Taylor & Francis
Taylor & Francis Group

Science and Technology of Advanced Materials: Methods

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tstm20

# Spglib: a software library for crystal symmetry search

Atsushi Togo, Kohei Shinohara & Isao Tanaka

**To cite this article:** Atsushi Togo, Kohei Shinohara & Isao Tanaka (2024) Spglib: a software library for crystal symmetry search, Science and Technology of Advanced Materials: Methods, 4:1, 2384822, DOI: 10.1080/27660400.2024.2384822

**To link to this article:** https://doi.org/10.1080/27660400.2024.2384822



Published online: 16 Oct 2024.

SCIENCE AND TECHNOLOGY OF ADVANCED MATERIALS: METHODS
2024, VOL. 4, NO. 1, 2384822
https://doi.org/10.1080/27660400.2024.2384822



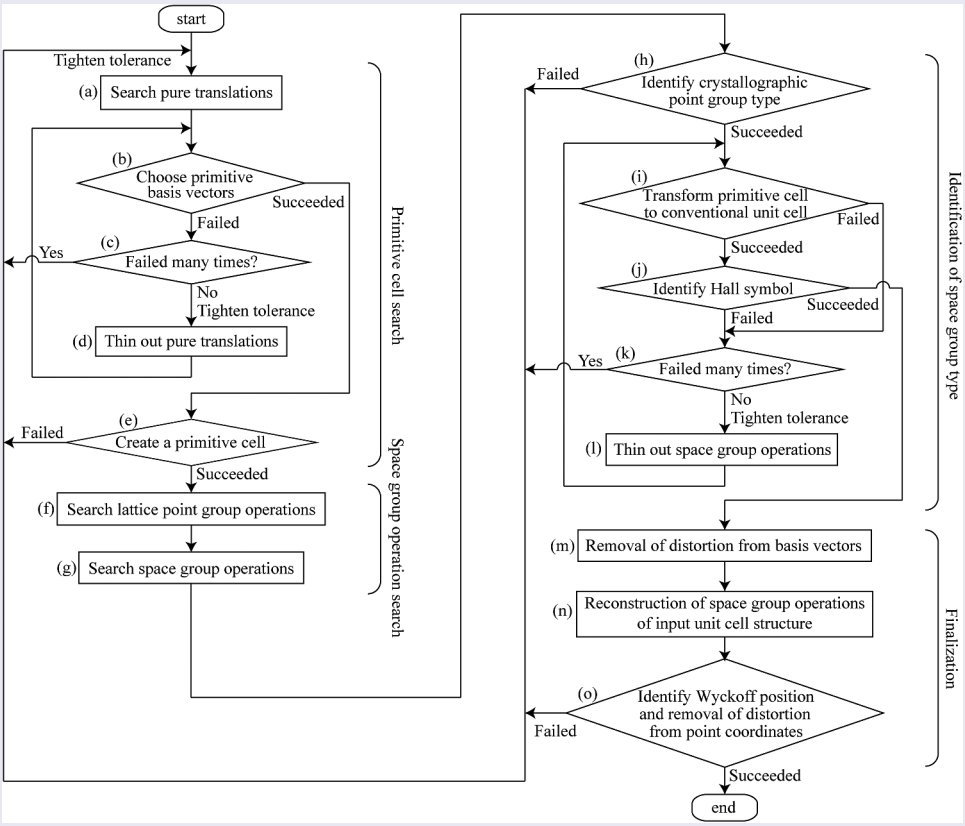
# Spglib: a software library for crystal symmetry search

Atsushi Togo[a,b], Kohei Shinohara[c] and Isao Tanaka[b,d,e]

[a]Center for Basic Research on Materials, National Institute for Materials Science, Tsukuba, Japan; [b]Center for Elements Strategy Initiative for Structural Materials, Kyoto University, Kyoto, Japan; [c]Preferred Networks, Inc, Tokyo, Japan; [d]Department of Materials Science and Engineering, Kyoto University, Kyoto, Japan; [e]Nanostructures Research Laboratory, Japan Fine Ceramics Center, Nagoya, Japan

**ABSTRACT**

A computer algorithm to search the 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 crystal structures.




**ARTICLE HISTORY**
Received 15 March 2024
Revised 18 July 2024
Accepted 22 July 2024

**KEYWORDS**
Space group; crystal symmetry; space group type; crystallographic point group

**IMPACT STATEMENT**

This paper presents a detailed description of the computer algorithm utilized in the spglib code, which is a computer software library for determining crystal symmetry and space group type classifications.

## 1. Introduction

Crystal symmetry is essential information for understanding various crystal properties. It is also useful for compressing information about the 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 types [1]. Here, the crystal structure is a mathematical model, where equivalent unit cells are arranged on a periodic lattice, and each unit cell is defined by basis vectors and point coordinates of atoms labeled by atomic species.

**CONTACT** Atsushi Togo togo.atsushi@nims.go.jp Center for Basic Research on Materials, National Institute for Materials Science, Tsukuba 305-0047, Japan

Assuming the periodicity, a crystal structure is represented by the numerical values of the basis vectors and point coordinates in a unit cell. The symmetry operations are also numerically represented with respect to the unit cell. Due to the periodicity of unit cells, another linear combination of the basis vectors can also be a valid unit cell, resulting in different matrix representations of the symmetry operations. It is inconvenient to represent the same crystal symmetry in various representations. To obtain a unique matrix representation of symmetry operations, conventional choices and settings for unit cell representations are employed in crystallography. These matrix representations of the space-group types are tabulated in the *International Tables for Crystallography Volume A* (ITA) [1].

Given a crystal structure, its symmetry operations are searched computationally, and then the space-group type is identified. If the crystal structure is provided in a standard setting, the identification is straightforward. However, it is often provided with a different choice of basis vectors and an arbitrary origin shift, i.e. a non-standard setting. Furthermore, when the basis vectors and point coordinates are slightly distorted with respect to the expected crystal symmetry, the symmetry search becomes a challenging task. The spglib code has been developed to solve these difficulties.

There already exist crystal symmetry-finding codes. The FINDSYM code in the ISOTROPY Software Suite [2–4] is well known; however, its source code is not publicly available. The cctbx [5–9] code is an open-source computational crystallography toolbox under a BSD-type variant license. The sginfo [5–8] code, which is currently distributed under an open-source software license, has been superseded by the space group toolbox (sgtbx) in the cctbx code. The AFLOW-SYM [10] 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, which is a permissive open-source software license [11].

Because the spglib code operates as a software library and is not directly visible to users, its widespread use across various software packages may not be apparent. However, the spglib code is widely utilized by numerous software packages for atomistic simulation, visualization, and utilities, including the CP2K [12], Octopus [13], phonopy [14], pymatgen [15], ASE [16], and Avogadro codes [17], as well as online materials databases such as the Materials Project [18] and NOMAD [19]. Since many software packages already rely on it, the spglib code is expected to be well maintained and have a longer lifespan. Therefore, this paper aims to engage developers of the spglib code by presenting a detailed algorithm implemented in the spglib code.

Using the spglib code, space group operations are searched from a crystal structure in which small deviations from ideal atomic positions are tolerated. As a result, we obtain the coset representatives $(\boldsymbol{W}_i, \boldsymbol{w}_i)$ of the space group $\mathbb{S}$ with respect to the translation group $\mathbb{T}$, defined by the coset decomposition:

$$\mathbb{S} = \sum_i (\boldsymbol{W}_i, \boldsymbol{w}_i)\mathbb{T}, \tag{1}$$

where $(\boldsymbol{W}_i, \boldsymbol{w}_i)$ is explained in Appendix A. Some other useful information about crystal symmetry is simultaneously derived while running the algorithm. The information obtained through the spglib code is applicable to research in materials science or condensed matter physics. Reading the references [20–22] can be useful for gaining a deeper understanding of these subjects.

In the algorithm outlined in this paper, an iterative approach is employed to determine space group operations for the initial input crystal structure at a given distance tolerance. The flowchart of the algorithm is presented in Figure A1. 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. Nested iteration loops are present in the algorithm, some of which are explicitly depicted in Figure A1. Additional loops are contained within the blocks. The tolerance value decreases whenever an operation fails in any of the loops. One exception is step (e), where attempts are made to both increase and decrease the tolerance value.

This paper focuses on the algorithms designed for searching space group operations and identifying space group types. The algorithm related to the magnetic space group, which builds upon the space group algorithm, is discussed in another publication [23]. A tolerance parameter, 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.

## 2. Tolerance parameter in crystal symmetry operation search

In the spglib code, numerical searches for symmetry operations of a given input unit cell utilize a small Euclidean distance, $\varepsilon$, as a tolerance parameter. The value of the tolerance parameter is adjusted during the algorithmic process to identify a possible space group.

This section explains how it is used to examine the symmetrical equivalence of two atomic points. The method of its adjustment is detailed in subsequent sections. Symbols used in this section are summarized in Appendices A.1–A.4.

### 2.1. Lattice translation to a point near origin

Each element of point coordinates, $\boldsymbol{x} = (x_1, x_2, x_3)^\top$, is confined within the interval $[-0.5, 0.5)$ by the operation:

$$x_i \leftarrow x_i - \lfloor x_i \rceil, \ i = 1, 2, 3, \tag{2}$$

where $\lfloor x_i \rceil$ denotes rounding $x_i$ to the nearest integer. For the coordinates, this operation is concisely represented as

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \lfloor \boldsymbol{x} \rceil, \tag{3}$$

and similarly for vectors as,

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \lfloor \boldsymbol{w} \rceil. \tag{4}$$

### 2.2. Determination of equivalent points under lattice periodicity

The tolerance parameter, $\varepsilon$, is used to determine whether two points $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}'$ occupy the same atomic site. This is examined using the position difference, $\Delta\boldsymbol{x} = \tilde{\boldsymbol{x}} - \boldsymbol{x}'$, by evaluating

$$|(\mathbf{a}, \mathbf{b}, \mathbf{c})\Delta x| < \varepsilon, \tag{5}$$

where $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ represent the basis vectors of the unit cell. To accommodate lattice periodicity, condition (5) is reformulated using operation (4) as

$$|(\mathbf{a}, \mathbf{b}, \mathbf{c})(\Delta\boldsymbol{x} - \lfloor \Delta\boldsymbol{x} \rceil)| < \varepsilon. \tag{6}$$

This expression is frequently used in the implementation.

