subgroup and supergroup relations. Another application under consideration is to use the change-of-basis matrices to transform symmetry-related information which is difficult to generate from first principles, but easily tabulated for the standard space-group representations, for example asymmetric units and Euclidean normalizers.

Algorithms similar to those presented here do exist, but have not been published. The algorithm of Biosym's (now Molecular Simulations Inc., http://www.msi.com) *InsightII FIND_SYMMETRY* module has not been published and is only available as part of a large commercial package. The program *PowderCell* (Kraus & Nolze, 1996, 1997) includes algorithms for the derivation of transformations between various settings of a space-group type, and also the generation of *translationsgleiche* and *klassengleiche* subgroups, but the source code is not available and the algorithms have not been published. The program *SPACER* (Stróz, 1997) includes algorithms for the characterization of symmetry operations, but the corresponding algorithm has not been published and the source code was not disclosed.

2. Notation

Full group: a set of symmetry operations which is closed under binary multiplication. Symmetry operations which are related by combinations of the unit translations are considered to be equal.

Group generators: a set of symmetry operations which is used to generate the full group. This set is not necessarily a minimal set and can be redundant.

Space-group type: see definition in Section 8.2.1 of IT83 (Wondratscheck, 1983). Two space groups belong to the same space-group type if they correspond to the same entry in IT83.

Space-group representation: a particular full group is a representation of the corresponding space-group type. For example, the space-group representations denoted by the H–M symbols *Pmna*, *Pnmb*, *Pbmn*, *Pcnm*, *Pncm* and *Pman* all correspond to the same space-group type (space group No. 53 in IT83). In general, each space-group type has an infinite number of representations.

The $(\mathbf{W}, \mathbf{w})$ formalism for symmetry operations from IT83 is used. $\mathbf{W}$ is the (3×3) *rotation part* or *rotation matrix* and $\mathbf{w}$ is the (3×1) *translation part* or *translation vector* of the operation.

To simplify the presentation, the following definition is introduced. Let $\mathbf{W}$ be the rotation part of a symmetry operation $(\mathbf{W}, \mathbf{w})$. The corresponding proper rotation matrix $\mathbf{W}_p$ is defined as

$$\mathbf{W}_p = \mathbf{W} \qquad \text{if } \det(\mathbf{W}) > 0,$$
$$\mathbf{W}_p = -\mathbf{W} \qquad \text{if } \det(\mathbf{W}) < 0.$$

Vectors and matrices are enclosed by square brackets. The superscript T denotes the transpose of a vector or matrix. However, matrices shown in Jones Faithful notation (*e.g.* x, y, z) are not enclosed by brackets.

The notation ${}^{[ex\ ey\ ez]}N$ and ${}^{[ex\ ey\ ez]}N^{-1}$ is used to describe a rotation matrix of type N $(1, 2, 3, 4, 6, \bar{1}, \bar{2} = m, \bar{3}, \bar{4}, \bar{6})$ with axis direction $[e_x\ e_y\ e_z]$. The superscript -1 is used for matrices with a negative sense of rotation (see also §4).

The first non-zero entry in a row of a matrix (if one exists) is called a *pivot*.

A square matrix $\mathbf{A}$ is called *unimodular* if $|\det(\mathbf{A})| = 1$.

$\mathbb{Z}$ = set of integer numbers.

$\mathbb{Q}$ = set of rational numbers.

3. Efficient generation of the symmetry operations of a space group

Given a list of symmetry operations, the generation of the full space group is, in its simplest form, a fairly trivial matter. Starting at the top of the list, each matrix in the list is multiplied by all preceding matrices, and also with itself. Each matrix product is again a symmetry operation, which is compared with the entries in the list. If not already present, it is appended to the list.

Thus, for a space group with a total of n_S symmetry operations, $n_S(n_S+1)/2$ matrix multiplications are necessary. For example, 18528 matrix multiplications are necessary for generating space group $Fm\bar{3}m$ (No. 225, M = 192). In addition, more than 1.7 million matrix comparisons are necessary.

The simplistic approach to the generation is therefore very inefficient for high-symmetry space groups. This problem has already been addressed partially by Hall (1981). The algorithm presented here is a generalization of Hall's procedure and is more robust.

The key to higher efficiency in group generation is to take advantage of the structure intrinsic to space groups. Two properties of any crystallographic space group are exploited:

(*a*) The list of centring vectors is a normal subgroup and can be maintained separately [see theorem T7.23 in Boisen & Gibbs (1990)].

(*b*) The inversion operation can be maintained separately (even if the inversion is not at the origin).

From this it follows that the resulting *space-group structure* consists of three parts:

(i) A list of n_Z centring vectors.

(ii) A variable f_I and the translation part $\mathbf{w}_{I,L}$ of an inversion operation, if present. The rotation part $-\mathbf{I}$ is implied. Let $f_I = 1$ if no inversion operation exists, $f_I = 2$ otherwise.

(iii) A list of $n_M = n_S/(n_Z f_I)$ representative symmetry matrices $(\mathbf{W}_L, \mathbf{w}_L)$.

Both properties (*a*) and (*b*) are commonly exploited. For example, the centring vectors are also listed separately in IT83, and many programs keep track of the inversion operation with a single variable similar to f_I. However, all programs known to the author only treat inversions at the origin in this manner.

The optimized group generation can be understood more easily by first considering the rotation parts only and ignoring both the centring vectors and all translation parts. This reduced set of symmetry operations describes the point group corresponding to the given space group. The maximum number of point group matrices is 48 (point group $m\bar{3}m$). For this and for all other centric point groups $-\mathbf{I}$ is one element of the group. It follows that each matrix $\mathbf{W}$ in the group is related to a second matrix $-\mathbf{I}\cdot\mathbf{W} = \mathbf{W}\cdot(-\mathbf{I}) = -\mathbf{W}$, which must also be an element of the group. Therefore the list of point-group matrices can be cut in half by keeping only one representative of a pair $\mathbf{W}$ and $-\mathbf{W}$. This corresponds to property (*b*) above. To record the reduction, a variable has to be introduced. In this discussion, f_I is used for this purpose. The *point-group structure* therefore consists of two parts: the variable f_I and the list of representative rotation parts with at most 24 elements.

When the translation parts and centring vectors are now considered, theorem T7.26 of Boisen & Gibbs (1990) can be used to arrive at property (*a*). The theorem states that if a space group has two elements $(\mathbf{W}, \mathbf{w}_1)$ and $(\mathbf{W}, \mathbf{w}_2)$ there must be a third element $(\mathbf{I}, \mathbf{w}_2 - \mathbf{w}_1)$. This means it is only necessary to keep one representative of $(\mathbf{W}, \mathbf{w}_1)$ and $(\mathbf{W}, \mathbf{w}_2)$ in the *list of representative symmetry matrices*, if $\Delta\mathbf{w} = \mathbf{w}_2 - \mathbf{w}_1$ is added to the list of centring vectors.

Property (*b*) can be derived by generalizing the reduction of the list of point-group operations due to the presence of $-\mathbf{I}$, and invoking property (*a*) if necessary. If the space group has an element $(-\mathbf{I}, \mathbf{w}_I)$, each matrix $(\mathbf{W}, \mathbf{w}_1)$ in the group is related to the matrices $(\mathbf{W}, \mathbf{w}_1)(-\mathbf{I}, \mathbf{w}_I) = (-\mathbf{W}, \mathbf{w}_2)$ and $(-\mathbf{I}, \mathbf{w}_I)(\mathbf{W}, \mathbf{w}_1) = (-\mathbf{W}, \mathbf{w}_3)$, with $\mathbf{w}_2$ defined as $\mathbf{w}_2 = \mathbf{W}\cdot\mathbf{w}_I + \mathbf{w}_1$ and $\mathbf{w}_3 = -\mathbf{w}_1 + \mathbf{w}_I$. Therefore, the list of space-group matrices can be reduced by keeping only one representative of a pair $(\mathbf{W}, \mathbf{w}_1)$ and $(-\mathbf{W}, \mathbf{w}_2)$, or $(\mathbf{W}, \mathbf{w}_1)$ and $(-\mathbf{W}, \mathbf{w}_3)$. To record the reduction [which also involves the pair $(\mathbf{I}, 0)$ and $(-\mathbf{I}, \mathbf{w}_I)$], a variable (f_I) has to be introduced, and the translation part $\mathbf{w}_I$ has to be stored. If $\mathbf{w}_2$ and $\mathbf{w}_3$ are not equal, property (*a*) is invoked, and $\Delta\mathbf{w} = \mathbf{w}_2 - \mathbf{w}_3$ is added to the list of centring vectors. It follows that the maximum number of elements in the list of representative symmetry matrices is still only 24.

The *structured group generation algorithm* has two main layers. The upper layer contains procedures to expand the list of centring vectors and symmetry matrices by repeated binary combination of the existing elements in the lists. For symmetry matrices, this combination is a matrix multiplication. For the centring vectors, the combination is a vector addition. Otherwise the two group generation algorithms are identical to the simplistic algorithm described earlier.

