Spglib: a software library for crystal symmetry search

Atsushi Togo, Kohei Shinohara, and Isao Tanaka

1 Center for Basic Research on Materials, National Institute for Materials Science, Tsukuba, Ibaraki 305-0047, Japan
2 Center for Elements Strategy Initiative for Structural Materials, Kyoto University, Sakyo, Kyoto 606-8501, Japan
3 Preferred Networks, Inc., Tokyo 100-0004, Japan
4 Department of Materials Science and Engineering, Kyoto University, Sakyo, Kyoto 606-8501, Japan
5 Nanostructures Research Laboratory, Japan Fine Ceramics Center, Atsuta, Nagoya 456-8587, Japan

A computer algorithm to search symmetries of crystal structures as implemented in the spglib code is described. An iterative algorithm is employed to robustly identify space group types tolerating a certain amount of distortion in the crystal structures. The source code is distributed under the 3-Clause BSD License, a permissive open-source software license. This paper focuses on the algorithm for identifying the space group symmetry of the crystal structures.

I. INTRODUCTION

Crystal symmetry is essential information to understand various crystal properties. It is also useful to compress information of physical states of crystals, e.g., in electronic structure calculations. The crystal symmetry is composed of a set of symmetry operations that map a crystal structure onto itself, and the set forms a space group. There are 230 space group types. Each crystal structure is uniquely assigned to one of the space-group. There are 230 space group types. Each crystal symmetry is derived simultaneously with running the FINDSYM code. After many years of development, the spglib code has been one of the key software libraries in the scientific community. Since many software packages rely on it already, the spglib code is expected to be well maintained and live longer. Therefore, this paper aims to invite developers of the spglib code by presenting a detailed algorithm implemented in the spglib code.

There already exist crystal symmetry finding codes. The FINDSYM code in the ISOTROPY Software Suite has been well known, however, the source code is unavailable in public. The cctbx code is another open-source crystallography toolbox under a BSD-type variant license. The sginfo code, currently distributed under an open source software license, has been superseded by the space group toolbox (sgtbx) in the cctbx code. The AFLWSYM code is an open-source crystal symmetry analysis code under the GNU general public license. The spglib code is another open-source code distributed under the 3-clause BSD license that is a permissive open-source software license.

Using the spglib code, space group operations are searched from a crystal structure in which small deviations of atomic positions from their ideal positions should be tolerated. As a result, we obtain the coset representatives of the space group \( S \) with respect to the translation group \( T \). Some other useful information about the crystal symmetry is derived simultaneously with running the algorithm. After many years of development, the spglib code has become one of the key software libraries in the scientific community. Since many software packages rely on it already, the spglib code is expected to be well maintained and live longer. Therefore, this paper aims to invite developers of the spglib code by presenting a detailed algorithm implemented in the spglib code.

In the algorithm outlined in this paper, we employ an iterative approach to determine space group operations for the initial input crystal structure at a given distance tolerance. The flowchart of the algorithm is presented in Fig. 1. There are four stages in the algorithm: primitive cell search, space group operation search, identification of space group type, and finalization. These stages are subdivided into smaller steps. The algorithm continues until the obtained symmetry operations satisfy all given crystallographic constraints under minimum adjustment of the tolerance value. There are nested iteration loops in the algorithm. Some loops are explicitly depicted in Fig. 1. A few more loops are contained inside the blocks. The tolerance value is decreased when the operation fails in each loop. One exception is step (e), where...
both increasing and decreasing the tolerance value are attempted.

This paper concentrates on the algorithms designed for searching space group operations and identifying the space group types. The algorithm related to the magnetic space group, which builds upon the space group algorithm, is discussed in another publication. A tolerance parameter, which is crucial for the algorithm in the spglib code, is detailed in the following section. Notations, algebra, and look-up tables utilized in this paper are compiled in the Appendices.

II. TOLERANCE PARAMETER IN CRYSTAL SYMMETRY OPERATION SEARCH

In the spglib code, numerical searches for symmetry operations of a provided input unit cell utilize a small Euclidean distance $\epsilon$ as a tolerance parameter. The value of the tolerance parameter is adjusted in the process of the algorithm to identify a possible space group. This section explains how it is employed to examine the symmetrical equivalence of two atomic points. How it is adjusted is explained in the following specific sections. Symbols used in this section are summarized in Appendices A1–A4.

A. Lattice translation to a point near origin

Each element of point coordinates, $x = (x_1, x_2, x_3)^T$, is confined within the interval $[-0.5, 0.5]$ by the operation:

$$x_i \leftarrow x_i - \lfloor x_i \rfloor, \quad i = 1, 2, 3, \quad (1)$$

and similarly for vectors,

$$w \leftarrow w - \lfloor w \rfloor. \quad (3)$$

B. Determination of equivalent points under lattice periodicity

The tolerance parameter $\epsilon$ is used to identify if two points $\tilde{x}$ and $x'$ occupy the same atomic site or not. Using the difference of positions $\Delta x = \tilde{x} - x'$, this is examined by

$$|(a, b, c)\Delta x| < \epsilon, \quad (4)$$

where $(a, b, c)$ represent the basis vectors of the unit cell. To accommodate the periodicity of the lattice, the condition (4) is reformulated using the operation (3) as

$$|(a, b, c)(\Delta x - [\Delta x])| < \epsilon. \quad (5)$$

This expression is frequently used in the implementation.
C. Examination of a symmetry operation

Given a space group operation \((W, w)\), which may or may not be a valid symmetry operation, an atomic point \(x\) is transformed to \(\tilde{x} = (W, w)x\). If \((W, w)\) represents a valid space group operation, \(\tilde{x}\) must be located at one of atomic sites \(x'\) with the same atomic type. In the algorithm, this is examined for all atoms in the input unit cell or the primitive cell using condition (5). When this is satisfied, the given space group operation \((W, w)\) is accepted.

III. MACHINE PRECISION ISSUE

It is assumed that the spglib code is used on 32- and 64-bit computer systems. As reported in Ref. [6], comparisons of lengths have to be carefully implemented in the code due to finite machine precision. For example, an inequality \(x < y\) may be implemented in \(x < y - e'\) with a small positive value \(e'\). Most of the inequalities that appear in the implementation are in the style of Eq. (5) and none of the special care is applied for this case. On the other hand, the operation to take modulo by \(x\) is performed using the above inequality using \(e'\). With this operation, a value \(x_i\) is converted to fall within the interval \([-e',1-e']\). \(e' = 10^{-10}\) is employed in the current version of the spglib code.

IV. PRIMITIVE CELL SEARCH

In the first stage of the space group symmetry search, a primitive cell is determined from lattice points in the input unit cell. The lattice points are obtained through pure translation operation search.