### 2.3. Examination of a symmetry operation

Given a space group operation $(\boldsymbol{W}, \boldsymbol{w})$, which may or may not be a valid symmetry operation, an atomic point $\boldsymbol{x}$ is transformed to $\tilde{\boldsymbol{x}} = (\boldsymbol{W}, \boldsymbol{w})\boldsymbol{x}$ by Eq. (A8). If $(\boldsymbol{W}, \boldsymbol{w})$ represents a valid space group operation, $\tilde{\boldsymbol{x}}$ must be located at one of atomic sites, $\boldsymbol{x}'$, having the same atomic type. In the algorithm, this is examined for all atoms in either the input unit cell or the primitive cell using condition (6). When this is satisfied, the given space group operation $(\boldsymbol{W}, \boldsymbol{w})$ is accepted.

## 3. Machine precision issue

It is assumed that the spglib code is used on both 32- and 64-bit computer systems. As reported in Ref [6], length comparisons must be carefully implemented in the code due to finite machine precision. For example, an inequality $x < y$ might be implemented as $x < y - \varepsilon'$ with $\varepsilon'$ being a small positive value [6]. Most inequalities in the implementation follow the style of Equation (6) and no special adjustments are applied in these cases. Conversely, the operation to take modulo by $\mathbb{Z}$ employs the above inequality, using $\varepsilon'$. Through this operation, a value $x_i$ is adjusted to fall within the interval $[-\varepsilon', 1 - \varepsilon')$. In the current version of the spglib code, $\varepsilon'$ is set to $10^{-10}$.

## 4. Primitive cell search

In the first stage of the space group symmetry search, a primitive cell is identified from lattice points in the input unit cell. These lattice points are obtained through a search for pure translation operations.

### 4.1. 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 $\boldsymbol{w}_I$ of pure translation operations of $\{(\boldsymbol{I}, \boldsymbol{w}_I)\}$ within the input unit cell, where $\boldsymbol{I}$ is the identity matrix. The pure translation operations of $\{(\boldsymbol{I}, \boldsymbol{w}_I)\}$ are searched as follows: Candidates of the translation parts $\boldsymbol{w}_I^{\mathrm{c}}$ are selected from vectors that extend from a fixed atomic site $\boldsymbol{x}$ to all atomic sites $\boldsymbol{x}'$ of the same atomic type $A$, i.e. $\boldsymbol{w}_I^{\mathrm{c}} = \boldsymbol{x}_A' - \boldsymbol{x}_A$. To minimize the computational demand, the fixed atomic site is chosen among atoms having an atomic type that comprises the smallest number of atoms [24]. Then, $(\boldsymbol{I}, \boldsymbol{w}_I^{\mathrm{c}})$ is examined as described in Sec. 2.3.

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

In typical use cases, this step is the most computationally demanding part of 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 worst-case scenarios [25].

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

Candidates for the three basis vectors of a primitive cell $(\mathbf{a}_{\mathrm{p}}^{\mathrm{c}}, \mathbf{b}_{\mathrm{p}}^{\mathrm{c}}, \mathbf{c}_{\mathrm{p}}^{\mathrm{c}})$ are chosen from the set of vectors $\mathbf{T}_{\mathrm{i}} \cup \mathbf{T}_{\mathrm{p}}$, where $\mathbf{T}_{\mathrm{i}} = \{\mathbf{a}_{\mathrm{i}}, \mathbf{b}_{\mathrm{i}}, \mathbf{c}_{\mathrm{i}}\}$ is the set composed of the basis vectors of the input unit cell, and $\mathbf{T}_{\mathrm{p}} = \{\boldsymbol{w}_I\}$ found at step (a). Three basis vectors are chosen to create a right-handed coordinate system. The volume of the primitive cell, $V_{\mathrm{p}}$, is expected to be approximately the volume of the

input unit cell, $V_{\rm i}$, divided by $|\mathbf{T}_{\rm p}|$. Therefore, the basis vectors of the primitive cell are searched under the following condition:

$$|\mathbf{T}_{\rm p}| = \lfloor V_{\rm i}/V_{\rm p} \rceil. \tag{7}$$

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

For distorted input unit cells, the condition (7) may not always be satisfied. For example, the number of pure translations found can be either more 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).

### 4.4. Step (d): Thinning out pure translations

Some of the pure translations that do not satisfy condition (7) 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.

### 4.5. Step (e): Creating a primitive cell

The primitive cell basis vectors $(\mathbf{a}_{\rm p}^{\rm c}, \mathbf{b}_{\rm p}^{\rm c}, \mathbf{c}_{\rm p}^{\rm c})$ found at step (b) are transformed into a different set of primitive cell basis vectors $(\mathbf{a}_{\rm p}, \mathbf{b}_{\rm p}, \mathbf{c}_{\rm p})$ by the Delaunay reduction [1,26]. This transformation is written as

$$(\mathbf{a}_{\rm p}, \mathbf{b}_{\rm p}, \mathbf{c}_{\rm p}) = (\mathbf{a}_{\rm p}^{\rm c}, \mathbf{b}_{\rm p}^{\rm c}, \mathbf{c}_{\rm p}^{\rm c})\mathbf{Q}_{\rm D}. \tag{8}$$

$\mathbf{Q}_{\rm D}$ is an integer matrix and is chosen such that $\det(\mathbf{Q}_{\rm D}) = 1$.

Similarly, the transformation of the basis vectors of the primitive cell basis vectors to those of the input unit cell $(\mathbf{a}_{\rm i}, \mathbf{b}_{\rm i}, \mathbf{c}_{\rm i})$ in a right handed coordinate system is written by the change-of-basis matrix $\mathbf{Q}_{\rm p\to i}$ as

$$(\mathbf{a}_{\rm i}, \mathbf{b}_{\rm i}, \mathbf{c}_{\rm i}) \approx (\mathbf{a}_{\rm p}, \mathbf{b}_{\rm p}, \mathbf{c}_{\rm p})\mathbf{Q}_{\rm p\to i} \tag{9}$$

where $\mathbf{Q}_{\rm p\to i}$ is chosen as an integer matrix with $\det(\mathbf{Q}_{\rm p\to i}) \geq 1$ although Equation (9) is an approximation if $\mathbf{T}_{\rm p} = \{\mathbf{w}\}$ found at step (a) is distorted with respect to the lattice of the primitive cell.

Point coordinates in the input unit cell, $\boldsymbol{x}_{\rm i}$, are transformed to their corresponding coordinates in the primitive cell, $\boldsymbol{x}_{\rm p*}$, by

$$\boldsymbol{x}_{\rm p*} = \mathbf{Q}_{\rm p\to i}\boldsymbol{x}_{\rm i} \tag{10}$$

where $\boldsymbol{x}_{\rm p*}$ is brought into the interval $[-0.5, 0.5)$ by operation (2). When $\det(\mathbf{Q}_{\rm p\to i}) > 1$, multiple $\boldsymbol{x}_{\rm i}$ should be mapped to a point $\boldsymbol{x}_{\rm p*}$ with respect to $(\mathbf{a}_{\rm p}, \mathbf{b}_{\rm p}, \mathbf{c}_{\rm p})$ using condition (6). The tolerance value $\varepsilon$ in condition (6) is adjusted until the multiplicity becomes $\det(\mathbf{Q}_{\rm p\to i})$ for all translationally independent $\boldsymbol{x}_{\rm p*}$. If this fails, the procedure restarts from step (a) with the tolerance value tightened from that previously used in step (a).

Upon successful verification, the point coordinates of translationally equivalent atoms are averaged to produce $\boldsymbol{x}_{\rm p}$ along with boundary treatment for $\boldsymbol{x}_{\rm p*}$ located close to the boundary of $[-0.5, 0.5)$. Finally, a primitive cell is created from $(\mathbf{a}_{\rm p}, \mathbf{b}_{\rm p}, \mathbf{c}_{\rm p})$ and $\{\boldsymbol{x}_{\rm p}\}$.

## 5. Space group operation search

The purpose of the second stage is to search a set of space group operations $\{(\boldsymbol{W}_{\rm p}, \boldsymbol{w}_{\rm p})\}$ of the primitive cell obtained in the first stage. Candidates of the rotation parts $\{\boldsymbol{W}^{\rm c}\}$ are given by an exhaustive search of lattice point group operations. Using obtained $\{\boldsymbol{W}^{\rm c}\}$ and the point coordinates $\{\boldsymbol{x}_{\rm p}\}$, a set of space group operations $\{(\boldsymbol{W}_{\rm p}, \boldsymbol{w}_{\rm p})\}$ is searched.

### 5.1. Step (f): Searching lattice point group operations

A set of possible candidates for lattice point group operations $\{\boldsymbol{W}_{\rm L}^{\rm c}\}$ is exhaustively generated by filling the matrix elements with −1, 0, or 1 under the constraint of $|\det(\boldsymbol{W}_{\rm L}^{\rm c})| = 1$. As described in Appendice A.3, a set of lattice point group operations of the primitive cell, $\{\boldsymbol{W}_{\rm L}\}$, is searched within $\{\boldsymbol{W}_{\rm L}^{\rm c}\}$ using the metric tensor $\boldsymbol{G}$ that is rotated by $\boldsymbol{W}_{\rm L}$ as $\tilde{\boldsymbol{G}} = (\boldsymbol{W}_{\rm L})^{\top}\boldsymbol{G}\boldsymbol{W}_{\rm L}$.