The upper layer also keeps track of the interdependencies of the elements of the space-group structure. Each centring vector has to be multiplied with all rotation matrices in the list of representative symmetry matrices to possibly generate additional centring vectors. Furthermore, if an inversion with translation part $\mathbf{w}_I$ is present, another centring vector $\Delta\mathbf{w}$ can arise for each element $(\mathbf{W}_L, \mathbf{w}_L)$ in the list of representative symmetry matrices,

$$\begin{aligned}&(\mathbf{W}_L, \mathbf{w}_L)(-\mathbf{I}, \mathbf{w}_I) = (-\mathbf{W}_L, \mathbf{W}_L\mathbf{w}_I + \mathbf{w}_L),\\ &(-\mathbf{I}, \mathbf{w}_I)(\mathbf{W}_L, \mathbf{w}_L) = (-\mathbf{W}_L, -\mathbf{w}_L + \mathbf{w}_I)].\end{aligned} \tag{1}$$

Invoking property (*a*) yields $\Delta\mathbf{w} = \mathbf{W}_L\mathbf{w}_I + 2\mathbf{w}_L - \mathbf{w}_I$.

The lower layer of the algorithm contains procedures for adding a centring vector, a translation part of an inversion operation, or a symmetry operation $(\mathbf{W}, \mathbf{w})$ to the space-group structure. In the current context, 'adding' means 'appending after testing'.

When adding a centring operation, the test is simply to check if the vector is already in the list. If the test is negative, the centring operation is appended to the list. No action is taken if the test is positive.

When adding a translation part $\mathbf{w}_I$ of an inversion operation, the algorithm uses f_I to test if there is already

a translation part $\mathbf{w}_{I,L}$. If there is no such translation part (if $f_I = 1$), the new translation part is stored as $\mathbf{w}_{I,L}$ and f_I is set to 2. Otherwise property (*a*) is invoked, and a centring operation $\mathbf{w}_{I,L} - \mathbf{w}_I$ is added.

When adding a symmetry operation $(\mathbf{W}, \mathbf{w})$, two tests are carried out. The first test is to check if the rotation part $\mathbf{W}$ is already in the list of representative symmetry matrices. If so, property (*a*) is again invoked and a centring operation $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}_L$ is added, where $\mathbf{w}_L$ is the translation part of the operation $(\mathbf{W}_L, \mathbf{w}_L)$ with $\mathbf{W} = \mathbf{W}_L$. Otherwise the second test is carried out, which determines whether the rotation part $-\mathbf{W}$ is already in the list. If so, a translation part $\mathbf{w}_I = \mathbf{w} + \mathbf{w}_L$ of an inversion operation has to be added [this follows from $(-\mathbf{I}, \mathbf{w}_I)(\mathbf{W}, \mathbf{w}) = (-\mathbf{W}, \mathbf{w}_I - \mathbf{w}) = (-\mathbf{W}, \mathbf{w}_L)$]. If not, the operation $(\mathbf{W}, \mathbf{w})$ is appended to the list.

The structured group generation for space group $Fm\bar{3}m$ involves only 300 matrix multiplications, approximately 3500 matrix comparisons, 120 multiplications of the centring vectors and rotation parts in the list of representative symmetry matrices, and a small number of vector additions and book-keeping operations. Overall, the algorithm is approximately 100 times faster than the simplistic method, and in general even faster than a table look-up from a file. However, efficiency is not the only gain. The fact that the space group is immediately structured is also very helpful when using the symmetry information in other applications, for example in the space-group type determination algorithm described in one of the following sections.

Remark. The efficiency of the group generation could be further improved by using a *Dimino* algorithm as implemented in the *GAP* package (GAP, 1995), where it can be applied to *n*-dimensional finite groups. An implementation in the framework of the SgInfo library (Grosse-Kunstleve, 1995*a*,*b*) was not attempted.

4. Characterization of symmetry operations

Characterizing a symmetry operation $(\mathbf{W}, \mathbf{w})$ involves the following.

(*a*) Determination of the properties of the rotation part:

Rotation-part type (1, 2, 3, 4, 6, $\bar{1}$, $\bar{2} = m$, $\bar{3}$, $\bar{4}$, $\bar{6}$).

Axis direction of the corresponding proper rotation $\mathbf{W}_p$.

Sense of rotation with respect to the axis direction

(*b*) Decomposition of the translation part in:

Intrinsic (screw or glide) part.

Location part (origin shift).

Algebraic procedures for these characterizations are given by Fischer & Koch (1983) and Boisen & Gibbs (1990).

When casting these procedures in programmable algorithms, working with *row echelon forms* is the key to the determination of the axes directions and the decompositions of the translation parts. General definitions of row echelon forms are found, for example, in Boisen & Gibbs (1990) and standard linear algebra text books such as Strang (1986). Any $(m \times n)$ matrix $\mathbf{M}$ can be converted into a row echelon form $\mathbf{U}$ with a series of row operations. The procedure is similar to Gaussian elimination and results in an upper triangular matrix $\mathbf{U}$. If the row operations are recorded in an $(m \times m)$ matrix $\mathbf{T}$, the equation $\mathbf{TM} = \mathbf{U}$ holds at any time. The actual algorithm for the computation of $\mathbf{U}$ and $\mathbf{T}$ was taken from the *CrystGAP* package (Eick *et al.*, 1997), which is freely available as part of the *GAP* system (GAP, 1995). The *CrystGAP* '*RowEchelonFormT*' algorithm has two special properties: $\mathbf{M}$, $\mathbf{U}$ and $\mathbf{T}$ are integer matrices, and the row operations are performed in a way such that $\mathbf{T}$ is unimodular at all times. Working with integer matrices is, in general, a substantial practical advantage. The fact that $\mathbf{T}$ is unimodular is important for the computation of the Smith normal form, which is needed later for the determination of change-of-basis matrices and structure-seminvariant vectors and moduli. For the computation of the axes' directions and the decomposition of the translation parts, $\mathbf{T}$ is not required to be unimodular. However, implementation is simplified by always using the same algorithm for the construction of $\mathbf{U}$.

Table 1. *Look-up table for rotation-part type N*

	tr(**W**)						
det(**W**)	−3	−2	−1	0	1	2	3
1			2	3	4	6	1
−1	$\bar{1}$	$\bar{6}$	$\bar{4}$	$\bar{3}$	$\bar{2}$		

An example will be used to show all parts of the algorithm for the characterization of symmetry operations. The example symmetry operation $(\mathbf{W}, \mathbf{w})$ to be characterized is given as

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} -1/3 \\ 0 \\ 1/6 \end{bmatrix}. \quad (2)$$

The rotation-part type N of $\mathbf{W}$ is easily derived from the determinant and trace of $\mathbf{W}$ by look-up in Table 1 (Fischer & Koch, 1983),

$$\det(\mathbf{W}) = 1, \quad \mathrm{tr}(\mathbf{W}) = 0 \Rightarrow N = 3.$$

The next step in the characterization, determination of the axis direction of $\mathbf{W}_p$, is only carried out if $|N| > 1$. The determination of the axis direction is equivalent to the determination of the eigenvector corresponding to the eigenvalue 1,

$$\mathbf{W}_p\mathbf{e} = \mathbf{e} \Rightarrow (\mathbf{W}_p - \mathbf{I})\mathbf{e} = 0, \quad (3)$$

where $\mathbf{e}$ is the eigenvector and $\mathbf{I}$ is the identity matrix. To determine the eigenvector for $\mathbf{W}_p$ in our example,

$$(\mathbf{W}_p - \mathbf{I}) = \begin{bmatrix} -1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & -1 \end{bmatrix} \quad (4)$$

is converted to the row echelon form

$$\mathbf{U} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (5)$$

The row echelon form immediately reveals the rank of $(\mathbf{W}_p - \mathbf{I})$, since it is equal to the number of non-zero rows in $\mathbf{U}$. The rank of $(\mathbf{W}_p - \mathbf{I})$ is always equal to 2 as in this case. The equation $(\mathbf{W}_p - \mathbf{I})\mathbf{e} = 0$ with $\mathbf{e} = [e_1, e_2, e_3]^T$ can now be solved by backsubstitution. The unknowns e_1, e_2 and e_3 go into two groups. One is made up of the *basic variables*, those that correspond to columns with pivots. The other group is made up of the *free variables*, corresponding to the columns without pivots. For a matrix of rank 2, as in the example, there is just one free variable. To obtain a non-trivial solution $\mathbf{e} \neq 0$, some value $\neq 0$ has to be assigned to the free variable. In the example, e_3 is set to 1 and backsubstitution yields $\mathbf{e} = [-1, 1, 1]^T$.

The particular backsubstitution algorithm used scales the solution vector with a factor d in order to obtain an integer vector, if necessary. In the context of other algorithms, d needs to be considered, but the value of d is not important in the present context.