A. Step (a): Searching pure translation operations

The input unit cell contains multiple lattice points in it if it is not a primitive cell. These lattice points are obtained as translation parts \(w_I\) of pure translation operations of \(\{(I, w_I)\}\) in the input unit cell, where \(I\) is the identity matrix. The pure translation operations of \(\{(I, w_I)\}\) are searched as follows. Candidates of the translation parts \(w_I^c\) are selected from vectors that extend from a fixed atomic site \(a\) to all atomic sites \(x'\) of the same atomic type \(A\). i.e., \(w_I^c = x' - x_A\). To minimize computational demand, the fixed atomic site is chosen among atoms having an atomic type that comprises the smallest number of atoms. Each candidate vector \(w_I^c\) is examined as described in Sec. [11]. If all \(\Delta x_A = (I, w_I^c)x_A - x'_A\) for all \(x_A\) satisfy the condition (5), this \((I, w_I^c)\) is a pure translation operation.

If the input unit cell is a primitive cell, only one \((I, w_I)\) with \(w_I = (0,0,0)^\top\) should be found, otherwise a set of multiple pure translation operations of \(\{(I, w_I)\}\) are obtained.

In typical use cases, this step is the most computationally demanding part in the entire process. The brute-force algorithm has a time complexity of \(\mathcal{O}(N^3)\), with \(N\) denoting the number of type \(A\) atoms. However, this complexity is empirically reduced to \(\mathcal{O}(N^2 \log N)\) by sorting the atoms, which minimizes the worst-case scenarios.

B. Step (b): Choosing basis vectors of primitive cell

Candidates for three basis vectors of a primitive cell \((a_p^c, b_p^c, c_p^c)\) are chosen from the set of vectors \(T_1 \cup T_p\), where \(T_1 = \{a_i, b_i, c_i\}\) is the set composed of the basis vectors of the input unit cell, and \(T_p = \{w_I\}\) found at step (a). The three basis vectors are chosen to create a right-handed coordinate system. The volume of the primitive cell, \(V_p\), is expected to be approximately the volume of the input unit cell, \(V_i\), divided by \(|T_p|\). Therefore, the basis vectors of the primitive cell are searched under the condition:

\[
|T_p| = |V_i/V_p|. \tag{6}
\]

C. Step (c): Failure of finding primitive cell basis vectors

For distorted input unit cells, the condition (6) may not always be satisfied. For example, the number of the pure translations found can either be more than or less than those expected. In this case, the sequence of steps (b), (c), and (d) is iterated by reducing the tolerance value. If this loop is repeated many times, the procedure restarts from step (a) with the tolerance value reduced from that previously used in step (a).

D. Step (d): Thinning out pure translations

Some of the pure translations that do not satisfy the condition (5) are discarded by re-examining the existing pure translations with a tightened tolerance value. Typically, this operation is far more computationally efficient than restarting from step (a) with a tightened tolerance value.

E. Step (e): Creating a primitive cell

The primitive cell basis vectors \((a_p^c, b_p^c, c_p^c)\) found at step (b) are transformed to a different set of primitive cell basis vectors \((a_p, b_p, c_p)\) by the Delaunay reduction. This transformation is written as

\[
(a_p, b_p, c_p) = (a_p^c, b_p^c, c_p^c)Q_D. \tag{7}
\]
Q, is an integer matrix and is chosen such that det(Q) = 1.

Similarly, the transformation of the basis vectors of the primitive cell basis vectors to those of the input unit cell \((a_i, b_i, c_i)\) in a right handed coordinate system is written by the change-of-basis matrix \(Q_{p-i}\) as

\[
(a_i, b_i, c_i) \approx (a_p, b_p, c_p)Q_{p-i}
\]

(8)

where \(Q_{p-i}\) is chosen as an integer matrix with \(\det(Q_{p-i}) \geq 1\) although Eq. (8) is an approximation if \(T_p = \{w\}\) found at step (a) is distorted with respect to the lattice of the primitive cell.

Point coordinates in the input unit cell, \(x_i\), are transformed to their corresponding coordinates in the primitive cell, \(x_{p*}\), by

\[
x_{p*} = Q_{p-i}x_i.
\]

(9)

where \(x_{p*}\) is brought in the interval \([-0.5, 0.5]\) by the operation \([\mathbb{I}]\). When \(\det(Q_{p-i}) > 1\), multiple \(x_i\) should be mapped to a point \(x_{p*}\) with respect to \((a_p, b_p, c_p)\) using condition \([\mathbb{I}]\). The tolerance value \(\epsilon\) in condition \([\mathbb{I}]\) is adjusted until the multiplicity becomes \(\det(Q_{p-i})\) for all translationally independent \(x_{p*}\). If this fails, the procedure restarts from step (a) with the tolerance value tightened from that previously used at step (a).

Upon successful verification, the point coordinates of translationally equivalent atoms are averaged to produce \(x_p\) along with boundary treatment for \(x_{p*}\) located close to the boundary of \([-0.5, 0.5]\). Finally, a primitive cell is created from \((a_p, b_p, c_p)\) and \(x_p\).

V. SPACE GROUP OPERATION SEARCH

The purpose of the second stage is to search a set of space group operations \(\{(W_p, w_p)\}\) of the primitive cell obtained in the first stage. Candidates of the rotation parts \(\{W\}\) are given by an exhaustive search of lattice point group operations. Using obtained \(\{W\}\) and the point coordinates \(x_p\), a set of the space group operations \(\{(W_p, w_p)\}\) is searched.

A. Step (f): Searching lattice point group operations

A set of possible candidates of lattice point group operations \(\{W_p\}\) is exhaustively generated by filling the matrix elements with -1, 0, or 1 under the constraint of \(|\det(W_p)| = 1\). As described in Appendix A3, a set of lattice point group operations of the primitive cell, \(\{W_L\}\), is searched within \(\{W_p\}\) using the metric tensor \(G\) that is rotated by \(W_L\) as \(\hat{G} = (W_L)^\dagger GW_L\).

Comparison of \(G\) and \(\hat{G}\) is performed as follows. The diagonal elements of the matrix \(\hat{G}\) provide information about the lengths of the basis vectors, while the off-diagonal elements indicate the angles between these vectors. The differences in lengths can be straightforwardly compared using the tolerance \(\epsilon\) based on Euclidean distance. For angle comparison, an angle tolerance parameter can be employed. However, this addition of an extra tolerance parameter may complicate the usage. Therefore, in the current implementation, if an angle tolerance is not explicitly specified, the distance tolerance \(\epsilon\) is approximated for angle comparisons. This approach is applied, for instance, in the evaluation of \(G_{12}\), as shown in the following equation:

\[
\sin|\Delta\theta| \sqrt{(\|a\| + \|b\|)(\|b\| + \|c\|)} < \epsilon,
\]

(10)

where \(\Delta\theta\) is the angle difference between the two pairs of vectors \(a - b\) and \(\hat{a} - \hat{b}\),

\[
\Delta\theta = \arccos \left( \frac{\hat{a} \cdot \hat{b}}{\|\hat{a}\|\|\hat{b}\|} \right) - \arccos \left( \frac{a \cdot b}{\|a\|\|b\|} \right).
\]

The angle difference \(\Delta\theta\) is compared with the tolerance value with the averaged lengths of basis vectors. The left-hand side of Eq. (10) is given by the matrix elements of \(G\) and \(\hat{G}\) such as \(a \cdot b = \sqrt{G_{12}}, \|a\| = \sqrt{G_{41}},\) and \(\|b\| = \sqrt{G_{22}}\).