Comparison of $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$ is performed as follows: The diagonal elements of the matrix $\boldsymbol{G}$ 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 $\varepsilon$ based on Euclidean distance. For angle comparison, an angle tolerance parameter can be employed. However, this addition of an extra tolerance parameter may complicate usage. Therefore, in the current implementation, if an angle tolerance is not explicitly specified, the distance tolerance $\varepsilon$ is approximated for angle comparisons. This approach is applied, for instance, in the evaluation of $G_{12}$, as shown in the following equation:

$$\sin|\Delta\theta|\sqrt{\frac{(|\tilde{\mathbf{a}}| + |\mathbf{a}|)(|\tilde{\mathbf{b}}| + |\mathbf{b}|)}{4}} < \varepsilon, \tag{11}$$

where $\Delta\theta$ is the angle difference between the two pairs of vectors $\mathbf{a}$-$\mathbf{b}$ and $\tilde{\mathbf{a}}$-$\tilde{\mathbf{b}}$,

$$\Delta\theta = \arccos\left(\frac{\tilde{\mathbf{a}}\cdot\tilde{\mathbf{b}}}{|\tilde{\mathbf{a}}||\tilde{\mathbf{b}}|}\right) - \arccos\left(\frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right). \tag{12}$$

The angle difference $\Delta\theta$ is compared with the tolerance value using the averaged lengths of the basis vectors. The left-hand side of Equation (11) is given

by the matrix elements of $\boldsymbol{G}$ and $\tilde{\boldsymbol{G}}$ such as $\mathbf{a}\cdot\mathbf{b} = G_{12}$, $|\mathbf{a}| = \sqrt{G_{11}}$, and $|\mathbf{b}| = \sqrt{G_{22}}$.

In a similar way to that applied in step (a), the corresponding translation part $\boldsymbol{w}_\mathrm{p}$ is searched using $\boldsymbol{W}_\mathrm{L}$ instead of $\boldsymbol{I}$ in step (a).

### 5.2. Step (g): Searching space group operations

A set of space group operations $\{(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})\}$ is searched in the following way. The rotation matrices $\{\boldsymbol{W}_\mathrm{L}\}$ found in step (f) are used as candidates of the rotation parts of $\{(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})\}$. From Equation (A8), a space group operation $(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})$ satisfies $\tilde{\boldsymbol{x}} = \boldsymbol{W}_\mathrm{p}\boldsymbol{x} + \boldsymbol{w}_\mathrm{p}$. Therefore, candidates of translation parts for $\boldsymbol{W}_\mathrm{L}$ are given by $\boldsymbol{w}_\mathrm{p}^\mathrm{c} = \tilde{\boldsymbol{x}} - \boldsymbol{W}_\mathrm{L}\boldsymbol{x}$ over possible combinations of $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$.

To limit its search space, for a fixed $\boldsymbol{x}$, $\tilde{\boldsymbol{x}}$ are selected from all atoms with the same atomic type as $\boldsymbol{x}$ in the primitive cell. Then, $(\boldsymbol{W}_\mathrm{L}, \boldsymbol{w}_\mathrm{p}^\mathrm{c})$ is examined by applying condition (6) with $\Delta\boldsymbol{x} = \left(\boldsymbol{W}_\mathrm{L}, \boldsymbol{w}_\mathrm{p}^\mathrm{c}\right)\boldsymbol{x} - \tilde{\boldsymbol{x}}$ for all $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$. If none of $\boldsymbol{w}_\mathrm{p}^\mathrm{c}$ is found, this $\boldsymbol{W}_\mathrm{L}$ is discarded, otherwise $(\boldsymbol{W}_\mathrm{L}, \boldsymbol{w}_\mathrm{p}^\mathrm{c})$ is adopted as $(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})$. This procedure is repeated over all elements of $\{\boldsymbol{W}_\mathrm{L}\}$. Finally, a set of space group operations $\{(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})\}$ is obtained. In the next stage, it is verified that $\{(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})\}$ constitutes the coset representatives of the factor group $\mathbb{S}/\mathbb{T}$.

## 6. Identification of space group type

In this third stage, a space group type is identified by comparing the set of the space group operations $\{(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})\}$ obtained in the last stage with those sets coded in the Hall symbols [27,28]. 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 the matrix representations decoded from the Hall symbols, and the origin shift is simultaneously determined during the matching process [8]. The algorithm presented in this section follows almost exactly that reported by Grosse-Kunstleve and Adams in Ref [8], and it is described as implemented in the spglib code.

### 6.1. Step (h): Identify crystallographic point group

The crystallographic point group is given by collecting the rotation parts of the space group operations, i.e. $\mathbb{P} = \{\boldsymbol{W}_\mathrm{p} | (\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p}) \in \{(\boldsymbol{W}_\mathrm{p}, \boldsymbol{w}_\mathrm{p})\}\}$. The crystallographic point group type is identified from the traces and determinants of the matrices of $\{\boldsymbol{W}_\mathrm{p}\}$ by using the look-up Tables B1 and B2 presented in Appendix B. If the identification of the crystallographic point group type fails, the procedure restarts from step (a) with the tolerance value tightened from that previously used in step (a).

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

The 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 B2.

The transformation of the basis vectors of the primitive cell $(\mathbf{a}_\mathrm{p}, \mathbf{b}_\mathrm{p}, \mathbf{c}_\mathrm{p})$ to those of the conventional unit cell $(\mathbf{a}_\mathrm{c}, \mathbf{b}_\mathrm{c}, \mathbf{c}_\mathrm{c})$ is written as

$$(\mathbf{a}_\mathrm{c}, \mathbf{b}_\mathrm{c}, \mathbf{c}_\mathrm{c}) = (\mathbf{a}_\mathrm{p}, \mathbf{b}_\mathrm{p}, \mathbf{c}_\mathrm{p})\mathbf{M}'. \tag{13}$$

$\mathbf{M}'$ is constructed from three axis directions $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ such as

$$\mathbf{M}' = \begin{pmatrix} e_{x_1} & e_{y_1} & e_{z_1} \\ e_{x_2} & e_{y_2} & e_{z_2} \\ e_{x_3} & e_{y_3} & e_{z_3} \end{pmatrix}, \tag{14}$$

with $\det(\mathbf{M}') > 0$. These axis directions are determined from the rotation axes that characterize the Laue class. The rotation axis direction of each $\boldsymbol{W}_\mathrm{p}$ is found by solving the following equation:

$$\boldsymbol{W}^\mathrm{prop}\mathbf{e} = \mathbf{e}, \tag{15}$$

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

$$(\boldsymbol{W}^\mathrm{prop})^n = \boldsymbol{I}. \tag{16}$$

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

$$\mathbf{S}\mathbf{e}' = \mathbf{0}, \tag{17}$$

where $\mathbf{S} = \sum_{i=1}^{n_\mathrm{pri}} \boldsymbol{W}_i^\mathrm{pri}$. The conditions that the primary, secondary ($\mathbf{e}^\mathrm{sec}$), and ternary ($\mathbf{e}^\mathrm{ter}$) axis directions have to satisfy are listed in Table 1 for Laue classes. For the Laue class $\bar{1}$, $\mathbf{M}'$ is determined so as to make the left-hand side of Equation (13) become the Niggli cell [6,29–32]. For the Laue class $2/m$, the $\mathbf{b}$ axis direction is set as $\mathbf{e}^\mathrm{pri}$ along the two-fold rotation axis by Equation (15). The $\mathbf{a}$ and $\mathbf{c}$ axis directions are found to be perpendicular to $\mathbf{e}^\mathrm{pri}$ by Equation (17). Therefore $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z) = (\mathbf{e}^\mathrm{ter}, \mathbf{e}^\mathrm{pri}, \mathbf{e}^\mathrm{sec})$. For the Laue classes of $4/m$, $4/mmm$, $\bar{3}$, $\bar{3}m$, $6/m$, and $6/mmm$, the $\mathbf{c}$ axis direction is set as $\mathbf{e}^\mathrm{pri}$ along the four or three-fold rotation axis by Equation (15). $\mathbf{e}^\mathrm{sec}$ is found to be perpendicular to $\mathbf{e}^\mathrm{pri}$ by Equation (17), and $\mathbf{e}^\mathrm{ter} = \boldsymbol{W}^\mathrm{pri}\mathbf{e}^\mathrm{sec}$. Therefore $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z) = (\mathbf{e}^\mathrm{sec}, \mathbf{e}^\mathrm{ter}, \mathbf{e}^\mathrm{pri})$.

**Table 1.** Axis directions for Laue classes. $n^{\mathrm{pri}}$, $n^{\mathrm{sec}}$, and $n^{\mathrm{ter}}$ are the rotation orders of the primary ($\boldsymbol{W}^{\mathrm{pri}}$), secondary ($\boldsymbol{W}^{\mathrm{sec}}$), and ternary ($\boldsymbol{W}^{\mathrm{ter}}$) proper rotation matrices, respectively.

| Laue class | Condition implemented in the spglib code |
|---|---|
| $\bar{1}$ | Do nothing |
| $2/m$ | $n^{\mathrm{pri}}=2$ for $\mathbf{b}_{\mathrm{c}}$, $\mathbf{Se}^{\mathrm{sec}}=\mathbf{0}$, $\mathbf{Se}^{\mathrm{ter}}=\mathbf{0}$, $\mathbf{e}^{\mathrm{sec}}\neq\mathbf{e}^{\mathrm{ter}}$ |
| $4/m$ | $n^{\mathrm{pri}}=4$ for $\mathbf{c}_{\mathrm{c}}$, $\mathbf{Se}^{\mathrm{sec}}=\mathbf{0}$, $\mathbf{e}^{\mathrm{ter}}=\boldsymbol{W}^{\mathrm{pri}}\mathbf{e}^{\mathrm{sec}}$ |
| $4/mmm$ | Same as $4/m$ |
| $\bar{3}$ | $n^{\mathrm{pri}}=3$ for $\mathbf{c}_{\mathrm{c}}$, $\mathbf{Se}^{\mathrm{sec}}=\mathbf{0}$, $\mathbf{e}^{\mathrm{ter}}=\boldsymbol{W}^{\mathrm{pri}}\mathbf{e}^{\mathrm{sec}}$ |
| $\bar{3}m$ | Same as $\bar{3}$ |
| $6/m$ | Same as $\bar{3}$ |
| $6/mmm$ | Same as $\bar{3}$ |
| $mmm$ | $n^{\mathrm{pri}}=n^{\mathrm{sec}}=n^{\mathrm{ter}}=2$ for $\mathbf{a}_{\mathrm{c}}$, $\mathbf{b}_{\mathrm{c}}$, $\mathbf{c}_{\mathrm{c}}$ |
| $m\bar{3}$ | Same as $mmm$ |
| $m\bar{3}m$ | $n^{\mathrm{pri}}=n^{\mathrm{sec}}=n^{\mathrm{ter}}=4$ for $\mathbf{a}_{\mathrm{c}}$, $\mathbf{b}_{\mathrm{c}}$, $\mathbf{c}_{\mathrm{c}}$ |