The scaling makes sure that the presented algorithm produces the shortest eigenvector with integer components, but the sign is more or less arbitrary, *e.g.* $\mathbf{e} = [-1, 1, 1]^T$ and $\mathbf{e} = [1, -1, -1]^T$ are both valid eigenvectors. To make axes' directions comparable, and to define a reference orientation for the determination of the sense of rotation, the following convention is adopted: for a pair of eigenvectors $[e_1, e_2, e_3]^T$ and $[-e_1, -e_2, -e_3]^T$, the eigenvector chosen is the one with $e_3 > 0$. If $e_3 = 0$, the one chosen is that with $e_2 > 0$. If $e_2 = 0$, the one chosen is that with $e_1 > 0$. According to this convention, we would choose the eigenvector $\mathbf{e} = [-1, 1, 1]^T$ for our example above.

For $N < -1$ (mirror plane, roto-inversion), the eigenvector $\mathbf{e}$ of $\mathbf{W}_p$ is also an eigenvector of $\mathbf{W}$, having the eigenvalue -1. For a mirror plane, $\mathbf{e}$ is normal to the mirror plane. For the other roto-inversions, $\mathbf{e}$ is parallel to the roto-inversion axis.

For $|N| > 2$, the sense of rotation with respect to the axis direction is determined according to theorem TA3.9 in Boisen & Gibbs (1990). The axis direction $\mathbf{e} = [e_1, e_2, e_3]^T$ and the matrix

$$\mathbf{W}_p = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad (6)$$

are evaluated. $\mathbf{e}$ points in the positive direction of the rotation represented by $\mathbf{W}_p$ if one of the following conditions is true:

(*a*) $e_2 = e_3 = 0$ and $e_1 r_{32} > 0$,

(*b*) $r_{21}e_3 - r_{31}e_2 > 0$.

In our example, condition (*b*) is true, and the final result for the rotation part $\mathbf{W}$ is that it represents a threefold rotation about $[-1, 1, 1]^T$ with a positive sense of rotation.

Following the procedure given by Fischer & Koch (1983), the first step in the analysis of the translation part $\mathbf{w}$ is the decomposition into the intrinsic part $\mathbf{w}_i$ and the location part $\mathbf{w}_l$. For this, $(\mathbf{W}, \mathbf{w})^n = (\mathbf{I}, \mathbf{t})$ has to be computed, where the rotational order $n = |N|$, except for $N = -1$ and $N = -3$. For those two cases, $n = -2N$. Now the intrinsic part is obtained as $\mathbf{w}_i = (1/n)\mathbf{t}$, and the location part is the difference $\mathbf{w}_l = \mathbf{w} - \mathbf{w}_i$. For example,

$$\left(\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} -1/3 \\ 0 \\ 1/6 \end{bmatrix} \right)^3 = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1/2 \\ 1/2 \\ 1/2 \end{bmatrix} \right)$$

$$\Rightarrow \mathbf{w}_i = \begin{bmatrix} -1/6 \\ 1/6 \\ 1/6 \end{bmatrix}, \quad \mathbf{w}_l = \begin{bmatrix} -1/6 \\ -1/6 \\ 0 \end{bmatrix}. \qquad (7)$$

The symmetry operation $(\mathbf{W}, \mathbf{w}_l)$ corresponds to a pure rotation or reflection at the same location as the original operation $(\mathbf{W}, \mathbf{w})$. The location of $(\mathbf{W}, \mathbf{w}_l)$ is described by the solutions of the equation $(\mathbf{W}, \mathbf{w}_l)\mathbf{x} = \mathbf{x}$. The equation is rearranged as $(\mathbf{W} - \mathbf{I})\mathbf{x} = -\mathbf{w}_l$, and the row echelon form $\mathbf{U}$ of $(\mathbf{W} - \mathbf{I})$ computed as before. However, since this is an inhomogeneous system of equations, $\mathbf{T}$, the matrix that is used to record the row operations, is also needed:

$$(\mathbf{W}-\mathbf{I}) = \begin{bmatrix} -1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & -1 \end{bmatrix} \rightarrow \mathbf{U} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{T} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ -1 & 1 & 1 \end{bmatrix} \Rightarrow \mathbf{T}(-\mathbf{w}_l) = \begin{bmatrix} -1/6 \\ -1/6 \\ 0 \end{bmatrix}. \qquad (8)$$

Setting the free variable equal to 0 and solving the system $\mathbf{T}(\mathbf{W} - \mathbf{I})\mathbf{x} = \mathbf{U}\mathbf{x} = \mathbf{T}(-\mathbf{w}_l)$ for $\mathbf{x}$ by straightforward backsubstitution yields $\mathbf{x} = [0, -1/6, 0]^T$. This is a fixed point of the symmetry operation $(\mathbf{W}, \mathbf{w}_l)$. For rotation-part types $N > 1$ and $N = -2$, the combination with the axis direction of $\mathbf{W}$ is a convenient description of all fixed points. For the other rotation-part types, the fixed point is unique.

Remark. $\mathbf{x}$ could also be determined by the equation $\mathbf{x} = (\mathbf{W} - \mathbf{I})^+(-\mathbf{w}_l)$, where $(\mathbf{W} - \mathbf{I})^+$ is the pseudo-inverse of $(\mathbf{W} - \mathbf{I})$ (Strang, 1986). However, in the given framework, this mathematically more elegant formula-

tion is in practice more cumbersome than the algorithm presented.

5. Determination of space-group type

Any arbitrary set of symmetry operations obtained through group multiplication must belong to one of the 230 space-group types (Wondratschek, 1983), and can be transformed to the corresponding standard representation of IT83 by some change-of-basis matrix $(\mathbf{C}, \mathbf{c})$, with $\det(\mathbf{C}) > 0$. A symmetry operation $(\mathbf{W}_O, \mathbf{w}_O)$ in the original (arbitrary) basis system is transformed to $(\mathbf{W}_S, \mathbf{w}_S)$ in the standard basis system with a similarity transformation (Strang, 1986) $(\mathbf{W}_S, \mathbf{w}_S) = (\mathbf{C}, \mathbf{c}) \times (\mathbf{W}_O, \mathbf{w}_O)(\mathbf{C}, \mathbf{c})^{-1}$. The space-group type of a given set of symmetry operations can be determined by trying to construct $(\mathbf{C}, \mathbf{c})$ for each of the 230 standard settings tabulated in IT83 as described below. Only one of the trials can succeed. With a few obvious optimizations, a space-group type determination typically takes only a few hundredths of a second on a Pentium II computer.

The symmetry operations of the 230 standard settings are encoded as Hall symbols. The Hall symbols are translated to generators which are then expanded to the full group.

The algorithm for the construction of $(\mathbf{C}, \mathbf{c})$ can be subdivided into the following basic steps:

(*a*) Construction of $(\mathbf{P}, 0)$, which transforms $\mathbf{G}_o$ to $\mathbf{G}_p$, where $\mathbf{G}_o$ is the set of symmetry operations in the original setting and $\mathbf{G}_p$ is the set of symmetry operations in a primitive setting.

(*b*) Determination of the point group of $\mathbf{G}_o$ (and $\mathbf{G}_p$).

(*c*) Construction of $(\mathbf{M}', 0)$, which transforms $\mathbf{G}_p$ to $\mathbf{G}_{M'}$, where the symmetry operations of $\mathbf{G}_{M'}$ refer to a convenient 'standard' basis for the given point group.

(*d*) Adjustment of $(\mathbf{M}', 0)$ to $(\mathbf{M}, 0)$ for certain combinations of Laue groups and centring types.

For triclinic, tetragonal, trigonal, hexagonal and cubic space groups, the transformation matrix $(\mathbf{C}, 0) = (\mathbf{M}, 0)(\mathbf{P}, 0)$ is already determined at this point. The final step of the algorithm is:

(*e*) Determination of the origin shift $(\mathbf{I}, \mathbf{c})$.

For monoclinic and orthorhombic space groups, a trial loop over six transformation matrices $(\mathbf{A}, 0)$ for alternative cell choices or settings is necessary (corresponding to the six columns in Table 4.3.1 of IT83). The trial matrices $(\mathbf{A}, 0)$ are generated from the identity matrix $\mathbf{I}$ and the crystal-system-specific rotation matrices $\mathbf{R}_2$ and $\mathbf{R}_3$ of Table 2. Table 3 lists the actual $\mathbf{A}$ for each pass of the trial loop. For each trial transformation matrix $(\mathbf{C}, 0) = (\mathbf{A}, 0)(\mathbf{M}, 0)(\mathbf{P}, 0)$, step (*e*) is attempted.

It is worth emphasizing that a knowledge of a metric is not needed for any part of the algorithm. All computations can be performed with the symmetry operations alone. A certain algebraic form of a symmetry operation implies restrictions for the metric of the underlying basis system. This means the essential properties of an actual metric are encoded in the symmetry operations, and the metric itself is not necessary for the construction of $(\mathbf{C}, \mathbf{c})$.