In a similar way to that applied in step (a), the corresponding translation part \(w_p\) is searched using \(W_L\) instead of \(I\) in step (a).

B. Step (g): Searching space group operations

A set of the space group operations \(\{(W_p, w_p)\}\) is searched in the following way. The rotation matrices \(\{W_L\}\) found in step (f) are used as candidates of the rotation parts of \(\{(W_p, w_p)\}\). From Eq. (A5), a space group operation \((W_p, w_p)\) satisfies \(\bar{x} = W_p x + w_p\). Therefore, candidates of translation parts for \(W_L\) are given by \(w^c_p = \bar{x} - W_L x\) over possible combinations of \(x\) and \(\bar{x}\).

To limit its search space, for a fixed \(x\), \(\bar{x}\) is selected from the all atoms with the same atomic type as \(x\) in the primitive cell. Then, \((W_L, w^c_p)\) is examined by applying condition \([\mathbb{I}]\) with \(\Delta x = (W_L, w^c_p)x - \bar{x}\) for all \(x\) and \(\bar{x}\). If none of \(w^c_p\) is found, this \(W_L\) is discarded, otherwise \((W_L, w^c_p)\) is adopted as \((W_p, w_p)\). This procedure is repeated over all elements of \(\{W_L\}\). Finally, a set of the space group operations \(\{(W_p, w_p)\}\) is obtained. In the next stage, it is verified that \(\{(W_p, w_p)\}\) constitutes the coset representatives of the factor group \(S/T\).

VI. IDENTIFICATION OF SPACE GROUP TYPE

In this third stage, a space group type is identified by comparing the set of the space group operations \(\{(W_p, w_p)\}\) obtained in the last stage with those sets
coded in the Hall symbols[16,17]. To achieve this, the primitive cell is transformed into the corresponding conventional unit cell in a specific setting. The space group operations for this setting are matched with those matrix representations decoded from the Hall symbols and the origin shift is simultaneously determined at the matching process. The algorithm presented in this section follows almost exactly as that reported by Grosse-Kunstleve and Adams in Ref. 8, and it is described as implemented in the spglib code.

A. Step (h): Identify crystallographic point group

The crystallographic point group is given by collecting rotation parts of the space group operations, i.e., \( P = \{ W_p | (W_p, w_p) \in \{(W_p, w_p) \} \} \). The crystallographic point group type is identified from the traces and determinants of the matrices of \( (W_p) \) by using the look-up Tables V and VI presented in Appendix I. If the identification of the crystallographic point group type failed, the procedure restarts from step (a) with the tolerance value tightened from that previously used in step (a).

B. Step (i): Transformation from primitive cell to conventional unit cell

Laue class is the information necessary to transform the basis vectors of the primitive cell to those of the corresponding conventional unit cell. It is easily known once the crystal class, which is equivalent to the crystallographic point group type, is determined as shown in Table IV. The transformation of the basis vectors of the primitive cell \((a_p, b_p, c_p)\) to those of the conventional unit cell \((a_c, b_c, c_c)\) is written as

\[
(a_c, b_c, c_c) = (a_p, b_p, c_p)M'.
\]

(12)

\(M'\) is constructed from three axis directions \((e_x, e_y, e_z)\) such as

\[
M' = \begin{bmatrix}
  e_{x1} & e_{y1} & e_{z1} \\
  e_{x2} & e_{y2} & e_{z2} \\
  e_{x3} & e_{y3} & e_{z3}
\end{bmatrix},
\]

(13)

with \(\det(M') > 0\). These axis directions are determined from rotation axes that characterize the Laue class. The rotation axis direction of each \(W_p\) is found by solving the following equation:

\[
W_{prop}^p e = e,
\]

(14)

where \(W_{prop}^p = \det(W_p)W_p\) is the proper rotation matrix of \(W_p\). The rotation axis direction \(e\) can be determined through an exhaustive search, wherein three integer values are evaluated as the components of \(e\). The rotation order \(n\) of \(W_p\) is defined by the smallest \(n > 0\) that satisfies

\[
(W_{prop}^p)^n = I.
\]

(15)

Except for the Laue class \(\bar{1}\), the primary axis direction \(e^{pri}\) is determined by selecting a primary proper rotation matrix \(W^{pri}\) of the rotation order \(n^{pri}\) presented in Table IV. The axis direction \(e'\) perpendicular to \(e^{pri}\) is determined to satisfy the following equation:

\[
Se' = 0,
\]

(16)

where \(S = \sum_i n_i W_i^{pri}\). The conditions that the primary, secondary \((e^{sec})\), and ternary \((e^{ter})\) axis directions have to satisfy are listed in Table IV for Laue classes. For the Laue class \(1\), \(M'\) is determined so as to make the left-hand side of Eq. 12 become the Niggli cell. For the Laue class \(2/m\), the \(b\) axis direction is set as \(e^{por}\) along the two fold rotation axis by Eq. 16. Therefore \((e_x, e_y, e_z) = (e^{ter}, e^{sec})\). For the Laue classes of \(4/m, 4/mmm, 3, 3m, 6/m,\) and \(6/mmm\), the \(c\) axis direction is set as \(e^{sec}\) along the four or three fold rotation axis by Eq. 14. \(e^{sec}\) is found to be perpendicular to \(e^{ter}\) by Eq. 16 and \(e^{sec} = W^{pri} e^{sec}\). Therefore \((e_x, e_y, e_z) = (e^{sec}, e^{ter}, e^{pri})\). Among possible sets of \((e^{sec}, e^{ter}, e^{pri})\), one having the smallest \(\det(M')\) \(\neq 0\) is selected to avoid wrongly-centred \((a_c, b_c, c_c)\). For the Laue classes of \(mmm, m3, m3\), three axis directions along two or four-fold rotation axes are determined by Eq. 14. When \(\det(M') < 0\), the secondary and ternary axis directions are swapped to make the system of basis vectors right-handed.

For convenience in the following steps, the basis vectors are further transformed to have a specific centering type by multiplying a correction matrix \(M\) with \(M'\) for the Laue classes of \(2/m, mmm\), and the rhombohedral system. Otherwise, \(M\) is considered as an identity matrix.

