Consistent representations of and conversions between 3D rotations

Modelling Simul. Mater. Sci. Eng. **23** (2015) 083501 (22pp) doi:10.1088/0965-0393/23/8/083501

Tutorial

Consistent representations of and conversions between 3D rotations

D Rowenhorst[1], A D Rollett[2], G S Rohrer[2], M Groeber[3], M Jackson[4], P J Konijnenberg[5,6] and M De Graef[2,7]

[1] The US Naval Research Laboratory, 4555 Overlook Ave SW, Washington, DC 20375, USA
[2] Department of Materials Science and Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3] Air Force Research Laboratory, Materials & Manufacturing Directorate, Wright Patterson AFB, OH 45433, USA
[4] BlueQuartz Software, 400 S. Pioneer Blvd, Springboro, OH 45066, USA
[5] Max-Planck-Inst. für Eisenforschung, Max-Planck-Str 1, 40237 Düsseldorf, Germany
[6] Bruker-Nano GmbH, Am Studio 2D, 12489, Berlin, Germany

E-mail: David.Rowenhorst@nrl.navy.mil, rollett@andrew.cmu.edu, rohrer@cmu.edu, degraef@cmu.edu, mike.groeber@gmail.com, mike.jackson@bluequartz.net and p.konijnenberg@mpie.de

Received 15 March 2015, revised 12 July 2015
Accepted for publication 19 August 2015
Published 5 October 2015

Abstract

In materials science the orientation of a crystal lattice is described by means of a rotation relative to an external reference frame. A number of rotation representations are in use, including Euler angles, rotation matrices, unit quaternions, Rodrigues–Frank vectors and homochoric vectors. Each representation has distinct advantages and disadvantages with respect to the ease of use for calculations and data visualization. It is therefore convenient to be able to easily convert from one representation to another. However, historically, each representation has been implemented using a set of often tacit conventions; separate research groups would implement different sets of conventions, thereby making the comparison of methods and results difficult and confusing. This tutorial article aims to resolve these ambiguities and provide a consistent set of conventions and conversions between common rotational representations, complete with worked examples and a discussion of the trade-offs necessary to resolve all ambiguities. Additionally, an open source Fortran-90 library of conversion routines for the different representations is made available to the community.

[7] Department of Materials Science and Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213-3980, USA

Keywords: 3D rotation, quaternion, crystallography

(Some figures may appear in colour only in the online journal)

1. Introduction

While the laws of physics are usually expressed in a frame-invariant form, in practice, one must nearly always select one or more reference frames with respect to which physical quantities are expressed. In materials science and engineering, we use a variety of reference frames to describe crystal structures and material properties (the Bravais lattice); orientations of grains in microstructures (typically expressed in the sample reference frame); grid coordinates in scanning-type data acquisitions (e.g. an acquisition grid for electron back-scatter diffraction data sets); and so on. With the exception of crystallographic reference frames, the majority of reference frames used in the materials community are basic Cartesian frames, with orthonormal basis vectors.

The crystallographic reference frames have been defined for all crystal systems by the International Union for Crystallography in the international tables for crystallography (ITC), Volume A [1]. While the conventions of ITC are not always strictly followed (e.g. crystal structures are often described in a non-standard space group setting), on the whole the conventions of ITC-A are followed rather closely by most researchers in the field. When crystallographic quantities are converted to a Cartesian reference frame, however, there are several possible choices for the basis vectors (see section 2.1 for more details) so that one must always be cautious when interpreting crystallographic data that has been transformed to a Cartesian frame; for instance, among the different commercial vendors of electron back-scatter diffraction (EBSD) systems, different reference frame conventions are in use, making it sometimes difficult to compare data for non-orthogonal crystal systems in particular. In this paper, we will restrict our attention to the case of 3D rotations only, and assume that all reference frames have the same origin.