Table 2. *Crystal-system-specific rotation matrices $\boldsymbol{R}_2$ and $\boldsymbol{R}_3$ used in Table 3*

Crystal system	$\mathbf{R}_2$	$\mathbf{R}_3$
Monoclinic	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = {}^{[101]}2$	$\begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} = {}^{[010]}3$
Orthorhombic	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} = {}^{[110]}2$	$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = {}^{[111]}3$

Table 3. *Trial matrices A for monoclinic and orthorhombic space groups*

Loop pass	Trial A
1	$\mathbf{I}$
2	$\mathbf{R}_3$
3	$\mathbf{R}_3.\mathbf{R}_3$
4	$\mathbf{R}_2$
5	$\mathbf{R}_2.\mathbf{R}_3$
6	$\mathbf{R}_2.\mathbf{R}_3.\mathbf{R}_3$

The following is a detailed description of the steps listed above.

5.1. *Step (a): construction of (P, 0)*

The first step in the determination of $(\mathbf{P}, 0)$ is to build an expanded list of the centring operations. This is performed by looping all combinations of subtracting 1 from the non-zero components of each centring operation. For example, if the centring operations are given as $[2/3, 1/3, 1/3]^T$ and $[1/3, 2/3, 2/3]^T$ (*R*-centred cell), the expanded list is:

$[2/3, 1/3, 1/3]^T$, $[1/3, 2/3, 2/3]^T$,
$[2/3, 1/3, -2/3]^T$, $[1/3, 2/3, -1/3]^T$,
$[2/3, -2/3, 1/3]^T$, $[1/3, -1/3, 2/3]^T$,
$[2/3, -2/3, -2/3]^T$, $[1/3, -1/3, -1/3]^T$,
$[-1/3, 1/3, 1/3]^T$, $[-2/3, 2/3, 2/3]^T$,
$[-1/3, 1/3, -2/3]^T$, $[-2/3, 2/3, -1/3]^T$,
$[-1/3, -2/3, 1/3]^T$, $[-2/3, -1/3, 2/3]^T$,
$[-1/3, -2/3, -2/3]^T$, $[-2/3, -1/3, -1/3]^T$.

To make the following steps more efficient, this list is reduced by eliminating linearly dependent vectors (*e.g.* $[2/3, 1/3, -2/3]^T$ and $[-2/3, -1/3, 2/3]^T$). The reduced list is then sorted such that shorter vectors are at the beginning of the list. Finally, the unit translations $[1, 0, 0]^T$, $[0, 1, 0]^T$ and $[0, 0, 1]^T$ are appended. This is necessary because in the general case the expanded list of centring operations does not necessarily span the whole three-dimensional space (*e.g.* in the case of an

Table 4. *Look-up table for the determination of point-group type*

Conditions					Point group
#3 + #$\bar{3}$ = 8 (cubic)	n_M = 12	Acentric			23
		Centric			$m\bar{3}$
	n_M = 24	Acentric	#4 = 6		432
			#$\bar{4}$ = 6		$\bar{4}3m$
		Centric			$m\bar{3}m$
#6 + #$\bar{6}$ = 2 (hexagonal)	n_M = 6	Acentric	#6 = 2		6
			#$\bar{6}$ = 2		$\bar{6}$
		Centric			6/m
	n_M = 12	Acentric	#6 = 2	#2 = 7	622
				#$\bar{2}$ = 6	6mm
			#$\bar{6}$ = 2		$\bar{6}m2$
		Centric			6/mmm
#3 + #$\bar{3}$ = 2 (trigonal)	n_M = 3	Acentric			3
		Centric			$\bar{3}$
	n_M = 6	Acentric	#2 = 3		32
			#$\bar{2}$ = 3		3m
		Centric			$\bar{3}m$
#4 + #$\bar{4}$ = 2 (tetragonal)	n_M = 4	Acentric	#4 = 2		4
			#$\bar{4}$ = 2		$\bar{4}$
		Centric			4/m
	n_M = 8	Acentric	#4 = 2	#2 = 5	422
				#$\bar{2}$ = 4	4mm
			#$\bar{4}$ = 2		$\bar{4}m2$
		Centric			4/mmm
#2 + #$\bar{2}$ = 3 (orthorhombic)	Acentric	#2 = 3			222
		#$\bar{2}$ = 2			$mm2$
	Centric				mmm
#2 + #$\bar{2}$ = 1 (monoclinic)	Acentric	#2 = 1			2
		#$\bar{2}$ = 1			m
	Centric				2/m
n_M = 1 (triclinic)	Acentric				1
	Centric				$\bar{1}$

H-centred cell). The final list for the example *R*-centred cell is

$$[-1/3, 1/3, 1/3]^T,$$
$$[-1/3, 1/3, -2/3]^T,$$
$$[-1/3, -2/3, 1/3]^T,$$
$$[-1/3, -2/3, -2/3]^T,$$
$$[2/3, 1/3, 1/3]^T,$$
$$[2/3, 1/3, -2/3]^T,$$
$$[2/3, -2/3, 1/3]^T,$$
$$[1, 0, 0]^T,$$
$$[0, 1, 0]^T,$$
$$[0, 0, 1]^T.$$

The construction of **P** is now attempted with a three-deep loop over all possible subsets of three vectors (without permutations). For each pass of the loop, these three vectors $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ are used as a basis, and the corresponding transformation matrix is [see theorem T2.10 in Boisen & Gibbs (1990)]

Table 5. *Look-up table used for the construction of (M′, 0)*

Laue group	n_e		|N|	
$\bar{1}$	1	1		
2/m	1	2		
mmm	3	2	2	2
4/m	1	4		
4/mmm	2	4	2	
$\bar{3}$	1	3		
$\bar{3}m$	2	3	2	
6/m	1	3		
6/mmm	2	3	2	
$m\bar{3}$	2	3	2	
$m\bar{3}m$	2	3	4	

$$\mathbf{P}^{-1} = \begin{bmatrix} p_{1,1} & p_{2,1} & p_{3,1} \\ p_{1,2} & p_{2,2} & p_{3,2} \\ p_{1,3} & p_{2,3} & p_{3,3} \end{bmatrix}. \quad (9)$$

If $\det(\mathbf{P}^{-1})$ is equal to the number of centring operations, a valid **P** is obtained by inverting $\mathbf{P}^{-1}$. If $-\det(\mathbf{P}^{-1})$ is equal to the number of centring operations, the first column of $\mathbf{P}^{-1}$ is multiplied by -1 before performing the inversion.

Any **P** obtained in this way could be used to transform $\mathbf{G}_o$ to $\mathbf{G}_p$. However, some **P** can result in symmetry matrices with rotation parts containing fractional elements. If this is the case, the search loop over the subsets of three vectors is continued until a **P** has been found that produces only symmetry operations with rotation parts with integer elements.

Remark. For the example *R*-centred cell, the first three vectors in the list above satisfy all search conditions. This means that the search loop returns a valid **P** after only one pass. However, in general the search loop can be computationally expensive [$l(l-1)(l-2)/6$ passes, where l is the number of elements in the list], and it is the feeling of the author that a more elegant search for a primitive basis could exist.

5.2. *Step (b): determination of the point group*

The point-group type of a space group can easily be determined by first counting the number #*N* of times each rotation-part type *N* occurs in the list of representative symmetry matrices with $n_M = \Sigma\#N$ elements, and then matching against the list of possible cases in Table 4.

5.3. *Step (c): construction of (**M′**, 0)*

The basic idea for the construction of $(\mathbf{M}', 0)$ is to use the axes' directions of Laue-group-specific symmetry operations as a new basis. It is trivial to derive the Laue-group type from the point-group type as determined in the previous step (*b*). For a given Laue-group type, the list of representative symmetry matrices is searched for the n_e elements with the rotation-part types |*N*| listed in Table 5.

For triclinic space groups (Laue group $\bar{1}$), no axis direction is available because any vector is an eigenvector of $\mathbf{W}_p$ (which corresponds to the symmetry operation listed in Table 5), and $\mathbf{M}' = \mathbf{I}$ can simply be used. For all other cases with $n_e = 1$, only one axis direction $\mathbf{e}$ is available, which is used as basis vector $\mathbf{b}_z$. The two other vectors $\mathbf{b}_x$ and $\mathbf{b}_y$ which are necessary to define a basis have to be found with a different technique. The vectors $\mathbf{b}_x$ and $\mathbf{b}_y$ are determined in a Laue-group-dependent manner that is described below. Once $\mathbf{b}_x$ and $\mathbf{b}_y$ are determined, $\mathbf{M}'^{-1}$ is set to

$$\mathbf{M}'^{-1} = \begin{bmatrix} b_{x,1} & b_{y,1} & b_{z,1} \\ b_{x,2} & b_{y,2} & b_{z,2} \\ b_{x,3} & b_{y,3} & b_{z,3} \end{bmatrix}. \qquad (10)$$

If $\det(\mathbf{M}'^{-1}) < 0$, the first and the second column of $\mathbf{M}'^{-1}$ are swapped. Finally, $\mathbf{M}'$ is obtained by inverting $\mathbf{M}'^{-1}$.