Among the possible sets of $(\mathbf{e}^{\mathrm{sec}}, \mathbf{e}^{\mathrm{ter}}, \mathbf{e}^{\mathrm{pri}})$, one having the smallest $|\det(\mathbf{M}')| \neq 0$ is selected to avoid wrongly-centred $(\mathbf{a}_{\mathrm{c}}, \mathbf{b}_{\mathrm{c}}, \mathbf{c}_{\mathrm{c}})$. For the Laue classes of $mmm$, $m\bar{3}$, and $m\bar{3}m$, three axis directions along two- or four-fold rotation axes are determined by Equation (15). When $\det(\mathbf{M}') < 0$, 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 $\mathbf{M}$ with $\mathbf{M}'$ for the Laue classes of $2/m$, $mmm$, and the rhombohedral system. Otherwise, $\mathbf{M}$ is considered as an identity matrix. For those, the correction matrices are listed in Table 2. The current centering type is easily identified from $\mathbf{M}'$ using Table 3.

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 rhombohedrally-centred hexagonal cell is obtained by $\mathbf{M}'$ in either the obverse or reverse setting. This is transformed to the primitive rhombohedral cell by $\mathbf{M}_{\mathrm{obv}}$ if it is the obverse setting or by $\mathbf{M}_{\mathrm{rev}}$ if it is the reverse setting. Only one of $\mathbf{M}'\mathbf{M}_{\mathrm{obv}}$ or $\mathbf{M}'\mathbf{M}_{\mathrm{rev}}$ has to be an integer matrix, which is chosen as the transformation matrix to a primitive rhombohedral cell.

### 6.3. 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 [27]. The method to decode the Hall symbols is also found in *International Tables for Crystallography Volume B* [28]. The 530 sets of matrix representations are pre-decoded and stored in the spglib source code.

To perform the comparison, the set of 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 step (i) as described in

**Table 2.** Correction matrices $\mathbf{M}$.

| | |
|---|---|
| $2/m$, $A \to C$ | $\mathbf{M}_{2/m,A} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & \bar{1} & 0 \\ 1 & 0 & 0 \end{pmatrix}$ |
| $2/m$, $B \to C$ | $\mathbf{M}_{2/m,B} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ |
| $2/m$, $I \to C$ | $\mathbf{M}_{2/m,I} = \begin{pmatrix} 1 & 0 & \bar{1} \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ |
| $mmm$, $A \to C$ | $\mathbf{M}_{mmm,A} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ |
| $mmm$, $B \to C$ | $\mathbf{M}_{mmm,B} = \mathbf{M}_{2/m,B}$ |
| Obverse hexagonal cell → primitive rhombohedral cell | $\mathbf{M}_{\mathrm{obv}} = \begin{pmatrix} \frac{2}{3} & \frac{\bar{1}}{3} & \frac{\bar{1}}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{\bar{2}}{3} \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$ |
| Reverse hexagonal cell → primitive rhombohedral cell | $\mathbf{M}_{\mathrm{rev}} = \begin{pmatrix} \frac{1}{3} & \frac{\bar{2}}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{\bar{1}}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{\bar{1}}{3} & \frac{1}{3} \end{pmatrix}$ |
| Otherwise, | $\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ |

**Table 3.** Conditions to determine the centering types.

| Centering type | $\det(\mathbf{M}')$ | Row vectors of $\mathbf{M}'$ |
|---|---|---|
| $P$ | 1 | – |
| $A$ | 2 | $^{\exists}i, \sum_j \lvert M'_{ij}\rvert = 1, \lvert M'_{i1}\rvert = 1$ |
| $B$ | 2 | $^{\exists}i, \sum_j \lvert M'_{ij}\rvert = 1, \lvert M'_{i2}\rvert = 1$ |
| $C$ | 2 | $^{\exists}i, \sum_j \lvert M'_{ij}\rvert = 1, \lvert M'_{i3}\rvert = 1$ |
| $I$ (body) | 2 | $i, \sum_j \lvert M'_{ij}\rvert = 2$ |
| $R$ (rhombohedral) | 3 | – |
| $F$ (face) | 4 | – |

Equation (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 $|\mathbf{a}_{\mathrm{c}}| \leq |\mathbf{b}_{\mathrm{c}}|$, $|\mathbf{a}_{\mathrm{c}}| \leq |\mathbf{c}_{\mathrm{c}}|$, or $|\mathbf{b}_{\mathrm{c}}| \leq |\mathbf{c}_{\mathrm{c}}|$, 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 $Pa\bar{3}$, two change-of-basis matrices, $\mathbf{M}'\mathbf{M}$ and $\mathbf{M}'\mathbf{M}\mathbf{Q}'_{Pa\bar{3}}$, are examined to match [8], where $\mathbf{Q}'_{Pa\bar{3}}$ used in the spglib code is given by

$$\mathbf{Q}'_{Pa\bar{3}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & \bar{1} & 0 \\ 1 & 0 & 0 \end{pmatrix}. \tag{18}$$

The matrix representations of the space group operations of $\{(\boldsymbol{W}_{\mathrm{p}}, \boldsymbol{w}_{\mathrm{p}})\}$ in the second stage are given for the primitive cell. Each $(\boldsymbol{W}_{\mathrm{p}}, \boldsymbol{w}_{\mathrm{p}})$ is transformed to that in the conventional unit cell, $(\boldsymbol{W}_{\mathrm{c}}, \boldsymbol{w}_{\mathrm{c}})$,

by $\mathbf{M}'\mathbf{M}\mathbf{Q}'$ as described in Equation (C2). After this transformation, the rotation matrices of $\{\boldsymbol{W}_c\}$ 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 the generators of the space group operations represented in the primitive cell setting. For a system having any centering, $(\boldsymbol{W}_c, \boldsymbol{w}_c)$ is transformed to $(\boldsymbol{W}_X, \boldsymbol{w}_X)$ by using the transformation matrix $\mathbf{P}_X$ as presented in Table 4.

When the space group operation is represented by $(\boldsymbol{W}_X, \boldsymbol{w}_X)$ with reference to the origin $\mathbf{O}$ and by $(\boldsymbol{W}_X, \boldsymbol{w}_X^{\mathrm{d}})$ with respect to $\mathbf{O}^{\mathrm{d}}$, and both matrix representations describe the same operation, they are interconnected by

$$(\boldsymbol{W}_X - \boldsymbol{I})\mathbf{p}_{\mathrm{p}} = \boldsymbol{w}_X - \boldsymbol{w}_X^{\mathrm{d}}, \tag{19}$$

where $\mathbf{p}_{\mathrm{p}}$ is the origin shift from $\mathbf{O}^{\mathrm{d}}$ to $\mathbf{O}$, i.e., $\mathbf{O} = \mathbf{O}^{\mathrm{d}} + \mathbf{p}_{\mathrm{p}}$.

Consider $(\boldsymbol{W}_X^{\mathrm{d}}, \boldsymbol{w}_X^{\mathrm{d}})$ as the reference provided from the dataset, and $(\boldsymbol{W}_X, \boldsymbol{w}_X)$ as derived using this symmetry-finding algorithm, where $\boldsymbol{W}_X^{\mathrm{d}} = \boldsymbol{W}_X$. To determine $\mathbf{p}_{\mathrm{p}}$, at most three matrix representations of the generators are required. For example, using these three generators, we can solve the equation below:

$$\begin{pmatrix} \boldsymbol{W}_{X,1} - \boldsymbol{I} \\ \boldsymbol{W}_{X,2} - \boldsymbol{I} \\ \boldsymbol{W}_{X,3} - \boldsymbol{I} \end{pmatrix} \mathbf{p}_{\mathrm{p}} = \begin{pmatrix} \boldsymbol{w}_{X,1} - \boldsymbol{w}_{X,1}^{\mathrm{d}} \\ \boldsymbol{w}_{X,2} - \boldsymbol{w}_{X,2}^{\mathrm{d}} \\ \boldsymbol{w}_{X,3} - \boldsymbol{w}_{X,3}^{\mathrm{d}} \end{pmatrix} = \Delta\boldsymbol{w}_{\mathrm{p}} \ (\mathrm{mod}\ \mathbb{Z}). \tag{20}$$

This is rewritten as

$$\mathbf{N}\mathbf{p}_{\mathrm{p}} = \Delta\boldsymbol{w}_{\mathrm{p}} \ (\mathrm{mod}\ \mathbb{Z}), \tag{21}$$

where

$$\mathbf{N} = \begin{pmatrix} \boldsymbol{W}_{X,1} - \boldsymbol{I} \\ \boldsymbol{W}_{X,2} - \boldsymbol{I} \\ \boldsymbol{W}_{X,3} - \boldsymbol{I} \end{pmatrix}. \tag{22}$$

A set of solutions is obtained by applying the Smith normal form $\mathbf{D}$ given by

$$\mathbf{D} = \mathbf{U}\mathbf{N}\mathbf{V}, \tag{23}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the unimodular matrices. In the case with three generators, $\mathbf{N}$ is a $9 \times 3$ matrix and its Smith normal form $\mathbf{D}$ becomes

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{24}$$

The $3 \times 9$ matrix $\mathbf{T}$ of the inverse diagonal elements of $\mathbf{D}$ is made as

$$\mathbf{T} = \begin{pmatrix} \frac{1}{\lambda_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\lambda_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\lambda_3} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{25}$$

When $\lambda_n = 0$, the corresponding elements of $\mathbf{T}$ are set to 0. Since $\mathbf{D}\mathbf{V}^{-1}\mathbf{p}_{\mathrm{p}} = \mathbf{U}\Delta\boldsymbol{w}_{\mathrm{p}}$, one of the solutions is given by

$$\mathbf{p}_{\mathrm{p}} = \mathbf{V}\mathbf{T}\mathbf{U}\Delta\boldsymbol{w}_{\mathrm{p}}. \tag{26}$$

A Python script was written to compute the matrices $\mathbf{VTU}$ for crystal systems and their centering types and axis settings using the SageMath code [33]. The precomputed $\mathbf{VTU}$ matrices are stored in the spglib source code together with the corresponding sets of the rotation parts of the generators in the primitive cell setting.