### Table I. Axis directions for Laue classes. \(n^{pri} \), \(n^{sec}\), and \(n^{ter}\) are the rotation orders of the primary (\(W^{pri}\)), secondary (\(W^{sec}\)), and ternary (\(W^{ter}\)) proper rotation matrices, respectively.

| Laue class | Condition implemented in the spglib code | Do nothing |
|------------|---------------------------------------|------------|
| \(2/m\)   | \(n^{pri} = 2\) for \(a_c, b_c\), \(e^{sec} = 0\), \(e^{ter} = 0\), \(e^{ter} \neq e^{sec}\) | \(n^{pri} = 2\) for all \(a_c, b_c, c_c\) |
| \(4/m\)   | \(n^{pri} = 4\) for \(c_c\), \(e^{sec} = 0\), \(e^{ter} = W^{pri} e^{ter}\) | \(n^{pri} = 4\) for all \(a_c, b_c, c_c\) |
| \(4/mmm\) | Same as \(4/m\) | \(n^{pri} = 4\) for all \(a_c, b_c, c_c\) |
| \(3\)     | \(n^{pri} = 3\) for \(c_c\), \(e^{sec} = 0\), \(e^{ter} = W^{pri} e^{sec}\) | \(n^{pri} = 3\) for all \(a_c, b_c, c_c\) |
| \(3m\)    | Same as \(3\) | \(n^{pri} = 3\) for all \(a_c, b_c, c_c\) |
| \(6/m\)   | Same as \(3\) | \(n^{pri} = 3\) for all \(a_c, b_c, c_c\) |
| \(6/mmm\) | Same as \(3\) | \(n^{pri} = 3\) for all \(a_c, b_c, c_c\) |
| \(mmm\)   | \(n^{pri} = n^{sec} = n^{ter} = 2\) for \(a_c, b_c, c_c\) | \(n^{pri} = 2\) for all \(a_c, b_c, c_c\) |
| \(m3\)    | Same as \(mmm\) | \(n^{pri} = 3\) for all \(a_c, b_c, c_c\) |
| \(m3m\)   | \(n^{pri} = n^{sec} = n^{ter} = 4\) for \(a_c, b_c, c_c\) | \(n^{pri} = 4\) for all \(a_c, b_c, c_c\) |
The current centering type is easily identified from Table. III. For those, the correction matrices are listed in Table II.

For the Laue class 2/m, the basis vectors with the I, A, and B centering types are transformed to those with the C centering type. For the Laue class mmm, those with the A, and B centering types are transformed to those with the C centering type. For the rhombohedral system, a rhombohedral-centred hexagonal cell is obtained by \( \mathbf{M}' \) in either the obverse or reverse setting. This is transformed to the primitive rhombohedral cell by \( \mathbf{M}_{\text{obv}} \) if it is the obverse setting or by \( \mathbf{M}_{\text{rev}} \) if it is the reverse setting. Only one of \( \mathbf{M}' \mathbf{M}_{\text{obv}} \) or \( \mathbf{M}' \mathbf{M}_{\text{rev}} \) has to be an integer matrix, which is chosen as the transformation matrix to a primitive rhombohedral cell.

### Table II. Correction matrices \( \mathbf{M} \)

| Centering type | \( \det(\mathbf{M}') \) | Row vectors of \( \mathbf{M}' \) |
|---------------|----------------|------------------|
| \( P \)       | 1              | -                |
| \( A \)       | 2              | \( 3_i, \sum_j |M'_{ij}| = 1, |M'_{ij}| = 1 \) |
| \( B \)       | 2              | \( 3_i, \sum_j |M'_{ij}| = 1, |M'_{ij}| = 1 \) |
| \( C \)       | 2              | \( 3_i, \sum_j |M'_{ij}| = 1, |M'_{ij}| = 1 \) |
| \( I \) (body) | 2              | -                |
| \( R \) (rhombohedral) | 3         | -                |
| \( F \) (face)  | 4              | -                |

For those, the correction matrices are listed in Table II.

### C. Step (j): Identification of Hall symbol

The space group operations obtained in the second stage are compared with the datasets generated by decoding the Hall symbols. The Hall symbols are the explicit-origin space group notation proposed by Hall in Ref. [16]. The method to decode the Hall symbols is also found in *International Tables for Crystallography Volume B*. The 530 sets of matrix representations are pre-decoded and stored in the spglib source code.

To perform the comparison, the set of the space group operations has to be represented in the same coordinate system as that in the datasets. Using the transformation matrix \( \mathbf{M}' \mathbf{M} \) obtained in the step (i) as described in Eq. (C2), the space group operations are transformed into those corresponding to one specific conventional unit cell setting.

To match those in different unique axes, settings, or cell choices described by the Hall symbols, an additional change-of-basis matrix \( \mathbf{Q}' \) is employed. In the spglib code, the matrix \( \mathbf{Q}' \) can be selected such that after the transformation, it favors the conditions \( |a_c| \leq |b_c| \), \( |a_c| \leq |c_c| \), or \( |b_c| \leq |c_c| \), and being similar to the input unit cell in orientation. This preference is permissible when the choice of change-of-basis matrix is not constrained by the Hall symbol. Thus, the change-of-basis matrix \( \mathbf{M}' \mathbf{M} \) is updated by \( \mathbf{Q}' \) to \( \mathbf{M}' \mathbf{M} \mathbf{Q}' \).

For the space group type \( Pa3 \), two change-of-basis matrices, \( \mathbf{M}' \mathbf{M} \mathbf{Q}'_P \), \( \mathbf{M}' \mathbf{M} \mathbf{Q}'_A \), are examined to match with \( \mathbf{Q}'_Pa3 \) used in the spglib code.

\[
\mathbf{Q}'_{Pa3} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}
\]  

The matrix representations of the space group operations of \( \{ \mathbf{W}_r, \mathbf{w}_r \} \) in the second stage are given for the primitive cell. Each \( \{ \mathbf{W}_r, \mathbf{w}_r \} \) is transformed to that in the conventional unit cell, \( \mathbf{W}_c, \mathbf{w}_c \), by \( \mathbf{M}' \mathbf{Q}' \mathbf{M} \) as described in Eq. (C2). After this transformation, the rotation matrices of \( \mathbf{W}_c \) are directly comparable with those in the datasets.

To compare the translation parts to those in the datasets, their origins have to be aligned. An origin shift is determined using generators of the space group operations represented in the primitive cell setting. For the system having any centering, \( \{ \mathbf{W}_c, \mathbf{w}_c \} \) is transformed to \( \{ \mathbf{W}_X, \mathbf{w}_X \} \) by using the transformation matrix \( \mathbf{P}_X \) as presented in Table IV.