To arrive at a conventional basis for Laue group $2/m$, $\mathbf{b}_x$ and $\mathbf{b}_y$ must be chosen perpendicular to $\mathbf{b}_z$. Theorem TA4.1 in Boisen & Gibbs (1990) provides a method for performing this without involving a metric tensor. According to the theorem, a vector $\mathbf{x}$ is in the plane perpendicular to the axis direction $\mathbf{e}$ of a proper rotation matrix $\mathbf{W}_p$ with rotational order n, if and only if $\mathbf{S} \cdot \mathbf{x} = 0$, with $\mathbf{S} = \mathbf{W}_p + \mathbf{W}_p^2 + \ldots + \mathbf{W}_p^n$. To solve the equation $\mathbf{S} \cdot \mathbf{x} = 0$, the row echelon matrix $\mathbf{U}$ is computed from $\mathbf{S}$. For example,

$$\mathbf{S} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \rightarrow U = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (11)$$

The rank of $\mathbf{S}$ is always equal to 1 as in this case, and $\mathbf{x}$ has two free components. Four solution vectors to the system $\mathbf{S} \cdot \mathbf{x} = 0$ are obtained by assigning, in turn, the pairs of values $[1, 0]$, $[0, 1]$, $[1, 1]$ and $[1, -1]$ (corresponding to the four shortest vectors with $\mathbf{x} \geq 0$ in the two-dimensional Cartesian plane) to the free components. The third component for each of the four solution vectors is computed by backsubstitution into $\mathbf{U} \cdot \mathbf{x} = 0$. There are six possibilities for picking a pair of basis vectors $\mathbf{b}_x$ and $\mathbf{b}_y$ out of the four solutions. The pair which results in the smallest value of $|\det(\mathbf{M}'^{-1})|$ is accepted for the final $\mathbf{M}'$.

For the Laue groups 4/m, $\bar{3}m$ and $6/m$, $\mathbf{U}$ and the four solution vectors are computed as before for Laue group $2/m$. Each of the four solution vectors is, in turn, used as trial $\mathbf{b}_x$, and $\mathbf{b}_y = \mathbf{W}_p \cdot \mathbf{b}_x$ (note the choice of $|N| = 3$ for Laue group $6/m$). The $\mathbf{b}_x$ which results in the smallest value of $|\det(\mathbf{M}'^{-1})|$ is accepted for the final $\mathbf{M}'$.

For Laue group *mmm*, the three basis vectors necessary for the construction of $\mathbf{M}'^{-1}$ are immediately available. These are the three axis directions of the three twofold axes (referring to $\mathbf{W}_p$). It only remains to arrange the vectors such that $\det(\mathbf{M}'^{-1}) > 0$.

In the other cases with $n_e = 2$, two axis directions are available. Let $\mathbf{W}_{p,z}$ and $\mathbf{e}_z = \mathbf{b}_z$ correspond to the symmetry operation listed first in Table 5, and let $\mathbf{W}_{p,x}$ and $\mathbf{e}_x = \mathbf{b}_x$ correspond to the second one. The third vector is obtained from $\mathbf{b}_y = \mathbf{W}_{p,z} \cdot \mathbf{e}_x$ (note the choice of $|N| = 3$ for Laue group $6/mmm$).

Table 6. *Correction matrices* $\boldsymbol{M}_c$

Condition	Correction matrix $\mathbf{M}_c$
Laue group $2/m$	$\begin{bmatrix} 010 \\ 001 \\ 100 \end{bmatrix} = {}^{[111]}3^{-1}$
Laue groups $4/m$ and $4/mmm$ *C*-centred	$\begin{bmatrix} 110 \\ 1-10 \\ 00-1 \end{bmatrix} = C \rightarrow P$
Laue groups $4/m$ and $4/mmm$ *F*-centred	$\begin{bmatrix} 110 \\ -110 \\ 001 \end{bmatrix} = F \rightarrow I$
Laue groups $\bar{3}3$ and $\bar{3}m$ *obverse* setting	$\begin{bmatrix} -100 \\ 0-10 \\ 001 \end{bmatrix} = {}^{[001]}2$
Laue groups $\bar{3}m$ and $6/mmm$ *H*-centred	$\begin{bmatrix} 110 \\ -120 \\ 001 \end{bmatrix} = H \rightarrow P$
Point group $m\bar{3}$ *P*-centred, *c*-glide plane perpendicular to $[1, 0, 0]^{\mathrm{T}}$	$\begin{bmatrix} 0-10 \\ 100 \\ 001 \end{bmatrix} = {}^{[001]}4$

5.4. *Step (d): adjustment of (**M**′, 0) to (**M**, 0)*

The algorithm of step (*c*) does not always find a standard basis in the sense of IT83. To keep step (*c*) simple, the *c* axis is systematically chosen as unique axis for monoclinic space groups, instead of the standard *unique axis b*. For Laue group $4/mmm$, the twofold axis used as $\mathbf{b}_x$ is more or less randomly chosen among two possibilities. If by chance the twofold axis was used which runs parallel to $[1, 1, 0]^T$ in the standard basis, a *C*-centred cell will result instead of a primitive cell, or an *F*-centred cell instead of an *I*-centred cell. For very unusual settings of Laue group $4/m$ (for example, the group generated by $-2x - y$, $2x - z$, $-x + y + z$), the same kind of problem can result from the choice of basis vectors $\mathbf{b}_x$ and $\mathbf{b}_y$ as explained above. A similar case exists for Laue groups $\bar{3}m$ and $6/mmm$, where the random choice of the twofold axis can lead to an *H*-centred cell (see Section 1.2 in IT83) instead of a

Table 7. *Look-up table used in the determination of the origin shift (**I**, **c**)*

Crystal system	Axes directions **e** and rotation-part types $\|N\|$	
Triclinic	1	
Monoclinic	$^{[010]}2$	
Orthorhombic	$^{[001]}2$	$^{[100]}2$
Tetragonal	$^{[001]}4$	$^{[100]}2$
Trigonal	$^{[001]}3$	$^{[\bar{1}10]}2$ or $^{[110]}2$
Hexagonal	$^{[001]}6$	$^{[\bar{1}10]}2$
Cubic	$^{[001]}2$ or $^{[001]}4$	$^{[111]}3$

primitive cell. Another problem can occur for rhombohedral lattices (Laue groups $\bar{3}$ and $\bar{3}m$), where the choice of basis vectors can lead to an *obverse* setting instead of the standard *reverse* setting (see Fig. 5.7 in IT83). A special case also arises for the cubic space group $Pa\bar{3}$ (No. 205). In the standard representation of IT83, there is an *a*-glide plane perpendicular to the basis vector **a** and a *c*-glide plane perpendicular to the basis vector **b**. However, the matrix $\mathbf{M}'$ can produce a representation with a *c*-glide plane perpendicular to **a** and an *a*-glide plane perpendicular to **b**.

All these problems are easily detected and corrected by multiplying $\mathbf{M}'$ by a correction matrix $\mathbf{M}_c$ listed in Table 6. It would be possible to avoid these problems in the first place by making step (*c*) more sophisticated, but the overall algorithm would be more complex.

5.5. *Step (e): determination of the origin shift (**I**, **c**)*

Let $\mathbf{G}_T$ be the group of symmetry operations in one of the 230 standard settings of IT83, and let $\mathbf{G}_C$ be the group of symmetry operations obtained by transforming $\mathbf{G}_O$ with $(\mathbf{C}, 0)$ as obtained through steps (*a*)–(*d*). If the number of symmetry operations in $\mathbf{G}_T$ and $\mathbf{G}_C$ are equal, and if the rotation part of each symmetry operation in $\mathbf{G}_T$ can be matched with an operation in $\mathbf{G}_C$, the following algorithm for the determination of $(\mathbf{I}, \mathbf{c})$ is attempted.

For computing the origin shift, it is sufficient to find an $(\mathbf{I}, \mathbf{c})$ that transforms the generators of $\mathbf{G}_C$ to the generators of $\mathbf{G}_T$. As was shown by various authors (for example, Boisen & Gibbs, 1990), any space group can be generated from at most three generators plus centring operations. Noncentrosymmetric space groups need at most two generators.

The first step of the algorithm is therefore to search the space-group structure for up to two generators. The choice of generators is based on the crystal system as shown in Table 7. The crystal system itself is derived from the point group as determined in step (*b*). For $|N| > 2$, the symmetry operations with a positive sense of rotation are chosen. For trigonal space groups, only one of the twofold axes in Table 7 can be present. For cubic space groups, the fourfold axis is used if present, otherwise the twofold is used.

The list of generators is built for both $\mathbf{G}_T$ and $\mathbf{G}_C$. For centrosymmetric space groups, the inversion operations are added to both lists of generators, and the original one or two generators are converted to proper rotations through multiplication by the inversion operation if necessary. After these manipulations, the rotation parts in both lists must be equal. However, due to the origin shift $(\mathbf{I}, \mathbf{c})$ to be determined, the translation parts in general are not equal.