With the origin shift $\mathbf{p}_{\mathrm{p}}$ obtained in Equation (26), the translation parts of the space group operations are compared with the datasets by Equation (19). Finally, the Hall symbol is identified by verifying that $\{(\boldsymbol{W}_X, \boldsymbol{w}_X)\}$ is mapped to $\{(\boldsymbol{W}_X^{\mathrm{d}}, \boldsymbol{w}_X^{\mathrm{d}})\}$.

For subsequent use, the basis vectors of the primitive cell as obtained in Equation (8) are transformed to those of the conventional unit cell by

**Table 4.** 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 $\mathsf{P}_X$ 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.

$$\mathsf{P}_A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{\bar{1}}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad \mathsf{P}_B = \begin{pmatrix} \frac{1}{2} & 0 & \frac{\bar{1}}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}, \quad \mathsf{P}_C = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{\bar{1}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathsf{P}_I = \begin{pmatrix} \frac{\bar{1}}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{\bar{1}}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{\bar{1}}{2} \end{pmatrix}, \quad \mathsf{P}_F = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}, \quad \mathsf{P}_R = \begin{pmatrix} \frac{2}{3} & \frac{\bar{1}}{3} & \frac{\bar{1}}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{\bar{2}}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}.$$

$$(\mathbf{a}_c, \mathbf{b}_c, \mathbf{c}_c) = (\mathbf{a}_p, \mathbf{b}_p, \mathbf{c}_p)\mathbf{M}'\mathbf{M}\mathbf{Q}'. \quad (27)$$

Furthermore, the origin shift in the coordinate system of the conventional unit cell is given by

$$\mathbf{p}_c = \mathbf{P}_X\mathbf{p}_p. \quad (28)$$

### 6.4. Step (k): Failure of identification of space group type

When the Hall symbol matching at step (j) fails, the tolerance value is shortened, and the current set of the space group operations is re-examined in step (l). Following this, the sequence from steps (i) to (l) is reiterated. This process continues until a Hall symbol is successfully identified. In cases where this loop is executed numerous times without success, the entire procedure is restarted from step (a), employing a tolerance value shortened from the one utilized in the last attempt at step (a).

### 6.5. Step (l): Thinning out space group operations

The space group operations in $\{(\boldsymbol{W}_p, \boldsymbol{w}_p)\}$ are re-evaluated in a way similar to step (g) with the shortened tolerance value. As a result, certain space group operations in $\{(\boldsymbol{W}_p, \boldsymbol{w}_p)\}$ may be excluded. Since $\boldsymbol{w}_p$ 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.

## 7. 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 and a distortion-free crystal structure of the conventional unit cell, which is derived from the input unit cell, is suggested.

### 7.1. Step (m): Removal of distortion from basis vectors

The tolerance introduced in the symmetry identification process may lead to some distortion in $(\mathbf{a}_c, \mathbf{b}_c, \mathbf{c}_c)$ compared to their ideal values for the determined lattice system. This distortion is rectified by averaging the lengths of symmetrically equivalent basis vectors to align with the expected values for the lattice system. The specific lattice system conditions required for this adjustment are detailed in Appendix D, where the basis vectors $(\mathbf{a}_s, \mathbf{b}_s, \mathbf{c}_s)$ are adjusted for an intuitive alignment with the Cartesian axes. In this rectification process, the basis vectors $(\mathbf{a}_c, \mathbf{b}_c, \mathbf{c}_c)$ are subject to a rotation in Cartesian coordinates, approximately represented by the equation:

$$(\mathbf{a}_s, \mathbf{b}_s, \mathbf{c}_s) \approx (\boldsymbol{R}\mathbf{a}_c, \boldsymbol{R}\mathbf{b}_c, \boldsymbol{R}\mathbf{c}_c), \quad (29)$$

where $(\mathbf{a}_s, \mathbf{b}_s, \mathbf{c}_s)$ represent the idealized basis vectors of the conventional unit cell.

### 7.2. Step (n): Reconstruction of matrix representations of space group operations of input unit cell

The set of matrix representations of the space group operations corresponding to the Hall symbol, $\{(\boldsymbol{W}^d, \boldsymbol{w}^d)\}$, is obtained from the dataset. This is transformed to that of the input unit cell $\{(\boldsymbol{W}_i, \boldsymbol{w}_i)\}$ using the following procedure.

The basis vectors $(\mathbf{a}_c, \mathbf{b}_c, \mathbf{c}_c)$, as specified by Equation (27), follow the symmetry of $\{(\boldsymbol{W}^d, \boldsymbol{w}^d)\}$. However, the basis vectors resulting from applying a rotation $\boldsymbol{W}^{d*} \in \{\boldsymbol{W}^d\}$ are also valid:

$$(\mathbf{a}'_c, \mathbf{b}'_c, \mathbf{c}'_c) = (\mathbf{a}_c, \mathbf{b}_c, \mathbf{c}_c)\boldsymbol{W}^{d*}. \quad (30)$$

Consequently, as a potentially more suitable choice of basis vectors, $(\mathbf{a}_c, \mathbf{b}_c, \mathbf{c}_c)$ may be replaced by $(\mathbf{a}'_c, \mathbf{b}'_c, \mathbf{c}'_c)$. This replacement aims to minimize the following expression:

$$|\mathbf{a}_s - \mathbf{a}'_c|^2 + |\mathbf{b}_s - \mathbf{b}'_c|^2 + |\mathbf{c}_s - \mathbf{c}'_c|^2. \quad (31)$$

As this involves a transformation of the coordinate system, $\mathbf{p}_c$ as given by Equation (28) is also transformed according to:

$$\mathbf{p}'_c = \boldsymbol{W}^{d*-1}(\mathbf{p}_c - \boldsymbol{w}^{d*}). \quad (32)$$

Details of this coordinate transformation are provided in Appendix E. For conventional unit cells having any centering, $(\boldsymbol{W}^{d*}, \boldsymbol{w}^{d*})$ that gives the shortest $\mathbf{p}'_c$ is employed. With the use of $\mathbf{p}'_c$, the translation parts of $\{(\boldsymbol{W}^d, \boldsymbol{w}^d)\}$ are redefined as:

$$\mathbf{w}^{d'} = (\boldsymbol{W}^d - \mathbf{I})\mathbf{p}'_c + \boldsymbol{w}^d. \quad (33)$$

Every $(\boldsymbol{W}^d, \boldsymbol{w}^{d'})$ is transformed to that of the primitive cell $(\boldsymbol{W}^d_p, \boldsymbol{w}^{d'}_p)$ as the transformation detailed in Appendix C. The transformation matrix $\mathbf{Q}_{c\to p}$ is defined by the equation:

$$(\mathbf{a}_p, \mathbf{b}_p, \mathbf{c}_p) = (\mathbf{a}'_c, \mathbf{b}'_c, \mathbf{c}'_c)\mathbf{Q}_{c\to p}. \quad (34)$$

Note that $\mathbf{Q}^{-1}_{c\to p}$ is an integer matrix. To ensure that $\{(\boldsymbol{W}^d_p, \boldsymbol{w}^{d'}_p)\}$ represents the set of coset representatives, only one space group operation is included in $\{(\boldsymbol{W}^d_p, \boldsymbol{w}^{d'}_p)\}$ for each rotation part $\boldsymbol{W}^d_p$.

Then, $\{(\boldsymbol{W}_{\rm p}^{\rm d}, \boldsymbol{w}_{\rm p}^{\rm d'})\}$ is transformed to that of the input unit cell $\{(\boldsymbol{W}_{\rm i,p}\boldsymbol{w}_{\rm i,p})\}$. The transformation matrix $\mathbf{Q}_{\rm p\to i}$ is defined by the equation:

$$(\mathbf{a}_{\rm i}, \mathbf{b}_{\rm i}, \mathbf{c}_{\rm i}) \approx (\mathbf{a}_{\rm p}, \mathbf{b}_{\rm p}, \mathbf{c}_{\rm p})\mathbf{Q}_{\rm p\to i}, \tag{35}$$

where the elements of the matrix $\mathbf{Q}_{\rm p\to i}$ are rounded to their nearest integer values, thereby constituting an integer matrix. Thus obtained $\boldsymbol{W}_{\rm i,p}$ may not be an integer matrix if the order of the lattice point group of $(\mathbf{a}_{\rm i}, \mathbf{b}_{\rm i}, \mathbf{c}_{\rm i})$ is smaller than that of $(\mathbf{a}_{\rm p}, \mathbf{b}_{\rm p}, \mathbf{c}_{\rm p})$. Those space group operations with non-integer matrices of $\boldsymbol{W}_{\rm i,p}$ are excluded from $\{(\boldsymbol{W}_{\rm i,p}, \boldsymbol{w}_{\rm i,p})\}$.