The origin of this paper lies in a series of conversations between the co-authors regarding the details of transformations between various rotation representations. It became clear rather quickly that even in this small group of researchers, several different conventions were being used, leading to different representations of the same rotation and conflicting results. A small-scale 'rotation round-robin' was organized based on the following simple question: *For the Euler angle triplet* $\boldsymbol{\theta} = (\varphi_1, \Phi, \varphi_2) = (2.721\,670, 0.148\,401, 0.148\,886)$ *(radians) in the Bunge convention, what are the equivalent numerical expressions for the axis–angle pair, quaternion, Rodrigues–Frank vector, rotation matrix, and homochoric representations?* All of these rotation representations will be defined in detail later in this paper. The outcome of the round robin was rather surprising: no two authors obtained exactly the same results, with the most common difference being conflicting signs in those rotation representations that involve the rotation axis explicitly. It should also be noted that only two of the authors had implemented all six rotation representations; the others provided results for a subset of the six representations. Given that all co-authors have many years of experience in the field, it was quite surprising to find such a level of disagreement for what was originally planned to be just an initial test to start a larger scale round robin.

While it is possible to perform nearly all texture-related computations using Euler angles, recent years have seen an increased interest in the use of the so-called *neo-Eulerian* representations of the form $\hat{\mathbf{n}}f(\omega)$, where $\hat{\mathbf{n}}$ is a unit vector along the rotation axis and ω the rotation angle. These representations, which will be defined in detail in the next section, offer several advantages over Euler angles, but are perhaps not as broadly understood in the materials community. With the recent availability of open source software for 3-D materials science

Table 1. Frequently used orthonormalizations [5].

$\mathbf{e}_1$	$\mathbf{a}_1/a$	$\mathbf{a}_1^*/a^*$	$\mathbf{a}_1^*/a^*$
$\mathbf{e}_2$	$\mathbf{e}_3 \times \mathbf{e}_1$	$\mathbf{e}_3 \times \mathbf{e}_1$	$\mathbf{a}_2/b$
$\mathbf{e}_3$	$\mathbf{a}_3^*/c^*$	$\mathbf{a}_3/c$	$\mathbf{e}_1 \times \mathbf{e}_2$

Note: The first column is the preferred choice listed in the international tables for crystallography [1].

(e.g. the DREAM.3D package [2], MTEX [3], or the orilib library [4]), which implements many of these alternative representations, it is becoming increasingly important for researchers to be aware of all sign conventions when sharing source code. This tutorial paper is the direct result of a detailed analysis of all orientation representations used in the materials community as well as all possible sources for sign errors or ambiguities. The objective of this paper is to describe methods that can be used to obtain accurate and consistent results when transforming from one reference frame to another, independent of how the rotation is represented.

2. Definitions

In this section, we begin with the definition of the standard Cartesian reference frame. Then we introduce the concept of a 3D rotation and define six different representations or parameterizations commonly used in the materials field. Along the way, we have several opportunities to make a selection between two options (for instance, left-handed versus right-handed, clockwise versus counterclockwise, etc); for each such selection, we will explicitly state our choice as a **convention**. Some conventions will appear to be obvious, but to avoid any uncertainty in later sections of this paper, we will explicitly state even the obvious conventions.

2.1. Reference frames

The Bravais lattice is characterized by basis vectors $\mathbf{a}_i\,(i = 1, \ldots, 3)$, with unit cell edge lengths (a, b, c) and angles (α, β, γ). The reciprocal lattice with basis vectors $\mathbf{a}_j^*$ is then defined by the relation

$$\mathbf{a}_i \cdot \mathbf{a}_j^* = \delta_{ij}, \tag{1}$$

where δ_{ij} is the Kronecker delta; the reciprocal lattice parameters are (a^*, b^*, c^*) and $(\alpha^*, \beta^*, \gamma^*)$. Using these two crystallographic lattices one then defines the standard Cartesian (orthonormal) reference frame by the unit vectors $\mathbf{e}_i$. Although there are infinite possibilities to do so, only a few orthonormalizations, shown in table 1, are frequently used.