When the space group operation is represented by \( \{ \mathbf{W}_X, \mathbf{w}_X \} \) with respect to the origin \( \mathbf{O} \), and by \( \{ \mathbf{W}_X, \mathbf{w}_X^4 \} \) with respect to \( \mathbf{O}^4 \), and both matrix rep-
generators are required. For example, using these three
where
A set of solutions is obtained by applying the Smith nor-
where representations describe the same operation, they are inter-
so-
where
This is rewritten as

\[
\begin{pmatrix}
W_{X,1} - I \\
W_{X,2} - I \\
W_{X,3} - I
\end{pmatrix}
\begin{pmatrix}
p
\end{pmatrix}
= \begin{pmatrix}
W_{X,1} - w^d_{X,1} \\
W_{X,2} - w^d_{X,2} \\
W_{X,3} - w^d_{X,3}
\end{pmatrix}
= \Delta w_p \pmod{Z}.
\]
(19)
This is rewritten as
\[
Np = \Delta w_p \pmod{Z},
\]
(20)
where
\[
N = \begin{pmatrix}
W_{X,1} - I \\
W_{X,2} - I \\
W_{X,3} - I
\end{pmatrix}.
\]
(21)
A set of solutions is obtained by applying the Smith nor-
its Smith normal form \( S \) becomes
\[
S = \begin{pmatrix}
\lambda_1 & 0 & 0 \\
0 & \lambda_2 & 0 \\
0 & 0 & \lambda_3
\end{pmatrix}.
\]
(23)
The \( 3 \times 9 \) matrix \( T \) of the inverse diagonal elements of \( S \) is made as
\[
T = \begin{pmatrix}
\frac{1}{\lambda_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{\lambda_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{\lambda_3} & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.
\]
(24)
When \( \lambda_p = 0 \), the corresponding elements of \( T \) are set to 0. Since \( SV^{-1}p = U\Delta w_p \), one of the solutions is given by
\[
p_p = VTU\Delta w_p.
\]
(25)
A Python script was written to compute the matrices \( V \) for crystal systems and their centering types and
Consider \( \begin{pmatrix} W_{X,1}^d, w^d_{X,1} \end{pmatrix} \) as the reference provided from the dataset, and \( \begin{pmatrix} W_{X,1}^d, w^d_{X,1} \end{pmatrix} \) as derived using this symmetry-finding algorithm, where \( W_{X,1}^d = W_{X} \). To determine \( p_p \), at most three matrix representations of generators are required. For example, using these three generators, we can solve the equation below:
\[
\begin{pmatrix}
W_{X,1} - I \\
W_{X,2} - I \\
W_{X,3} - I
\end{pmatrix}
\begin{pmatrix}
p
\end{pmatrix}
= \begin{pmatrix}
w_{X,1,1} - w_{X,1,1}^d \\
w_{X,2,1} - w_{X,2,1}^d \\
w_{X,3,1} - w_{X,3,1}^d
\end{pmatrix} = \Delta w_p \pmod{Z}.
\]
(19)
This is rewritten as
\[
Np = \Delta w_p \pmod{Z},
\]
(20)
where
\[
N = \begin{pmatrix}
W_{X,1} - I \\
W_{X,2} - I \\
W_{X,3} - I
\end{pmatrix}.
\]
(21)
A set of solutions is obtained by applying the Smith normal form \( S \) given by
\[
S = UNV,
\]
(22)
where \( U \) and \( V \) are the unimodular matrices. In the case with the three generators, \( N \) is a \( 9 \times 3 \) matrix and

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} \\
0 & \frac{1}{2} & \frac{1}{2}
\end{pmatrix},
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix},
\begin{pmatrix}
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{pmatrix},
\begin{pmatrix}
0 & \frac{1}{2} & 0 \\
0 & \frac{1}{2} & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{pmatrix},
\begin{pmatrix}
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{pmatrix}.
\]
TABLE IV. Transformation matrices used in the Hall symbol matching. These matrices transform conventional unit cell settings to respective primitive cell settings. The subscripts \( X \) of the matrices \( P_X \) indicate the centering types: \( A, B, C \) for the base centering types, \( I \) and \( F \) for the body and face centering types, respectively, and \( R \) for the (obverse) rhombohedral centering type.

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & \frac{1}{2} & \frac{1}{2} \\
0 & \frac{1}{2} & \frac{1}{2}
\end{pmatrix},
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix},
\begin{pmatrix}
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{pmatrix},
\begin{pmatrix}
0 & \frac{1}{2} & 0 \\
0 & \frac{1}{2} & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{pmatrix},
\begin{pmatrix}
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{pmatrix}.
\]
E. Step (I): Thinning out space group operations

The space group operations in \( \{(W_p, w_p)\} \) are re-evaluated in a way similar to step (g) with the shortened tolerance value. As a result, certain space group operations in \( \{(W_p, w_p)\} \) may be excluded. Since \( w_p \) is given, this re-examination requires significantly less computational demand compared to the full execution of step (g). Following this, the procedure revisits step (i) with the refined set of space group operations.

VII. FINALIZATION

In the fourth stage, the results from the previous stages are organized to enhance their usefulness and intuitiveness. The matrix representations of the space group operations for the input unit cell are reconstructed using the transformation matrix, origin shift, and the Hall symbol dataset to minimize distortions in the translational parts. Additionally, Wyckoff positions are determined to minimize distortions in the translational parts. The tolerance introduced in the symmetry identification process may lead to some distortion in \( (a_2, b_2, c_2) \). As this involves a transformation of the coordinate system, \( p_c \) as given by Eq. (27) is also transformed according to:

\[ p_c' = W_d^{d*} (p_c - w_d^*). \]  (31)

Details of this coordinate transformation are provided in Appendix C. For conventional unit cells having any centering, \( (W_d^*, w_d^*) \) that gives the shortest \( p_c' \) is employed. With the use of \( p_c' \), the translation parts of \( \{(W_d, w_d)\} \) are redefined as:

\[ w_d^* = (W_d - I)p_c' + w_d. \]  (32)

Every \( (W_d, w_d^*) \) is transformed to that of the primitive cell \( (W_p^*, w_p^*) \) as the transformation detailed in Appendix C. The transformation matrix \( Q_{c^*} \) is defined by the equation:

\[ (a_p, b_p, c_p) = (a_c', b_c', c_c')Q_{c^*}. \]  (33)

Note that \( Q_{c^*}^{-1} \) is an integer matrix. To ensure that \( \{(W_p^*, w_p^*)\} \) represents the set of coset representatives, only one space group operation is included in \( \{(W_d, w_d^*)\} \) for each rotation part \( W_p^* \). Then, \( \{(W_d, w_d^*)\} \) is transformed to that of the input unit cell \( \{(W_{I^*}, w_{I^*})\} \). The transformation matrix \( Q_{p^*} \) is defined by the equation:

\[ (a_i, b_i, c_i) \approx (a_p, b_p, c_p)Q_{p^*}. \]  (34)

where the elements of the matrix \( Q_{p^*} \) are rounded to their nearest integer values, thereby constituting an integer matrix. Thus obtained \( W_{I^*} \) may not be an integer matrix if the order of the lattice point group of \( (a_i, b_i, c_i) \) is smaller than that of \( (a_p, b_p, c_p) \). Those space group operations with non-integer matrices of \( W_{I^*} \) are excluded from \( \{(W_{I^*}, w_{I^*})\} \).