The next step is to transform the symmetry operations in both generator lists to a primitive setting. The transformation matrix $(\mathbf{P}_g, 0)$ for this purpose is again obtained with the algorithm of step (*a*). Let *ng* be the number of operations in the generator lists, and let $(\mathbf{W}_T, \mathbf{w}_T)_j$ and $(\mathbf{W}_C, \mathbf{w}_C)_j$, $j = 1, \ldots, ng$, be the operations in the generator list for $\mathbf{G}_T$ and $\mathbf{G}_C$, respectively, in the primitive setting. Since the same transformation matrix is used for both generator lists, the rotation parts must still be equal, and it is convenient to set $\mathbf{W}_j = \mathbf{W}_{T,j} = \mathbf{W}_{C,j}$.

The origin shift $(\mathbf{I}, \mathbf{c}_p)$ in the primitive setting can be found by solving the equation $(\mathbf{I}, \mathbf{c}_p)(\mathbf{W}_j, \mathbf{w}_{C,j})(\mathbf{I}, -\mathbf{c}_p) = (\mathbf{W}_j, \mathbf{w}_{T,j}) (\mathrm{mod}\ \mathbb{Z})$ simultaneously for all *j*. The condition '(mod $\mathbb{Z}$)' has to be introduced because the symmetry operations $(\mathbf{W}, \mathbf{w})$ and $(\mathbf{W}, \mathbf{w} + \mathbf{u})$, $\mathbf{u} = [u_1, u_2, u_3]^t$, $u_i \in \mathbb{Z}$, can, in certain space groups, have different characters. For example, in space group $P31m$ (No. 157), the operation $\bar{x}, \bar{x} + y, z$ is a mirror plane (perpendicular to $[2, 1, 0]^T$), but $\bar{x} + 1, \bar{x} + y, z$ is a *b*-glide plane.

The previous equation can be rearranged as follows,

$$\begin{aligned}
&(\mathbf{I}, \mathbf{c}_p)(\mathbf{W}_j, \mathbf{w}_{C,j})(\mathbf{I}, -\mathbf{c}_p) = (\mathbf{W}_j, \mathbf{w}_{T,j}) \,(\mathrm{mod}\ \mathbb{Z}) \\
&\quad \Leftrightarrow (\mathbf{W}_j, \mathbf{w}_{C,j} + \mathbf{c}_p)(\mathbf{I}, -\mathbf{c}_p) = (\mathbf{W}_j, \mathbf{w}_{T,j}) \,(\mathrm{mod}\ \mathbb{Z}) \\
&\quad \Leftrightarrow (\mathbf{W}_j, \mathbf{w}_{C,j} + \mathbf{c}_p - \mathbf{W}_j\mathbf{c}_p) = (\mathbf{W}_j, \mathbf{w}_{T,j}) \,(\mathrm{mod}\ \mathbb{Z}) \\
&\quad \Rightarrow \mathbf{w}_{C,j} - (\mathbf{W}_j - \mathbf{I})\mathbf{c}_p = \mathbf{w}_{T,j} \,(\mathrm{mod}\ \mathbb{Z}) \\
&\quad \Leftrightarrow (\mathbf{W}_j - \mathbf{I})\mathbf{c}_p = \mathbf{w}_{C,j} - \mathbf{w}_{T,j} \,(\mathrm{mod}\ \mathbb{Z}). \qquad (12)
\end{aligned}$$

The resulting equation system has $nr = 3ng$ rows (known equations) and $nc = 3$ columns (unknown variables). For example, for the maximum $ng = 3$, the design of the (9×3) system is

$$\mathbf{M} \cdot \mathbf{c}_p = \begin{bmatrix} (W_1 - I) \\ (W_2 - I) \\ (W_3 - I) \end{bmatrix} \mathbf{c}_p = \begin{bmatrix} w_{C,1} - w_{T,1} \\ w_{C,2} - w_{T,2} \\ w_{C,3} - w_{T,3} \end{bmatrix} = b \,(\mathrm{mod}\ \mathbb{Z}). \qquad (13)$$

As pointed out recently by Eick *et al.* (1997), the system $\mathbf{M} \cdot \mathbf{c}_p = \mathbf{b} \,(\mathrm{mod}\ \mathbb{Z})$ can be solved by computing the *Smith normal form* $\mathbf{D} = \mathbf{PMQ}$ [see also, for example, Cohen (1993) or Sims (1994)]. $\mathbf{D}$ is a matrix in diagonal form with diagonal entries $d_1, \ldots, d_n$.

The algorithm for the computation of the Smith normal form was also taken from the *CrystGAP* package. It is the first part of the function '*SolveInhomModZ*'. The proof that the Smith normal

form for a given matrix is unique, the theoretical background leading to the algorithm and the proof that the algorithm always terminates are quite sophisticated, but the recipe for the algorithm is very simple and shown below for the sake of completeness:

M is the $(m \times n)$ matrix to be converted to the Smith normal form.

(*a*) Initialize: **P** := $(m \times m)$ identity matrix, **Q** := $(n \times n)$ identity matrix.

(*b*) **M** := row echelon form of **M**, apply row operations also to **P**.

(*c*) If **M** is a diagonal matrix: End.

(*d*) **M** := $\mathbf{M}^T$.

(*b*) **M** := row echelon form of **M**, apply row operations also to **Q**.

(*c*) If **M** is a diagonal matrix: End.

(*d*) **M** := $\mathbf{M}^T$, go to step (*b*).

The system $\mathbf{Dx} = \mathbf{Pb} = \mathbf{v}$, $\mathbf{v} = [v_1, \ldots, v_n]$ has solutions (mod $\mathbb{Z}$) if and only if $v_i = 0$ whenever $d_i = 0$. If this condition is true, the solutions for $\mathbf{Dx} = \mathbf{Pb}$ are described by

$$x_i \in \left\{0, \frac{1}{d_i}, \ldots, \frac{d_i - 1}{d_i}\right\} + \frac{v_i}{d_i} \qquad \text{if} \quad d_i \neq 0 \qquad (14)$$

and $x_i \in \mathbb{Q}$ otherwise.

For the purpose of computing $\mathbf{c}_p$, if solutions exist, the components of **x** are set to $x_i = v_i/d_i$ if $d_i \neq 0$, and $x_i = 0$ otherwise. The other solutions from the set are not needed. Finally, $\mathbf{c}_p$ is obtained *via* $\mathbf{c}_p = \mathbf{x} \cdot \mathbf{Q}$, and the origin shift (**I**, **c**) is obtained as $\mathbf{c} = \mathbf{P}_g^{-1} \cdot \mathbf{c}_p$. This completes the construction of the change-of-basis matrix (**C**, **c**).

Remark. The transformation matrices (**C**, **c**) produced in this way are not, in general, unique. For example, any combination of (**C**, **c**) and an operation of the Euclidean normalizer (Koch & Fischer, 1983) of the given space-group representation is again a valid transformation matrix.

5.6. *Example of space-group type determination*

This example is based on a posting by Lutz (1997) to the sci.techniques.xtallography newsgroup. Lutz solved and refined a crystal structure in the noncentrosymmetric space group $P4_2bc$ (No. 106). The 'calc *MISSYM*' routine (Le Page, 1987) of the *PLATON* program package (Spek, 1990) found approximate centres of inversion at the following positions:

(1) 0.000 0.250 0.000
(2) 0.250 0.000 0.250
(3) 0.250 0.000 0.000
(4) 0.000 0.250 0.250.

To determine the new space-group type, the symmetry operations corresponding to this list are added to the operations of space group $P4_2bc$. The additional symmetry operations in Jones Faithful notation are

(1) $-x, -y + 1/2, -z$
(2) $-x + 1/2, -y, -z + 1/2$
(3) $-x + 1/2, -y, -z$
(4) $-x, -y + 1/2, -z + 1/2$.

The expanded space group is, of course, centrosymmetric, has four centring vectors and eight entries in the list of representative symmetry matrices in the space-group structure. Hence the order of this space-group representation is 32. The centring vectors are

(1) [0, 0, 0]
(2) [1/2, 1/2, 0]
(3) [1/2, 1/2, 1/2]
(4) [0, 0, 1/2].

Note that this is not one of the commonly used centring types (*A*, *B*, *C*, *I*, *R*, *F*).

The matrix (**P**, 0) obtained with step (*a*) is $2z, x - y, x + y$. The point-group type determined with step (*b*) is 4/*mmm*. The matrix (**M**′, 0) obtained with step (*c*) is $1/2y + 1/2z, -1/2y + 1/2z, x$. Application of the transformation (**M**′, 0)(**P**, 0) results in a *C*-centred setting. Therefore step (*d*) is needed to construct the adjusted matrix (**M**, 0) = $z, y, -x$. Application of

$$(\mathbf{C}, 0) = (\mathbf{M}, 0)(\mathbf{P}, 0) = x + y, x - y + 1/2, -2z$$

now produces a primitive setting. Finally, the determination of the origin shift (**I**, **c**) in step (*e*) succeeds for space group *P*4/*nmm* (No. 129), and the resulting change-of-basis matrix is

$$(\mathbf{C}, \mathbf{c}) = x + y, x - y + 1/2, -2z.$$

This matrix can immediately be used to transform the atomic coordinates of the refined structure.