These are all right-handed reference frames, i.e. $\mathbf{e}_1 \cdot (\mathbf{e}_2 \times \mathbf{e}_3) = +1$. Note that this convention is not followed in all branches of science; for instance, in computer graphics, left-handed Cartesian frames are quite common [6]. Hence, we formulate the first convention:

Convention 1. *When dealing with 3D rotations, all Cartesian reference frames will be right-handed.*

2.2. 3D rotations

The turning of an object or a reference frame by a given angle ω about a fixed point $\mathbf{d}$ is referred to as a *rotation*. Such an operation preserves the distances between the object points as well as the handedness of the object. Each rotation has an invariant line, represented by the unit vector $\hat{\mathbf{n}}$, which is known as the rotation axis; if the rotation is represented by the operator $\mathcal{R}$, then the action $\mathcal{R}(\hat{\mathbf{n}})$ produces $\hat{\mathbf{n}}$ again.

Consider a rotation characterized by $\mathbf{d}$, $\hat{\mathbf{n}}$, and ω. The rotation angle ω is taken to be positive for a counterclockwise rotation when looking from the end point of $\hat{\mathbf{n}}$ towards the fixed point $\mathbf{d}$. For simplicity, and without loss of generality, we will from here on take the fixed point to coincide with the origin of the reference frame, i.e. $\mathbf{d} = \mathbf{0}$.

Convention 2. *A rotation angle ω is taken to be positive for a counterclockwise rotation when viewing from the end point of the rotation axis unit vector $\hat{\mathbf{n}}$ towards the origin.*

2.3. Rotation matrices and active and passive interpretations

A rotation can be viewed as operating on the object, which is the *active* interpretation, or operating on the reference frame, which is the *passive* interpretation. An active rotation transforms object coordinates to new coordinates in the same reference frame; for the passive interpretation, the initial and final reference frames are different.

If we represent the rotated reference frame by the Cartesian basis vectors $\mathbf{e}'_i$, then the components of these vectors with respect to the original basis $\mathbf{e}_j$ form a transformation matrix α_{ij}, defined in terms of the nine dot products:

$$\alpha_{ij} = \mathbf{e}'_i \cdot \mathbf{e}_j. \tag{2}$$

The matrix α_{ij} defines a passive rotation from the old to the new reference frame, i.e. $\mathbf{e}'_i = \alpha_{ij}\mathbf{e}_j$, where we have used the Einstein summation convention for repeated indices. For a vector $\mathbf{p}$ with components p_j with respect to the basis vectors $\mathbf{e}_j$ and p'_i with respect to $\mathbf{e}'_i$, we have the transformation rule:

$$p'_i = \alpha_{ij}p_j. \tag{3}$$

Rotation matrices belong to the set of special orthonormal matrices, SO(3), and have the following properties: the determinant of the matrix is equal to $+1$; the transpose of the matrix is equal to its inverse; and the sum of the squares of the entries in each column/row equals $+1$.

It is common practice in the materials field, and particularly in the texture community, to always consider 3D rotations as operating on reference frames, i.e. the passive interpretation. We will follow this practice and formulate the third convention as:

Convention 3. *Rotations will be interpreted in the passive sense.*

Therefore, for a crystal whose standard Cartesian reference frame is rotated with respect to the specimen reference frame, the orientation of the crystal will be described as a passive rotation of the sample reference frame to coincide with the crystal's standard reference frame; in other words, the reference frame $\mathbf{e}_i$ is the specimen reference frame, and $\mathbf{e}'_i$ represents the grain's reference frame. The choice of the passive rotation interpretation is a sensible choice, since the grains in a polycrystalline material do not actually rotate; they have a stationary orientation with respect to the specimen reference frame (except, perhaps, during deformation). In other fields of science and engineering, the active rotation interpretation is the more natural one, for instance in the description of the orientation of the segments of a robotic arm; in this case, the segments actually do rotate themselves so that the active interpretation is more suitable.