When the input unit cell is not a primitive cell, $\{(\boldsymbol{W}_{\rm i,p}, \boldsymbol{w}_{\rm i,p})\}$ is extended by a set of lattice point vectors in the input unit cell, $\{\boldsymbol{t}_j\}$,

$$\{(\boldsymbol{W}_{\rm i}, \boldsymbol{w}_{\rm i})\} = \bigcup_j (\boldsymbol{I}, \boldsymbol{t}_j)\{(\boldsymbol{W}_{\rm i,p}, \boldsymbol{w}_{\rm i,p})\}. \tag{36}$$

The lattice point vectors of $\{\boldsymbol{t}_j\}$ are easily obtained from $\mathbf{Q}_{\rm p\to i}$ [14].

### 7.3. Step (o): Removal of distortion from point coordinates and determination of Wyckoff positions

Point coordinates in the primitive cell $\boldsymbol{x}_{\rm p}$ are transformed to those in the conventional unit cell $\boldsymbol{x}_{\rm c}$ by

$$\boldsymbol{x}_{\rm c} = \mathbf{Q}_{\rm c\to p}\boldsymbol{x}_{\rm p} + \mathbf{p}'_{\rm c}, \tag{37}$$

Applying $\{(\boldsymbol{W}^{\rm d}, \boldsymbol{w}^{\rm d})\}$ to $\{\boldsymbol{x}_{\rm c}\}$, symmetrically independent points and sets of symmetrically equivalent points are obtained.

Site symmetry operations of $\{(\boldsymbol{W}_{\boldsymbol{x},i}, \boldsymbol{w}_{\boldsymbol{x},i})\}$ at $\boldsymbol{x}_{\rm c}$ are the space group operations that leave the coordinates of a point $\boldsymbol{x}_{\rm c}$ unchanged, i.e.

$$(\boldsymbol{W}_{\boldsymbol{x},i}, \boldsymbol{w}_{x,i})\boldsymbol{x}_{\rm c} = \boldsymbol{x}_{\rm c}. \tag{38}$$

$\{(\boldsymbol{W}_{\boldsymbol{x},i}, \boldsymbol{w}_{\boldsymbol{x},i})\}$ is expected to form the site symmetry group $\mathbb{S}_{\boldsymbol{x}}$ of the finite order $|\mathbb{S}_{\boldsymbol{x}}|$. Using $\mathbb{S}_{\boldsymbol{x}}$, the special position operator [7] $(\boldsymbol{W}_{\boldsymbol{x}}^{\rm sp}, \boldsymbol{w}_{\boldsymbol{x}}^{\rm sp})$ is defined as

$$(\boldsymbol{W}_{\boldsymbol{x}}^{\rm sp}, \boldsymbol{w}_{\boldsymbol{x}}^{\rm sp}) = \frac{1}{|\mathbb{S}_{\boldsymbol{x}}|}\sum_{i=1}^{|\mathbb{S}_x|}(\boldsymbol{W}_{\boldsymbol{x},i}, \boldsymbol{w}_{\boldsymbol{x},i}). \tag{39}$$

Point coordinates $\boldsymbol{x}_{\rm c}$ can be slightly dislocated from the exact location. By Equation (39), the exact location $\boldsymbol{x}_{\mathbb{S}}$ of $\boldsymbol{x}_{\rm c}$ is obtained by

$$\boldsymbol{x}_{\mathbb{S}} = (\boldsymbol{W}_{\boldsymbol{x}}^{\rm sp}, \boldsymbol{w}_{\boldsymbol{x}}^{\rm sp})\boldsymbol{x}_{\rm c}. \tag{40}$$

In the spglib implementation, $\{(\boldsymbol{W}_{\boldsymbol{x},i}, \boldsymbol{w}_{\boldsymbol{x},i})\}$ is obtained from $\{(\boldsymbol{W}^{\rm d}, \boldsymbol{w}^{\rm d})\}$. Since $\{(\boldsymbol{W}^{\rm d}, \boldsymbol{w}^{\rm d})\}$ is the coset representatives of the lattice translation group of the conventional unit cell, i.e. it is a finite set unlike the space group, Equation (38) is examined using the condition (6) as

$$\Delta\boldsymbol{x}_i = (\boldsymbol{W}_{\boldsymbol{x},i}, \boldsymbol{w}_{\boldsymbol{x},i})\boldsymbol{x}_{\rm c} - \boldsymbol{x}_{\rm c}. \tag{41}$$

Using Equation (41), Equations (39) and (40) are rewritten as

$$\boldsymbol{x}_{\mathbb{S}} = \frac{1}{|\mathbb{S}_{\boldsymbol{x}}|}\sum_i^{|\mathbb{S}_x|}(\Delta\boldsymbol{x}_i - \lfloor\Delta\boldsymbol{x}_i\rceil) + \boldsymbol{x}_{\rm c}. \tag{42}$$

The number of the symmetrically equivalent points of the point $\boldsymbol{x}_{\mathbb{S}}$ in the conventional unit cell is called multiplicity $M_{\boldsymbol{x}}$. Note that $M_{\boldsymbol{x}}$ is defined with respect to the conventional unit cell but not the primitive cell as following the convention of the *International Tables for Crystallography Volume A* [1]. These have to satisfy the following relationship:

$$|\mathbb{S}_{\boldsymbol{x}}|M_{\boldsymbol{x}} = |\mathbb{S}/\mathbb{T}|\det(\mathbf{P}_X^{-1}), \tag{43}$$

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

The Wyckoff letter of $\boldsymbol{x}_{\mathbb{S}}$ is determined using *Coordinates* in the Wyckoff position dataset. *Coordinates* are listed in the *International Tables for Crystallography Volume A* [1]. The dataset in the spglib code was provided by Y. Seto [34] for all the Hall symbols. The first entries of *Coordinates* of Wyckoff positions for each Hall symbol is necessary to match $\boldsymbol{x}_{\mathbb{S}}$ with a Wyckoff letter. All those first entries of *Coordinates* are stored in the 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

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}.$$

The $3\times3$ and $3\times1$ matrices are encoded and stored in the spglib source code. Matching with the dataset is performed by examining $(x, x, \frac{1}{2})\boldsymbol{x}_{\mathbb{S}} = \boldsymbol{x}_{\mathbb{S}} \pmod{\mathbb{Z}}$. The multiplicity $M_{\boldsymbol{x}}$ is also stored in the dataset. With this $M_{\boldsymbol{x}}$, Equation (43) is verified. If this fails, the procedure restarts from step (a) with the tolerance value shortened from that used last time at step (a).

## 8. Summary

The spglib code is designed for the identification and symmetrization of crystal structures, which 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 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 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.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This work was supported by JSPS KAKENHI Grant Numbers JP21K04632, JP24K08021 and JP24H00190.

## References

[1] Aroyo MI. International tables for crystallography. 6th ed. Vol. A. Hoboken, NJ: Wiley; 2016.
[2] Stokes HT, Hatch DM. FINDSYM: program for identifying the space-group symmetry of a crystal. J Appl Crystallogr. 2005;38(1):237. doi: 10.1107/S0021889804031528
[3] FINDSYM. Available from: http://stokes.byu.edu/iso/findsym.php
[4] ISOTROPY Software Suite. Available from: http://stokes.byu.edu/iso/isotropy.php
[5] Grosse-Kunstleve RW, Sauter NK, Moriarty NW, et al. The computational crystallography toolbox: crystallographic algorithms in a reusable software framework. J Appl Crystallogr. 2002;35(1):126. doi: 10.1107/S0021889801017824
[6] Grosse-Kunstleve RW, Sauter NK, Adams PD. Numerically stable algorithms for the computation of reduced unit cells. Acta Crystallogr A Found Crystallogr. 2004;60(1):1–6. doi: 10.1107/S010876730302186X
[7] Grosse-Kunstleve RW, Adams PD. Algorithms for deriving crystallographic space-group information. II. Treatment of special positions. Acta Crystallogr A Found Crystallogr. 2002;58(1):60. doi: 10.1107/S0108767301016658
[8] Grosse-Kunstleve RW. Algorithms for deriving crystallographic space-group information. Acta Crystallogr A Found Crystallogr. 1999;55(2):383. doi: 10.1107/S0108767398010186
[9] Computational Crystallography Toolbox. Available from: https://cctbx.github.io/
[10] Hicks D, Oses C, Gossett E, et al. *AFLOW-SYM*: platform for the complete, automatic and self-consistent symmetry analysis of crystals. Acta Crystallogr A Found Adv. 2018;74(3):184. doi: 10.1107/S2053273318003066
[11] Spglib. version 2.3.1. https://github.com/spglib/spglib
[12] Kühne TD, Iannuzzi M, Del Ben M, et al. CP2K: an electronic structure and molecular dynamics software package - quickstep: efficient and accurate electronic structure calculations. J Chem Phys. 2020;152(19):194103. doi: 10.1063/5.0007045
[13] Tancogne-Dejean N, Oliveira MJT, Andrade X, et al. Octopus, a computational framework for exploring light-driven phenomena and quantum dynamics in extended and finite systems. J Chem Phys. 2020;152(12):124119. doi: 10.1063/1.5142502
[14] Togo A, Chaput L, Tadano T, et al. Implementation strategies in phonopy and phono3py. J Phys: Condens Matter. 2023;35(35):353001. doi: 10.1088/1361-648X/acd831
[15] Ong SP, Richards WD, Jain A, et al. Python materials genomics (pymatgen): a robust, open-source python library for materials analysis. J Comput Mater Sci. 2013;68:314. doi: 10.1016/j.commatsci.2012.10.028
[16] Larsen AH, Mortensen JJ, Blomqvist J, et al. The atomic simulation environment—a python library for working with atoms. J Phys: Condens Matter. 2017;29(27):273002. doi: 10.1088/1361-648X/aa680e
[17] Hanwell MD, Curtis DE, Lonie DC, et al. Avogadro: an advanced semantic chemical editor, visualization, and analysis platform. J Chem inform. 2012;4(1):17. doi: 10.1186/1758-2946-4-17
[18] Jain A, Ong SP, Hautier G, et al. Commentary: the materials project: a materials genome approach to accelerating materials innovation. APL Mater. 2013;1(1):011002. doi: 10.1063/1.4812323
[19] Scheidgen M, Himanen L, Ladines AN, et al. Nomad: a distributed web-based platform for managing materials science research data. J Open Source Softw. 2023;8(90):5388. doi: 10.21105/joss.05388
[20] Nye JF. Physical properties of crystals. Oxford, UK: Oxford University Press; 1985.
[21] Mller U. Symmetry relationships between crystal structures: applications of crystallographic group theory in crystal chemistry. Oxford, UK: Oxford University Press; 2013.
[22] El-Batanouny M, Wooten F. Symmetry and condensed matter physics: a computational ApproachPhysical properties of crystals. Cambridge, UK: Cambridge University Press; 2008.
[23] Shinohara K, Togo A, Tanaka I. Algorithms for magnetic symmetry operation search and identification of magnetic space group from magnetic crystal structure, acta crystallographica section a. Acta Crystallogr Sect A Found And Adv. 2023;79(5):390. doi: 10.1107/S2053273323005016
[24] Hannemann A, Hundt R, Schön JC, et al. A new algorithm for space-group determination. J Appl Crystallogr. 1998;31(6):922. doi: 10.1107/S0021889898008735
[25] Lamparski M. Large-scale atomistic computations of the phonons in twisted bilayer graphene [Phd thesis]. Troy (NY): Department of Physics, Applied Physics, and Astronomy Rens-selaer Polytechnic Institute; 2020. Available from: https://hdl.handle.net/20.500.13015/2605