When the input unit cell is not a primitive cell, \( \{(W_{I^*}, w_{I^*})\} \) is extended by a set of lattice point vectors in the input unit cell, \( \{t_j\} \),

\[ \{(W_{I^*}, w_{I^*})\} = \bigcup_j \{(I, t_j)\} \{(W_{I^*}, w_{I^*})\}. \]  (35)

The lattice point vectors of \( \{t_j\} \) are easily obtained from \( Q_{p^*}. \)
C. Step (o): Removal of distortion from point coordinates and determination of Wyckoff positions

Point coordinates in the primitive cell \( x_p \) are transformed to those in the conventional unit cell \( x_c \) by

\[
x_c = Q_{c \rightarrow p} x_p + p',
\]

(36)

Applying \( \{ (W^d, w^d) \} \) to \( \{ x_c \} \), symmetrically independent points and sets of symmetrically equivalent points are obtained.

Site symmetry operations of \( \{ (W_{x,i}, w_{x,i}) \} \) at \( x_c \) are the space group operations that leave coordinates of a point \( x_c \) unchanged, i.e.,

\[
(W_{x,i}, w_{x,i})x_c = x_c.
\]

(37)

\( \{ (W_{x,i}, w_{x,i}) \} \) is expected to form the site symmetry group \( S_x \) of the finite order \( |S_x| \). Using \( S_x \), the special position operator \( (W^{sp}, w^{sp}) \) is defined as

\[
(W^{sp}, w^{sp}) = \frac{1}{|S_x|} \sum_{i=1}^{S_x} (W_{x,i}, w_{x,i}).
\]

(38)

Point coordinates \( x_c \) can be slightly dislocated from the exact location. By Eq. (38), the exact location \( x_S \) of \( x_c \) is obtained by

\[
x_S = (W^{sp}, w^{sp})x_c.
\]

(39)

In the \texttt{spglib} implementation, \( \{ (W_{x,i}, w_{x,i}) \} \) is obtained from \( \{ (W^d, w^d) \} \). Since \( \{ (W^d, w^d) \} \) is the coset representatives of the lattice translation group of the conventional unit cell, i.e., it is a finite set not like a space group, Eq. (39) is examined using the condition (40) as

\[
\Delta x_i = (W_{x,i}, w_{x,i})x_c - x_c.
\]

(40)

Using Eq. (40), Eqs. (38) and (39) are rewritten as

\[
x_S = \frac{1}{|S_x|} \sum_{i=1}^{S_x} (\Delta x_i - |\Delta x_i|) + x_c.
\]

(41)

The number of the symmetrically equivalent points of the point \( x_S \) in the conventional unit cell is called multiplicity \( M_x \). Note that \( M_x \) is defined with respect to the conventional unit cell but not the primitive cell as following the convention of the \textit{International Tables for Crystallography Volume A}\textsuperscript{9}. These have to satisfy the following relationship:

\[
|S_x| M_x = |S/T| \det(P^{-1}_X),
\]

(42)

where \( |S/T| \) denotes the order of the factor group. This is same as the cardinality of the coset representatives obtained for the primitive cell. Obviously \( \det(P^{-1}_X) \) is equivalent to the number of the lattice points in the conventional unit cell. Finally, \( x_S \) of the symmetrically independent points are obtained, and these points are then expanded to their symmetrically equivalent points.

The Wyckoff letter of \( x_S \) is determined using \textit{Coordinates} in the Wyckoff position dataset. \textit{Coordinates} are listed in the \textit{International Tables for Crystallography Volume A}\textsuperscript{9}. The dataset in the \texttt{spglib} code was provided by Y. Seto\textsuperscript{23} for all the Hall symbols. The first entries of \textit{Coordinates} of Wyckoff positions for each Hall symbol is necessary to match \( x_S \) with a Wyckoff letter. All those first entries of \textit{Coordinates} are stored in the \texttt{spglib} source code in a matrix format. For example, the first entry of the Wyckoff letter \( f \) of \( P42_12 \) (No. 90) is \((x, x, \frac{1}{2})\), which is represented by

\[
\begin{pmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}.
\]

The \( 3 \times 3 \) and \( 3 \times 1 \) matrices are encoded and stored in the \texttt{spglib} source code. Matching with the dataset is performed by examining \((x, x, \frac{1}{2})x_S = x_S\) (mod \(Z\)). The multiplicity \( M_x \) is also stored in the dataset. With this \( M_x \), Eq. (42) is verified. If this fails, the procedure restarts from step (a) with the tolerance value shortened from that used last time at step (a).

VIII. SUMMARY

The \texttt{spglib} code is designed for the identification and symmetrization of crystal structures, that are provided as basis vectors, point coordinates, and atomic types, tolerating slight distortion. Utilizing established crystallography knowledge and algorithms, it examines crystal symmetry and validates symmetry operations searched numerically. During this process, an input tolerance value is adjusted to align matrix representations of symmetry operations with one of the space group types.

As the development of the \texttt{spglib} code has evolved, its source code has become less readable due to the series of incremental improvements made over time. This paper aims to clarify the implementation strategy of the current version of the \texttt{spglib} code, particularly for those keen on understanding the framework. Therefore, every detail is thoroughly described as it is implemented.

The accumulation of technical debt has complicated code maintenance, necessitating periodic major updates to support sustained scientific progress. This will be achieved by selecting a suitable programming language for each respective situation along with keeping the core of the code concise and efficient.

**ACKNOWLEDGMENTS**

We would like to thank Cristian Le and P. T. Jochym for their dedication to the continuous integration system of the \texttt{spglib} code. We also would like to thank Y. Seto for providing the space group type and Wyckoff position datasets. This work was supported by JSPS KAKENHI Grant Number JP21K04632.
Appendix A: Notations and conventions

In this section, the notations and conventions used in this paper are summarized. Basically, we follow and respect the notations and conventions of *International Tables for Crystallography Volume B* and Refs. 7 and 8.

1. Basis vectors \((a, b, c)\)

Basis vectors are represented by three column vectors:

\[
a = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}, \quad b = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}, \quad c = \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix},
\]

in the Cartesian coordinates.