2.4. Euler angles

Euler has shown that any arbitrary 3D rotation can be decomposed into three successive rotations around the coordinate axes. Since there are three independent axes, denoted *xyz* for simplicity, there are several possibilities for the decomposition. The name *Euler angles* is reserved for those decompositions for which two of the three axis symbols are equal, as in *zxz* or *yzy*. Hence, there are six possible Euler angle conventions. When the three axes are different, one refers to the angle triplet as *Tayt–Briant angles*; while they are used infrequently in materials science, they find frequent application in other fields, for instance in aviation (roll, pitch, and yaw motions of an airplane).

In texture analysis, the most commonly used definition of the Euler angles is the Bunge convention [7], which has an axis triplet *zxz*, with corresponding rotation angles $(\varphi_1, \Phi, \varphi_2)$ (applied from left to right).

Convention 4. *Euler angle triplets* $\boldsymbol{\theta} = (\varphi_1, \Phi, \varphi_2)$ *are implemented using the Bunge convention, with the angular ranges as* $\varphi_1 \in [0, 2\pi]$*,* $\Phi \in [0, \pi]$*, and* $\varphi_2 \in [0, 2\pi]$*.*

Note that, according to convention 3, the Euler angles are interpreted to represent a rotated reference frame. To obtain the active interpretation, one must apply the rotations with opposite angles, in the opposite order, $(-\varphi_2, -\Phi, -\varphi_1)$. The range of the Euler angle triplet defined above results in a total volume in Euler space of $8\pi^2 = \int\int\int \sin\Phi \, \mathrm{d}\varphi_1 \, \mathrm{d}\Phi \, \mathrm{d}\varphi_2$.

2.5. Axis–angle pair

The axis angle pair representation, $(\hat{\mathbf{n}}, \omega)$ provides a convenient way of writing down a rotation in terms of the angle ω and the axis $\hat{\mathbf{n}}$, which is typically represented either by a crystallographic direction $[uvw]$ or by a set of direction cosines $[c_1c_2c_3]$. Note that for non-cubic crystal systems, the normalization must be carried out on direction indices that first have been transformed to the Cartesian reference frame. Often, the axis angle pair is written in the form $\omega\,@\,\hat{\mathbf{n}}$.

Convention 5. *The rotation angle* ω *is limited to the interval* $[0, \pi]$*.*

For angles in the range $]\pi, 2\pi[$, the sign of the unit axis vector $\hat{\mathbf{n}}$ must be reversed, and ω replaced by $2\pi - \omega$. For angles outside the range $[0, 2\pi[$, the angle must first be reduced to the interval $[0, 2\pi[$ by adding or subtracting the appropriate integer multiple of 2π. The main reason for limiting ω to the interval $[0, \pi]$ is to allow for a seamless conversion to the Rodrigues–Frank vector introduced in section 2.7.1.

2.6. Quaternions

A quaternion, q, is defined as a four-component number of the form $q = q_0 + \mathrm{i}q_1 + \mathrm{j}q_2 + \mathrm{k}q_3$, where the imaginary units $(\mathrm{i}, \mathrm{j}, \mathrm{k})$ satisfy the following relations:

$$\begin{aligned} \mathrm{i}^2 &= \mathrm{j}^2 = \mathrm{k}^2 = -1; \quad \mathrm{ij} = -\mathrm{ji} = \mathrm{k}; \\ \mathrm{jk} &= -\mathrm{kj} = \mathrm{i}; \quad \mathrm{ki} = -\mathrm{ik} = \mathrm{j}. \end{aligned} \tag{4}$$