6. Generation of structure seminvariant vectors and moduli

Structure seminvariant (s.s.) vectors and moduli are a description of *permissible* or *allowed origin shifts*. A detailed treatment was published, for example, by Giacovazzo (1993). If the origin of the basis for a given space-group representation is shifted by an allowed origin shift $\mathbf{s}_k$, the symmetry environment of the old and the new origin is identical. Allowed origin shift $\mathbf{s}_k$ can be obtained as solutions of the system $(\mathbf{W}_j - \mathbf{I}) \cdot \mathbf{s}_k = 0$ (mod $\mathbb{Z}$) for all $\mathbf{W}_j$ simultaneously, where j runs over all symmetry operations of a *primitive setting* of the space group under consideration. $\mathbf{s}_k$ can be divided into two groups: *continuous* and *discrete* allowed origin shifts. For example, in space group *P*2 (No. 3), the origin can be shifted arbitrarily parallel to the twofold axis without

changing its symmetry environment. This corresponds to a continuous allowed origin shift. In the same space group, the origin can also be shifted by $[1/2, 0, 0]^T$, $[0, 1/2, 0]^T$ and all linear combinations of these vectors. These are the discrete allowed origin shifts. Other trivial discrete allowed origin shifts correspond to the centring vectors, if present, and the unit translations or their multiples.

Allowed origin shifts can be conveniently represented by a set of s.s. vectors and moduli. The mathematical connection between the allowed shifts and the s.s. vectors and moduli is explained in detail by Giacovazzo (1993). The resulting recipe is very simple. A s.s. vector representing continuous shifts is identical to the shift vector, and the modulus is 0. In the case of space group $P2$ for example, the continuous shifts along the twofold axis are represented by the vector $v_1 = [0, 1, 0]^T$ and the modulus $m_1 = 0$. The discrete shift $[1/2, 0, 0]^T$ is represented by $v_2 = [1, 0, 0]^T$ and $m_2 = 2$. Similarly, the shift $[0, 0, 1/2]^T$ is represented by $v_3 = [0, 0, 1]^T$ and $m_3 = 2$. This means that in order to obtain an integer vector the discrete shift vector is multiplied by the least common multiple of the denominators of its components, and the least common multiple becomes the modulus.

The number of *non-redundant* s.s. vectors and moduli can range from 0 (for example, space group $Im\bar{3}m$) to 3 (for example, space group $P2$). In general there are several choices for the non-redundant set. In the case of space group $P2$, for example, v_3 could be replaced by $[1, 0, 1]^T$ with modulus 2, corresponding to the vector $[1/2, 0, 1/2]$, which is a linear combination of $[1/2, 0, 0]^T$ and $[0, 0, 1/2]^T$. Much of the algorithm presented below is concerned with finding a concise non-redundant set.

The main steps of the s.s. determination algorithm are as follows:

(*a*) Determination of the point group.

(*b*) Point-group-specific selection of space-group generators.

(*c*) Determination of the continous allowed origin shifts (s.s. vectors v_i with moduli $m_i = 0$).

(*d*) Construction of the matrix $(\mathbf{P}, 0)$ which transforms $\mathbf{G}_o$ to $\mathbf{G}_p$, where $\mathbf{G}_o$ is the set of symmetry operations in the original setting and $\mathbf{G}_p$ is the set of symmetry operations in a primitive setting.

(*e*) Determination of the discrete allowed origin shifts (s.s. vectors $\mathbf{v}_i$ and moduli m_i with $m_i \neq 0$).

The algorithms used for steps (*a*) and (*d*) have already been presented above. The other steps are explained in detail below.

6.1. *Step (b): point-group-specific selection of space-group generators*

This step is very similar to step (*c*) of the determination of the space-group type. However, the axis directions of the generators do not, in general, coincide with basis vectors or simple linear combinations of basis vectors as before. Therefore the selection process has to be slightly more general.

For point group 1, the only generator selected is the identity matrix. For point group $\bar{1}$, the only generator selected is the centre of symmetry. All other point groups can be treated based on the crystal system.

In the monoclinic system, there are only $n_M = 2$ matrices in the list of representative symmetry matrices, namely the identity matrix and a matrix with rotation-part type $|N| = 2$ (twofold axis or a mirror plane). The latter is selected as the first generator. For centrosymmetric space groups, the centre of symmetry is selected as the second generator.

The situation is still very simple in the orthorhombic case. Of the $n_M = 4$ entries in the list of representative symmetry matrices, one is the identity matrix. Any two of the other three matrices can be selected as generators. If a centre of symmetry exists, it is selected as the third generator.

The tetragonal, trigonal and hexagonal cases can be treated in a uniform way. To find the first generator, the rotation-part type N_1 and axis direction $\mathbf{e}_1$ are determined for each entry in the list of representative symmetry matrices until the principal axis ($|N_1| = 4$, 3 or 6, respectively) is found. Next, the list of rotation parts is scanned for a matrix with $|N_2| = 2$ and $\mathbf{e}_2 \neq \mathbf{e}_1$. If such a matrix can be found (for example, for point groups 422, $3m$ *etc.*), it is used as the second generator. For centrosymmetric space groups, the centre of symmetry is selected as the second or third generator.

In the cubic case, n_M is either 12 or 48. To find the first generator, the list of rotation parts is scanned for a matrix with $|N_1| = 3$. The second generator is found by scanning for a matrix with $|N_2| = n_M/6$. If a centre of symmetry exists, it is selected as the third generator.

6.2. *Step (c): determination of continuous allowed origin shifts*

Continuous allowed origin shifts can be represented by some vector $\mathbf{s}$ which can be multiplied by any real value, in particular also infinitesimally small real values. Inspection of the equation $(\mathbf{W}_j - \mathbf{I}) \cdot \mathbf{s}_k = 0 \pmod{\mathbb{Z}}$ then shows that the left-hand side must be exactly equal to zero for continuous shifts. Therefore the problem of finding the continuous shifts is equivalent to the determination of the ordinary nullspace (Strang, 1986) of all $\mathbf{W}_j - \mathbf{I}$ simultaneously.

The same argument can be applied for centred space groups, where the allowed origin shifts are determined by the solutions of the system $(\mathbf{W}_j - \mathbf{I}) \cdot \mathbf{s}_k = 0 + \alpha\mathbf{B}_v \pmod{\mathbb{Z}}$ (Giacovazzo, 1993). $\mathbf{B}_v$ are the n_Z centric vectors and $\alpha = 0$, 1. As is shown below, the determination of the discrete permissible origin shifts involves the transformation of the rotation parts to a primitive setting. This is not necessary for the continuous shifts. On the contrary, in order to obtain a concise set of s.s.

vectors, it is best to work with the rotation parts of the centred setting.

To solve the equation system $(\mathbf{W}_j - \mathbf{I}) \cdot \mathbf{s}_k = 0$ for all $\mathbf{W}_j$ simultaneously, it is sufficient to solve the smaller system with the *ng* rotation parts of the generators selected in step (*b*). Since the maximum number for *ng* is 3, the biggest system to solve is just a (9×3) matrix. For example, for $n_G = 3$, the design of the system is

$$\mathbf{M}_Z \cdot \mathbf{v}_i = \begin{bmatrix} (W_1 - I) \\ (W_2 - I) \\ (W_3 - I) \end{bmatrix} \cdot \mathbf{v}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \qquad (15)$$

Note that each 0 on the right-hand side represents a null vector with three components.

To solve the system for $\mathbf{v}_i$, $\mathbf{M}_Z$ is transformed to its row echelon form $\mathbf{U}$. For example, for space group *Cm* (No. 8) with $n_G = 1$, the matrix $\mathbf{W}_1$ is

$$\mathbf{W}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \qquad (16)$$

In this example, $\mathbf{M}_Z$ and its reduced row echelon form $\mathbf{U}$ are

$$\mathbf{M}_Z = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \mathbf{U} = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (17)$$

Here the rank of $\mathbf{U}$ is 1 and therefore the nullspace of $\mathbf{M}_Z$ is a two-dimensional plane. In general, the dimensionality d_N of the nullspace of $\mathbf{M}_Z$ is 3 minus the rank of $\mathbf{U}$ and can range from 0 (for example, space group $P\bar{1}$) to 3 (space group *P*1).