2. Atomic point coordinates \(x\)

Coordinates of an atomic point \(x\) are represented by three values relative to basis vectors as follows:

\[
x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.
\]

A position vector \(x\) in the Cartesian coordinates are obtained by

\[
x = (a, b, c)x.
\]

3. Metric tensor \(G\) and point-group operation of lattice \(W\)

The metric tensor is defined by

\[
G = (a, b, c)^\top (a, b, c).
\]

A rotation matrix \(W\) is applied to the basis vectors such as

\[
(a, b, c) = (a, b, c)W.
\]

The metric tensor of \((a, b, c)\) is given by

\[
\tilde{G} = (a, b, c)^\top (a, b, c)
= W^\top (a, b, c)^\top (a, b, c)W
= W^\top GW.
\]

4. Space group operation \((W, w)\)

A crystal structure is transformed by a space group operation in which the coordinate system is at rest. Instead of rotating basis vectors as given in Eq. (A5), a point in direct space, \(x\), is transformed to a point \(\tilde{x}\) by a rotation by

\[
\tilde{x} = Wx.
\]

A space group operation has a rotation part \(W\) and a translation part \(w\). This is represented by the Seitz symbol \((W, w)\) that transforms \(x\) to \(\tilde{x}\) as

\[
\tilde{x} = (W, w)x = Wx + w.
\]

\(W\) and \(w\) are represented by a \(3 \times 3\) integer matrix and a \(3 \times 1\) column matrix, respectively. The point \(\tilde{x}\) has to be equal to one of the points \(x\) in the primitive cell up to lattice translation, for \((W, w)\) to be a space group operation.

5. Transformation of coordinate system \((P, p)\)

The coordinate system of a crystal structure at rest is transformed by a pair of a \(3 \times 3\) matrix \(P\) and a \(3 \times 1\) column matrix \(p\), which is denoted by \((P, p)\). The transformation matrix \(P\) changes a choice of basis vectors as follows

\[
(a, b, c) = (a_s, b_s, c_s)P,
\]

where \((a, b, c)\) and \((a_s, b_s, c_s)\) are, e.g., the basis vectors of a primitive cell and those of the conventional unit cell, respectively. The transformation matrix doesn’t rotate a crystal in the Cartesian coordinates, but just changes the choices of the basis vectors. The origin shift \(p\) gives the vector from the origin of an original coordinate system \(O_s\) to that of any other coordinate system \(O\), which is written as

\[
p = O - O_s.
\]

The origin shift does not move the crystal in the Cartesian coordinates, but just change the origin to measure the point coordinates.

The point coordinates in the original coordinate system \(x_s\) and those in the transformed coordinate system \(x\) are related by

\[
x_s = (P, p)x = Px + p,
\]

where \(p\) is given with respect to the original basis vectors. Equivalently,

\[
x = P^{-1}x_s - P^{-1}p.
\]

An illustration is presented in Fig. 1. In this case, the following \(P\) is applied:

\[
P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}.
\]
We expect respectively. A change-of-basis matrix relates them as and non-primitive cells as (W).

FIG. 2. Transformation of coordinate system.

Table V. Look-up table to identify the types of the rotation parts of the space group operations from their matrix representations.

| Type of $W$ | $\text{tr}(W)$ | $\det(W)$ |
|------------|----------------|------------|
| -6 | -2 | -1 |
| -4 | -1 | 0 |
| -3 | 1 | 1 |
| -2 | 2 | 3 |
| -1 | 4 | 5 |
| 1 | 6 | 6 |
| 2 | 3 | 4 |
| 3 | 5 | 6 |
| 4 | 6 | 0 |
| 5 | 0 | 2 |
| 6 | 0 | 0 |

Appendix B: Crystallographic point group

The crystallographic point group type (crystal class) is uniquely determined from matrix representations of the rotation parts of the coset representatives. The rotation type is identified using Table V with the determinant and trace of $W$. The crystal class is found comparing the list of numbers of the rotation types with Table VI. Crystal system and Laue class are uniquely assigned from the crystal class.

Appendix C: Transformation of matrix representation of space group operation

When basis vectors ($a_i$, $b_i$, $c_i$) are transformed to another basis vectors ($a_i'$, $b_i'$, $c_i'$) by a change-of-basis matrix Q as

$$(a_i, b_i, c_i) = (a_i, b_i, c_i)Q,$$  \hspace{1cm} (C1)

the matrix representation of the space group operation ($W_i$, $w_i$) is transformed to

$$(W_i, w_i) = (Q^{-1}W_iQ, Q^{-1}w_i).$$  \hspace{1cm} (C2)

When transforming space group operations of a primitive cell to those of a non-primitive cell, careful consideration is required. Denote the basis vectors of the primitive and non-primitive cells as ($a_p$, $b_p$, $c_p$) and ($a_{np}$, $b_{np}$, $c_{np}$), respectively. A change-of-basis matrix relates them as

$$(a_{np}, b_{np}, c_{np}) = (a_p, b_p, c_p)Q.$$  \hspace{1cm} (C3)

We expect

$$W_{np} = Q^{-1}W_pQ.$$  \hspace{1cm} (C4)

Appendix D: spglib convention of symmetrization of basis vectors

The spglib code uses specific conventions to idealize crystal structures, which are detailed for each crystal system.

1. Triclinic

(1) Niggli reduced cell is used for choosing $a$, $b$, and $c$. (2) $a$ is aligned with $+x$ direction of Cartesian co-
ordinates. (3) \( \mathbf{b} \) is positioned in \( x-y \) plane of Cartesian coordinates, ensuring that \( \mathbf{a} \times \mathbf{b} \) aligns with +z direction of Cartesian coordinates.

2. Monoclinic

(1) \( b \) axis is taken as the unique axis. (2) \( \alpha = 90^\circ \) and \( \gamma = 90^\circ \). (3) \( 90^\circ < \beta < 120^\circ \). (4) \( \mathbf{a} \) is aligned with +x direction of Cartesian coordinates. (5) \( \mathbf{b} \) is aligned with +y direction of Cartesian coordinates. (6) \( \mathbf{c} \) is positioned in \( x-z \) plane of Cartesian coordinates.

3. Orthorhombic

(1) \( \alpha = \beta = \gamma = 90^\circ \). (2) \( a = b = c \). (3) \( \mathbf{a} \) is aligned with +x direction of Cartesian coordinates. (4) \( \mathbf{b} \) is aligned with +y direction of Cartesian coordinates. (5) \( \mathbf{c} \) is aligned with +z direction of Cartesian coordinates.

4. Tetragonal

(1) \( \alpha = \beta = \gamma = 90^\circ \). (2) \( a = b \). (3) \( \mathbf{a} \) is aligned with +x direction of Cartesian coordinates. (4) \( \mathbf{b} \) is aligned with +y direction of Cartesian coordinates. (5) \( \mathbf{c} \) is aligned with +z direction of Cartesian coordinates.