The quaternion is often written as $q = (q_0, \mathbf{q})$ with q_0 the scalar part and $\mathbf{q}$ the vector part; in component notation we have $q = (q_0, q_1, q_2, q_3)$ or $q = (q_0, q_i)$ with $i = 1 \ldots 3$. The definition (4) of the pairwise (ordered) products of the imaginary units leads to the following expression for the quaternion product:

$$
\begin{aligned}
pq &= (p_0 q_0 - \mathbf{p}\cdot\mathbf{q}, q_0\mathbf{p} + p_0\mathbf{q} + \mathbf{p}\times\mathbf{q});\\
&= (p_0 q_0 - p_r q_r, q_0 p_i + p_0 q_i + \epsilon_{ijk} p_j q_k),
\end{aligned}
\tag{5}
$$

where ϵ_{ijk} is the permutation symbol, equal to $+1$ for even and -1 for odd permutations. The presence of a vector cross-product in the vector part of the quaternion product implies that quaternion multiplication does not commute, i.e. $pq \neq qp$.

The norm of a quaternion is defined as:

$$
|q| = [q_0^2 + q_1^2 + q_2^2 + q_3^2]^{\frac{1}{2}}.
\tag{6}
$$

Quaternions with unit norm are known as *unit quaternions* or *versors*, and they have a special relationship with 3D rotations, similar to the relation between regular unit complex numbers and 2D rotations. Unit quaternions can always be written in the form

$$
q = \cos\frac{\omega}{2} + \sin\frac{\omega}{2}(c_1\mathrm{i} + c_2\mathrm{j} + c_3\mathrm{k}),
\tag{7}
$$

where c_i are the direction cosines of the rotation axis unit vector $\hat{\mathbf{n}}$. As a consequence of convention 5, the scalar part $q_0 = \cos(\omega/2)$ of a unit quaternion representing a rotation will always be positive (or 0 for rotations with rotation angle π).

Unit quaternions are located on the sphere $\mathcal{S}^3$ inside the 4D quaternion space. The 3D surface area of this sphere is equal to $2\pi^2$. One can show that the unit quaternions q and $-q$ represent the same rotation so that all rotations can be represented by selecting the entire Northern ($q_0 \geqslant 0$) hemisphere, which has surface area π^2. Pure quaternions have a zero scalar part and they will turn out to be useful to carry out rotations of vectors. Despite the more complicated definition for quaternion multiplication, the quaternion representation of rotations is numerically very convenient and widely used.

It is well known that a regular vector, $\mathbf{r}$, can be rotated by means of quaternion multiplications, provided we convert the vector to a quaternion/versor. We define the quaternion rotation operator $L_p(\,)$, with p a unit quaternion, as follows:

$$
L_p(\mathbf{r}) \equiv \mathrm{vec}[p(0,\mathbf{r})p^*],
\tag{8}
$$

where p^* is the conjugate quaternion $[p_0, -\mathbf{p}]$, and vec[ ] stands for the vector part of the argument. In vector notation we obtain:

$$
L_p(\mathbf{r}) = (p_0^2 - \|\mathbf{p}\|^2)\mathbf{r} + 2(\mathbf{p}\cdot\mathbf{r})\mathbf{p} + 2p_0(\mathbf{p}\times\mathbf{r}).
\tag{9}
$$

The operator $L_p(\mathbf{r})$ rotates the vector from one orientation to another orientation and is hence an active operator. A simple example illustrates the use of the quaternion rotation operator. Consider a rotation of 120° @ [1 1 1]; the corresponding unit quaternion is given by:

$$
p = \left(\cos\frac{\pi}{3}, \frac{1}{\sqrt{3}}[1\,1\,1]\sin\frac{\pi}{3}\right) = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right).
$$

Taking $\mathbf{r} = [1\,0\,0]$ and using the vector expression (9) for the operator $L_p(\,)$, we obtain

$$
L_p([1\,0\,0]) = [0\,1\,0],
$$

as expected for an active rotation. If the conjugate of p is used, then we obtain:

$$
L_{p^*}([1\,0\,0]) = [0\,0\,1],
$$

so that passive rotations can be obtained simply by conjugating the quaternion p. This is equivalent to transposing a rotation matrix to switch between active and passive rotations.