[26] Delaunay B. Neue darstellung der geometrischen kristallographie. Z für Kristallogr - Crystalline Mater. 1933;84(1–6):109. doi: 10.1524/zkri.1933.84.1.109
[27] Hall SR. Space-group notation with an explicit origin. Acta Cryst A. 1981;37(4):517. doi: 10.1107/S0567739481001228
[28] Shmueli U. International tables for crystallography, vol. B. Hoboken, NJ: Wiley; 2008.
[29] Niggli P. Krystallographische und strukturtheoretische Grundbegriffe. In: Handbuch der Experimentalphysik. Vol. 7. Leipzig: Akademische Verlagsgesellschaft; 1928. p. 108–176.
[30] Gruber B. The relationship between reduced cells in a general Bravais lattice. Acta Cryst A. 1973;29(4):433. doi: 10.1107/S0567739473001063
[31] Křivý I, Gruber B. A unified algorithm for determining the reduced (Niggli) cell. Acta Cryst A. 1976;32(2):297. doi: 10.1107/S0567739476000636
[32] Niggli cell. Available from: http://atztogo.github.io/niggli/
[33] SageMath, the Sage Mathematics Software System. Available from: http://www.sagemath.org
[34] Seto's Home Page. Available from: https://yseto.net/?page_id=29&lang=en

## 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 A* [1] and Refs. [7,8].

### *A.1. Basis vectors* $(\mathbf{a}, \mathbf{b}, \mathbf{c})$

Basis vectors are represented by three column vectors:

$$\mathbf{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}, \mathbf{c} = \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix}, \tag{A1}$$

in the Cartesian coordinates.

### *A.2. Atomic point coordinates* $\boldsymbol{x}$

Coordinates of an atomic point $\boldsymbol{x}$ are represented by three values relative to basis vectors as follows:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}. \tag{A2}$$

A position vector $\boldsymbol{x}$ in the Cartesian coordinates are obtained by

$$\boldsymbol{x} = (\mathbf{a}, \mathbf{b}, \mathbf{c})\boldsymbol{x}. \tag{A3}$$

### *A.3. Metric tensor G and point-group operation of lattice W*

The metric tensor is defined by

$$\boldsymbol{G} = (\mathbf{a}, \mathbf{b}, \mathbf{c})^\top (\mathbf{a}, \mathbf{b}, \mathbf{c}). \tag{A4}$$

A rotation matrix $\boldsymbol{W}$ is applied to the basis vectors such as,

$$(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}) = (\mathbf{a}, \mathbf{b}, \mathbf{c})\boldsymbol{W}. \tag{A5}$$

The metric tensor of $(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}})$ is given by

$$\begin{aligned} \tilde{\boldsymbol{G}} &= (\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}})^\top (\tilde{\mathbf{a}}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}}) \\ &= \boldsymbol{W}^\top (\mathbf{a}, \mathbf{b}, \mathbf{c})^\top (\mathbf{a}, \mathbf{b}, \mathbf{c}) \boldsymbol{W} \\ &= \boldsymbol{W}^\top \boldsymbol{G} \boldsymbol{W}. \end{aligned} \tag{A6}$$

The lattice point group operations are obtained by searching matrices $\{\boldsymbol{W}\}$ that satisfy $\boldsymbol{G} = \boldsymbol{W}^\top \boldsymbol{G} \boldsymbol{W}$, where $\boldsymbol{W}$ is the integer matrix and the determinant is either 1 or $-1$.

### *A.4. Space group operation* $(\boldsymbol{W}, \boldsymbol{w})$

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 Equation (A5), a point in direct space, $\boldsymbol{x}$, is transformed to a point $\tilde{\boldsymbol{x}}$ by a rotation by

$$\tilde{x} = \boldsymbol{W}\boldsymbol{x}. \tag{A7}$$

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

$$\tilde{\boldsymbol{x}} = (\boldsymbol{W}, \boldsymbol{w})\boldsymbol{x} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{w}. \tag{A8}$$

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

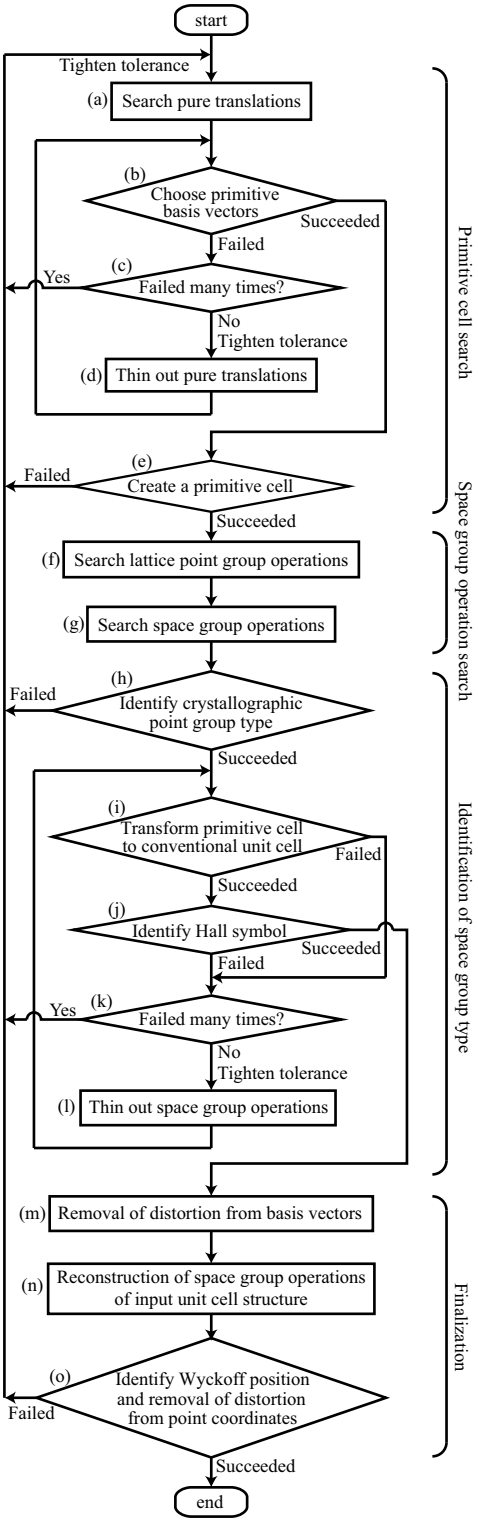



**Figure A1.** Flowchart of algorithm in the spglib code.

### *A.5. Transformation of coordinate system* $(\boldsymbol{P}, \boldsymbol{p})$

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

$$(\mathbf{a}, \mathbf{b}, \mathbf{c}) = (\mathbf{a}_\mathrm{s}, \mathbf{b}_\mathrm{s}, \mathbf{c}_\mathrm{s})\boldsymbol{P}, \tag{A9}$$

where $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ and $(\mathbf{a}_\mathrm{s}, \mathbf{b}_\mathrm{s}, \mathbf{c}_\mathrm{s})$ 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

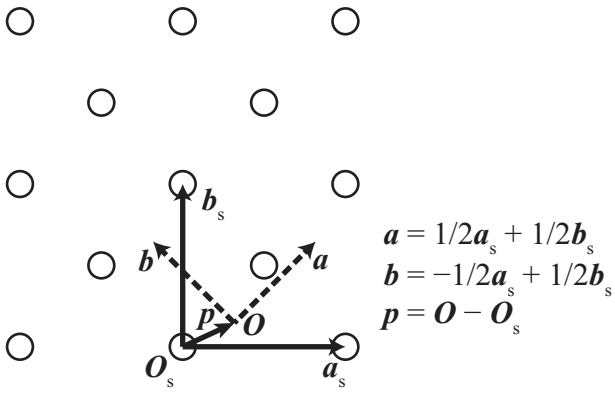



**Figure A2.** Transformation of coordinate system.

the Cartesian coordinates, but just changes the choices of the basis vectors. The origin shift $\boldsymbol{p}$ gives the vector from the origin of an original coordinate system $\boldsymbol{O}_s$ to that of any other coordinate system $\boldsymbol{O}$, which is written as

$$\boldsymbol{p} = \boldsymbol{O} - \boldsymbol{O}_s. \tag{A10}$$

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 $\boldsymbol{x}_s$ and those in the transformed coordinate system $\boldsymbol{x}$ are related by

$$\boldsymbol{x}_s = (\boldsymbol{P}, \boldsymbol{p})\boldsymbol{x} = \boldsymbol{P}\boldsymbol{x} + \boldsymbol{p}, \tag{A11}$$

where $\boldsymbol{p}$ is given with respect to the original basis vectors. Equivalently,

$$\boldsymbol{x} = \boldsymbol{P}^{-1}\boldsymbol{x}_s - \boldsymbol{P}^{-1}\boldsymbol{p}. \tag{A12}$$