The d_N vectors spanning the nullspace of $\mathbf{M}_Z$ are now determined as the solution of $\mathbf{U} \cdot \mathbf{s}_k = 0$. For $d_N = 3$, the three most concise solutions are simply $[1, 0, 0]^T$, $[0, 1, 0]^T$ and $[0, 0, 1]^T$. For $d_N = 1$, the algorithm for the determination of axes' directions can be reused. For $d_N = 2$, four solution vectors to the system $\mathbf{U} \cdot \mathbf{s}_k = 0$ are obtained by assigning, in turn, the pairs of values $[1, 0]$, $[0, 1]$, $[1, 1]$ and $[1, -1]$ to the free components [compare with the solution of $\mathbf{S} \cdot \mathbf{x} = 0$ in step (*c*) of the space-group type determination]. The third component for each of the four solution vectors is computed by backsubstitution in $\mathbf{U}$. The four resulting vectors are sorted based on the squared 'length', and the two shortest vectors are taken as the basis for the nullspace.

The word 'length' was put in quotes because a length is, in general, only defined for a given metric. However, for the purpose of finding the s.s. vectors with modulus 0, it would be very artificial and also unnecessary to require the knowledge of a metric. In this case, it is sufficient to assume a Cartesian basis.

6.3. *Step (e): determination of the s.s. vectors* $\boldsymbol{v}_i$ *and moduli* m_i *with* $m_i \neq 0$

To find the discrete allowed origin shifts, the system $(\mathbf{W}_j - \mathbf{I}) \cdot \mathbf{s}_k = 0 \pmod{\mathbb{Z}}$ has to be solved for $\mathbf{W}_j$ in a primitive setting of the space group. Therefore the generators selected in step (*b*) are transformed with the change-of-basis matrix determined in step (*d*). The transformed $\mathbf{W}_j^P$ are used in the same way as before in step (*b*) to construct a system $\mathbf{M}_p \cdot \mathbf{v}_i = 0 \pmod{\mathbb{Z}}$. For example, if *ng* = 3,

$$\mathbf{M}_p \cdot \mathbf{v}_i = \begin{bmatrix} (W_1^P - I) \\ (W_2^P - I) \\ (W_3^P - I) \end{bmatrix} \cdot \mathbf{v}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} (\text{mod } \mathbb{Z}). \qquad (18)$$

The solutions to this system are obtained by evaluating the Smith normal form $\mathbf{D} = \mathbf{PM_PQ}$ as explained in step (*e*) of the determination of the space-group type. However, this time more than one solution is needed. To obtain the *group of all discrete allowed origin shifts* (mod $\mathbb{Z}$), the d_i solutions of equation (14) are computed for each $d_i \neq 0$. Each of the shift vectors $\mathbf{s}_k = \mathbf{x} \cdot \mathbf{Q}$ obtained in this way is added to the group which is then expanded by pairwise addition as explained in the first part of this paper. The maximum order of a group of discrete allowed origin shifts is 8, corresponding to space group $P\bar{1}$.

The smallest number of shifts that generate the whole group can be used to obtain a non-redundant set of s.s. vectors and moduli representing the discrete shifts. Because the maximum order of the group is just 8, searching for this set can be performed with a straightforward trial-and-error algorithm. First, it is checked if the group can be generated from just one vector. If not, it is checked if the group can be generated with a pair of vectors. If this is also not possible, three vectors that generate the group are selected. Of course, this is always possible. Finally, the group-generating set of discrete shifts found in this way is transformed back to the centred setting, if necessary.

At this point, the problem of finding a non-redundant set of s.s. vectors and moduli is mathematically solved. Simply taking together the continuous and discrete shifts produces a solution. However, there are, in general, several possibilities for selecting group-generating discrete shifts. Further choices arise for centred space groups, where the centring vectors also belong to the group of discrete allowed origin shifts. Selecting the 'most concise' solution requires some extra effort. The selection algorithm used is not explained in detail, because it is mainly a cosmetic measure. The main idea is that all vectors in the group of discrete shifts are transformed to the centred setting, where they are combined with all centring vectors and continuous shifts. The latter also applies to primitive space groups. For each discrete shift in the primitive setting, the 'best' vector resulting from these combinations is selected,

based on the 'length' [as introduced before in step (*c*)] and the least common multiple of the denominators of the components (corresponding to the s.s. modulus). Before the shifts in the primitive setting are searched for a subset of generators as explained above, they are sorted based on the 'length' of the corresponding best vector in the centred setting. This is performed because the vectors in the centred setting will finally become visible.

To give a simple example, with the measures just described, the s.s. vector and modulus for the standard setting of space group *F*222 is $v_1 = [1, 1, 1]^T$, $m_1 = 4$. Without the measures it is $v_1 = [1, 3, 3]^T$, $m_1 = 4$. In general, the extra effort spent is rewarded by s.s. vectors and moduli which are equal to those tabulated by Giacovazzo (1993), and also results in 'sensible' choices for alternative settings which are not tabulated.

7. Source code availability

The algorithms described in this paper were developed using the SgInfo library of ANSI C procedures (Grosse-Kunstleve, 1995*a*,*b*). At the point of writing, the new version of the library is available as a developmental alpha-release without documentation. In the future, the full source code and documentation will be available on the SgInfo World-Wide-Web server. The library is free for non-commercial applications.

References

Boisen, M. B. Jr & Gibbs, G. V. (1990). *Mathematical Crystallography, Reviews in Mineralogy*, Vol. 15 (revised edition). Washington, D.C.: Mineralogical Society of America.

Burzlaff, H., Böhme, R. & Gomm, M. (1977). *CRYSTAN. A Crystallographic Program System for Minicomputers.* Institut für Angewandte Physik, Universität Erlangen, Germany.

Burzlaff, H. & Hountas, A. (1982). *J. Appl. Cryst.* **15**, 464–467.

Burzlaff, H. & Zimmermann, H. (1980). *Z. Kristallogr.* **153**, 151–179.

Cohen, H. (1993). *A Course in Computational Algebraic Number Theory*, 3rd, corrected printing 1996. Berlin: Springer-Verlag.

Eick, B., Gahler, F. & Nickel, W. (1997). *Acta Cryst.* A**53**, 467–474.

Fischer, W. & Koch, E. (1983). *International Tables for Crystallography*, Vol. A, ch. 11. Dordrecht: Kluwer.

GAP (1995). *Groups, Algorithms and Programming*, 5th ed, M. Schönert *et al.* Lehrstuhl D für Mathematik, RWTH, Aachen, Germany. See also http://www-gap.dcs.st-and.ac.uk/~gap/.

Giacovazzo, C. (1993). *International Tables for Crystallography*, Vol. B, ch. 2.2. Dordrecht: Kluwer.

Grosse-Kunstleve, R. W. (1995*a*). http://www.kristall.eth.ch/LFK/software/sginfo/.

Grosse-Kunstleve, R. W. (1995*b*). Abstract P30-02, Book of Abstracts, Sixteenth European Crystallographic Meeting (ECM16), Lund, Sweden, 6–11 August 1995. (Also available at http://www.kristall.ethz.ch/software/sginfo/Lund95/.)

Hahn, T. (1983). *International Tables for Crystallography*, Vol. A. Dordrecht: Kluwer.

Hall, S. R. (1981). *Acta Cryst.* A**37**, 517–525.

Hall, S. R. & Grosse-Kunstleve, R. W. (1999). *International Tables for Crystallography*, Vol. B. In the press.

Koch, E. & Fischer, W. (1983). *International Tables for Crystallography*, Vol. A, ch. 15. Dordrecht: Kluwer.

Kraus, W. & Nolze, G. (1996). *J. Appl. Cryst.* **29**, 301–303.

Kraus, W. & Nolze, G. (1997). http://www.bam.de/a_v/v_1/powder/g_cell.html.

Larson, A. C. (1969). *Acta Cryst.* A**25**, S1.

Larson, A. C. & von Dreele, R. B. (1995). *GSAS – General Structure Analysis System.* Los Alamos National Laboratory, NM 87545, USA.

Le Page, Y. (1987). *J. Appl. Cryst.* **20**, 264–269.

Le Page, Y. (1997). Private communication.

Lutz, M. (1997). Posting to sci.techniques.xtallography newsgroup, 7 Aug 1997.

Menzer, G. (1960). *Z. Kristallogr.* **113**, 155–177.

Shmueli, U. (1984). *Acta Cryst.* A**40**, 559–567.

Shmueli, U. (1993). *International Tables for Crystallography*, Vol. B, ch. 1.4. Dordrecht: Kluwer.

Sims, C. (1994). *Computation with Finitely Presented Groups*. Cambridge University Press.

Spek, A. L. (1990). *Acta Cryst.* A**46**, C34.

Strang, G. (1986). *Linear Algebra and its Applications.* Fort Worth: Harcourt Brace Jovanovich College.

Stróz, K. (1997). *J. Appl. Cryst.* **30**, 178–181.

Wondratscheck, H. (1983). *International Tables for Crystallography*, Vol. A, ch. 8. Dordrecht: Kluwer.

Yvon, K., Jeitschko, W. & Parthé, E. (1977). *J. Appl. Cryst.* **10**, 73–74.