5. Rhombohedral

(1) \( \alpha = \beta = \gamma \). (2) \( a = b = c \). (3) When projected onto \( x-y \) plane in Cartesian coordinates, \( \mathbf{a}, \mathbf{b}, \) and \( \mathbf{c} \) become \( \mathbf{a}_{xy}, \mathbf{b}_{xy}, \) and \( \mathbf{c}_{xy} \), respectively, with their angles denoted as \( \alpha_{xy}, \beta_{xy}, \gamma_{xy} \). (4) The projections of \( \mathbf{a}, \mathbf{b}, \) and \( \mathbf{c} \) along \( z \)-axis in Cartesian coordinates are \( \mathbf{a}_z, \mathbf{b}_z, \) and \( \mathbf{c}_z \), respectively. (5) \( \mathbf{a}_{xy} \) is oriented along a ray rotated \( 30^\circ \) counter-clockwise from the +z direction in Cartesian coordinates, with \( \mathbf{b}_{xy} \) and \( \mathbf{c}_{xy} \) positioned at angles of \( 120^\circ \) and \( 240^\circ \) counter-clockwise from \( \mathbf{a}_{xy} \), respectively. (6) \( \mathbf{a}_{xy} = \mathbf{b}_{xy} = \mathbf{c}_{xy} = \mathbf{a}_{2y} \). (7) \( \mathbf{a}_{xy} = \mathbf{b}_{xy} = \mathbf{c}_{xy} = \mathbf{a}_{2y} \).

6. Hexagonal

(1) \( \alpha = \beta = 90^\circ \). (2) \( \gamma = 120^\circ \). (3) \( a = b \). (4) \( \mathbf{a} \) is aligned with +x direction of Cartesian coordinates. (5) \( \mathbf{b} \) is positioned in \( x-y \) plane of Cartesian coordinates. (6) \( \mathbf{c} \) is aligned with +z direction of Cartesian coordinates.

7. Cubic

(1) \( \alpha = \beta = \gamma = 90^\circ \). (2) \( a = b = c \). (3) \( \mathbf{a} \) is aligned with +x direction of Cartesian coordinates. (4) \( \mathbf{b} \) is aligned with +y direction of Cartesian coordinates. (5) \( \mathbf{c} \) is aligned with +z direction of Cartesian coordinates.

Appendix E: Transformation of origin shift by space group operation

When considering a space group operation \((W, w)\) as a transformation of the coordinate system, as detailed in Appendix A5, the new point coordinates \(x_{\text{new}}\) are related to the original point coordinates \(x_{\text{orig}}\) through Eq. (A11) by

\[
x_{\text{orig}} = (W, w)x_{\text{new}}.
\]

By definition (A9),

\[
(a_{\text{new}}, b_{\text{new}}, c_{\text{new}}) = (a_{\text{orig}}, b_{\text{orig}}, c_{\text{orig}})W.
\]

Due to the space group operation \((W, w)\), the sets of point coordinates in the original and new coordinate systems are equal up to lattice translation,

\[
\{x_{\text{new}}\} = \{x_{\text{orig}}\} \pmod{\mathbb{Z}}.
\]

Consequently, the same set of matrix representations for the elements of \{\((W, w)\)\} is applicable to \(\{x_{\text{new}}\}\).

An origin shift from the original coordinate system is represented in the new coordinate system as follows. Considering an origin shift \((I, p)\) with respect to the original coordinate system, the point coordinates \(x_p\) after this origin shift are related to \(x_{\text{orig}}\) by

\[
x_{\text{orig}} = (I, p)x_p.
\]

From Eqs. (E4) and (E5), \(x_{\text{new}}\) and \(x_p\) are related as

\[
x_{\text{new}} = (W, w)^{-1}(I, p)x_p = (W^{-1}, -W^{-1}w)(I, p)x_p.
\]

Viewed from the new coordinate system, the origin shift \(p_{\text{new}}\) is determined by setting \(x_p = 0\),

\[
p_{\text{new}} = (W^{-1}, -W^{-1}w)p = W^{-1}(p - w).
\]
4 “ISOTROPY Software Suite,” http://stokes.byu.edu/iso/isotropy.php.
5 R. W. Grosse-Kunstleve, N. K. Sauter, N. W. Moriarty, and P. D. Adams, J. Appl. Crystallogr. 35, 126 (2002).
6 R. W. Grosse-Kunstleve, N. K. Sauter, and P. D. Adams, Acta Crystallogr. A 60, 1 (2004).
7 R. W. Grosse-Kunstleve and P. D. Adams, Acta Crystallogr. A 58, 60 (2002).
8 R. W. Grosse-Kunstleve, Acta Crystallogr. A 55, 383 (1999).
9 “Computational Crystallography Toolbox,” https://cctbx.github.io/.
10 D. Hicks, C. Oses, E. Gossett, G. Gomez, R. H. Taylor, C. Toher, M. J. Mehl, O. Levy, and S. Curtarolo, Acta Crystallogr. A 74, 184 (2018).
11 “Spglib,” https://github.com/spglib/spglib version 2.3.1.
12 K. Shinohara, A. Togo, and I. Tanaka, Acta Crystallographica Section A 79, 390 (2023).
13 A. Hannemann, R. Hundt, J. C. Schönh, and M. Jansen, J. Appl. Crystallogr. 31, 922 (1998).
14 M. Lamparski, Large-Scale Atomistic Computations of the Phonons in Twisted Bilayer Graphene, Phd thesis, Department of Physics, Applied Physics, and Astronomy Rensselaer Polytechnic Institute, Troy, New York (2020), available at https://hdl.handle.net/20.500.13015/2055.
15 B. Delaunay, Zeitschrift für Kristallographie - Crystalline Material, S. R. Hall, Acta Crystallogr. A 37, 517 (1981).
16 U. Shmueli, International Tables for Crystallography, Vol. B (Wiley, 2008).
17 P. Niggli, Krystallographische und strukturtheoretische Grundbegriffe. Handbuch der Experimentalphysik, Vol. 7 (Leipzig: Akademische Verlagsgesellschaft, 1928) pp. 108–176.
18 B. Gruber, Acta Crystallogr. A 29, 433 (1973).
19 I. Křivý and B. Gruber, Acta Crystallogr. A 32, 297 (1976).
20 Niggli cell, http://atztogo.github.io/niggli/.
21 SageMath, the Sage Mathematics Software System, http://www.sagemath.org.
22 A. Togo, L. Chaput, T. Tadano, and I. Tanaka, J. Phys. Condens. Matter 35, 353001 (2023).
23 “Seto’s Home Page,” https://yseto.net/?page_id=29&lang=en.