An illustration is presented in Figure A2. In this case, the following $\boldsymbol{P}$ is applied:

$$\boldsymbol{P} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \bar{\frac{1}{2}} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{A13}$$

## 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 [1]. The rotation type is identified using Table B1 with the determinant and trace of $\boldsymbol{W}$. The crystal class is found comparing the list of numbers of the rotation types with Table B2. 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 $(\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i)$ are transformed to another basis vectors $(\mathbf{a}_f, \mathbf{b}_f, \mathbf{c}_f)$ by a change-of-basis matrix $\mathbf{Q}$ as

$$(\mathbf{a}_f, \mathbf{b}_f, \mathbf{c}_f) = (\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i)\mathbf{Q}, \tag{C1}$$

the matrix representation of the space group operation $(\boldsymbol{W}_i, \boldsymbol{w}_i)$ is transformed to

$$(\boldsymbol{W}_f, \boldsymbol{w}_f) = (\mathbf{Q}^{-1}\boldsymbol{W}_i\mathbf{Q}, \mathbf{Q}^{-1}\boldsymbol{w}_i). \tag{C2}$$

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 $(\mathbf{a}_p, \mathbf{b}_p, \mathbf{c}_p)$ and $(\mathbf{a}_{np}, \mathbf{b}_{np}, \mathbf{c}_{np})$, respectively. A change-of-basis matrix relates them as

$$(\mathbf{a}_{np}, \mathbf{b}_{np}, \mathbf{c}_{np}) = (\mathbf{a}_p, \mathbf{b}_p, \mathbf{c}_p)\mathbf{Q}. \tag{C3}$$

We expect

$$\mathbf{W}_{np} = \mathbf{Q}^{-1}\boldsymbol{W}_p\mathbf{Q}. \tag{C4}$$

However, when $\mathbf{Q}$ breaks the point group symmetry of $(\mathbf{a}_p, \mathbf{b}_p, \mathbf{c}_p)$, $\boldsymbol{W}_{np}$ becomes a non-integer matrix. This situation arises when the lattice point group types associated with the basis vectors $(\mathbf{a}_p, \mathbf{b}_p, \mathbf{c}_p)$ and $(\mathbf{a}_{np}, \mathbf{b}_{np}, \mathbf{c}_{np})$ of the primitive and non-primitive cells, respectively, are not equivalent.

## 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.

### *D.1. Triclinic*

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

### *D.2. Monoclinic*

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

### *D.3. Orthorhombic*

(1) $\alpha = \beta = \gamma = 90°$. (2) **a** is aligned with $+x$ direction of Cartesian coordinates. (3) **b** is aligned with $+y$ direction of Cartesian coordinates. (4) **c** is aligned with $+z$ direction of Cartesian coordinates.

**Table B1.** Look-up table to identify the types of the rotation parts of the space group operations from their matrix representations.

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

**Table B2.** Look-up table of crystal class with the numbers of rotation types.

| Crystal system | Crystal class | Laue class | Numbers of types of $\boldsymbol{W}$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | −6 | −4 | −3 | −2 | −1 | 1 | 2 | 3 | 4 | 6 |
| Triclinic | 1 | $\bar{1}$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| | $\bar{1}$ | | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Monoclinic | 2 | $2/m$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| | $m$ | | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| | $2/m$ | | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Orthorhombic | 222 | $mmm$ | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 0 |
| | $mm2$ | | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 0 | 0 |
| | $mmm$ | | 0 | 0 | 0 | 3 | 1 | 1 | 3 | 0 | 0 | 0 |
| Tetragonal | 4 | $4/m$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 0 |
| | $\bar{4}$ | | 0 | 2 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| | $4/m$ | | 0 | 2 | 0 | 1 | 1 | 1 | 1 | 0 | 2 | 0 |
| | 422 | $4/mmm$ | 0 | 0 | 0 | 0 | 0 | 1 | 5 | 0 | 2 | 0 |
| | $4mm$ | | 0 | 0 | 0 | 4 | 0 | 1 | 1 | 0 | 2 | 0 |
| | $\bar{4}2m$ | | 0 | 2 | 0 | 2 | 0 | 1 | 3 | 0 | 0 | 0 |
| | $4/mmm$ | | 0 | 2 | 0 | 5 | 1 | 1 | 5 | 0 | 2 | 0 |
| Trigonal | 3 | $\bar{3}$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 |
| | $\bar{3}$ | | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 2 | 0 | 0 |
| | 32 | $\bar{3}m$ | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 0 | 0 |
| | $3m$ | | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 2 | 0 | 0 |
| | $\bar{3}m$ | | 0 | 0 | 2 | 3 | 1 | 1 | 3 | 2 | 0 | 0 |
| Hexagonal | 6 | $6/m$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 0 | 2 |
| | $\bar{6}$ | | 2 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 |
| | $6/m$ | | 2 | 0 | 2 | 1 | 1 | 1 | 1 | 2 | 0 | 2 |
| | 622 | $6/mmm$ | 0 | 0 | 0 | 0 | 0 | 1 | 7 | 2 | 0 | 2 |
| | $6mm$ | | 0 | 0 | 0 | 6 | 0 | 1 | 1 | 2 | 0 | 2 |
| | $\bar{6}2m$ | | 2 | 0 | 0 | 4 | 0 | 1 | 3 | 2 | 0 | 0 |
| | $6/mmm$ | | 2 | 0 | 2 | 7 | 1 | 1 | 7 | 2 | 0 | 2 |
| Cubic | 23 | $m\bar{3}$ | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 8 | 0 | 0 |
| | $m\bar{3}$ | | 0 | 0 | 8 | 3 | 1 | 1 | 3 | 8 | 0 | 0 |
| | 432 | $m\bar{3}m$ | 0 | 0 | 0 | 0 | 0 | 1 | 9 | 8 | 6 | 0 |
| | $\bar{4}3m$ | | 0 | 6 | 0 | 6 | 0 | 1 | 3 | 8 | 0 | 0 |
| | $m\bar{3}m$ | | 0 | 6 | 8 | 9 | 1 | 1 | 9 | 8 | 6 | 0 |

### *D.4. Tetragonal*

(1) $\alpha = \beta = \gamma = 90°$. (2) $a = b$. (3) **a** is aligned with $+x$ direction of Cartesian coordinates. (4) **b** is aligned with $+y$ direction of Cartesian coordinates. (5) **c** is aligned with $+z$ direction of Cartesian coordinates.

### *D.5. Rhombohedral*

(1) $\alpha = \beta = \gamma$. (2) $a = b = c$. (3) When projected onto $x - y$ plane in Cartesian coordinates, **a**, **b**, and **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 **a**, **b**, and **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° counter-clockwise from the $+x$ direction in Cartesian coordinates, with $\mathbf{b}_{xy}$ and $\mathbf{c}_{xy}$ positioned at angles of 120° and 240° counter-clockwise from $\mathbf{a}_{xy}$, respectively. (6) $\alpha_{xy} = \beta_{xy} = \gamma_{xy} = 120°$. (7) $a_{xy} = b_{xy} = c_{xy}$. (8) $a_z = b_z = c_z$.

### *D.6. Hexagonal*

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

### *D.7. Cubic*

(1) $\alpha = \beta = \gamma = 90°$. (2) $a = b = c$. (3) **a** is aligned with $+x$ direction of Cartesian coordinates. (4) **b** is aligned with $+y$ direction of Cartesian coordinates. (5) **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 $(\boldsymbol{W}, \boldsymbol{w})$ as a transformation of the coordinate system, as detailed in Appendix

A.5, the new point coordinates $\boldsymbol{x}_{\text{new}}$ are related to the original point coordinates $\boldsymbol{x}_{\text{orig}}$ through Equation (A11) by

$$\boldsymbol{x}_{\text{orig}} = (\boldsymbol{W}, \boldsymbol{w})\boldsymbol{x}_{\text{new}}. \tag{E1}$$

By definition (A9),

$$(\mathbf{a}_{\text{new}}, \mathbf{b}_{\text{new}}, \mathbf{c}_{\text{new}}) = (\mathbf{a}_{\text{orig}}, \mathbf{b}_{\text{orig}}, \mathbf{c}_{\text{orig}})\boldsymbol{W}. \tag{E2}$$

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

$$\{\boldsymbol{x}_{\text{new}}\} = \{\boldsymbol{x}_{\text{orig}}\} \ (\text{mod } \mathbb{Z}). \tag{E3}$$

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

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

$$\boldsymbol{x}_{\text{orig}} = (\boldsymbol{I}, \boldsymbol{p})\boldsymbol{x}_{\boldsymbol{p}}. \tag{E4}$$

From Equations. (E1) and (E4), $\boldsymbol{x}_{\text{new}}$ and $\boldsymbol{x}_{\boldsymbol{p}}$ are related as

$$\boldsymbol{x}_{\text{new}} = (\boldsymbol{W}, \boldsymbol{w})^{-1}(\boldsymbol{I}, \boldsymbol{p})\boldsymbol{x}_{\boldsymbol{p}} = (\boldsymbol{W}^{-1}, -\boldsymbol{W}^{-1}\boldsymbol{w})(\boldsymbol{I}, \boldsymbol{p})\boldsymbol{x}_{\boldsymbol{p}}. \tag{E5}$$

Viewed from the new coordinate system, The origin shift $\boldsymbol{p}_{\text{new}}$ is determined by setting $\boldsymbol{x}_{\boldsymbol{p}} = \boldsymbol{0}$,

$$\boldsymbol{p}_{\text{new}} = (\boldsymbol{W}^{-1}, -\boldsymbol{W}^{-1}\boldsymbol{w})\boldsymbol{p} = \boldsymbol{W}^{-1}(\boldsymbol{p} - \boldsymbol{w}). \tag{E6}$$