The sign choices in equation (4) have become standard in the mathematical literature and wherever quaternions are used in applications. It is therefore rather interesting that in 1844, Sir Hamilton himself [8] discussed the possibility of a different sign convention for the definition of quaternion multiplication. In section 5 of his 18-installment paper published between 1844 and 1850, Hamilton writes that an entirely consistent alternative definition of the quaternion product can be obtained by changing the order in which the imaginary units are considered from the traditional $(\mathrm{i}, \mathrm{j}, \mathrm{k})$ to $(\mathrm{k}, \mathrm{j}, \mathrm{i})$ so that the defining products become:

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = -1; \quad \mathrm{ji} = -\mathrm{ij} = \mathrm{k}; \quad \mathrm{kj} = -\mathrm{jk} = \mathrm{i}; \quad \mathrm{ik} = -\mathrm{ki} = \mathrm{j}. \tag{10}$$

Note the sign reversals with respect to the standard definition. As a consequence of this alternate definition, the quaternion product becomes:

$$\begin{aligned} pq &= (p_0 q_0 - \mathbf{p} \cdot \mathbf{q}, q_0 \mathbf{p} + p_0 \mathbf{q} - \mathbf{p} \times \mathbf{q}); \\ &= (p_0 q_0 - p_r q_r, q_0 p_i + p_0 q_i - \epsilon_{ijk} p_j q_k), \end{aligned} \tag{11}$$

It is not too difficult to see that the standard definition of the quaternion product, equation (5), corresponds to a right-handed triad $(\mathrm{i}, \mathrm{j}, \mathrm{k})$ of imaginary units, whereas the alternate definition in equation (11) corresponds to a left-handed triad $(\mathrm{k}, \mathrm{j}, \mathrm{i})$. While the right-handed quaternion product has become the standard in modern mathematics, there is no reason not to consider the left-handed product as being on an equal footing. One should note, and this will become important in the following section, that one could also redefine the permutation symbol, ϵ_{ijk}, to be -1 for even permutations and $+1$ for odd permutations; in that case, one could maintain the plus sign in front of the vector cross-product in equation (5) and the sign change caused by the left-handed imaginary triad $(\mathrm{k}, \mathrm{j}, \mathrm{i})$ would be hidden in the permutation symbol. In the remainder of this paper, we prefer to keep the standard definition of the permutation symbol, and we will change the definition of quaternion multiplication as needed by the application, as discussed in section 4.

2.7. Neo-Eulerian representations

The term *neo-Eulerian* was coined by Frank [9] and refers to rotation representations of the form $\hat{\mathbf{n}} f(\omega)$, where f is a suitably chosen monotonic function of the rotation angle. While there are many possible choices for this function, we will only introduce two particular choices: the Rodrigues–Frank vector, and the homochoric vector.

2.7.1. Rodrigues–Frank vector. The Rodrigues–Frank vector, $\boldsymbol{\rho}$, is obtained by setting $f(\omega) = \tan(\omega/2)$:

$$\boldsymbol{\rho} = \hat{\mathbf{n}} \tan \frac{\omega}{2}. \tag{12}$$

Since the rotation angle belongs to the interval $[0, \pi]$, all rotations around an axis $\hat{\mathbf{n}}$ are represented by points along the half-line formed by this unit vector, with a rotation by π represented by the point at infinity $(+\infty)$. Rotation angles in the range $[\pi, 2\pi]$ are equivalent to setting the unit vector equal to its opposite, $-\hat{\mathbf{n}}$, and using a positive angle $2\pi - \omega$. Restricting the rotation angle to $[0, \pi]$ (convention 5) has the added advantage that $\tan(\omega/2)$ is always positive and is thus equal to the length of the Rodrigues-Frank vector $\boldsymbol{\rho}$. Angles outside this range may cause a negative value for the tangent, which would conflict with the positive nature of vector norms.

Given a Rodrigues–Frank vector $\boldsymbol{\rho}$, the corresponding rotation angle can be derived from $\omega = 2\arctan(|\boldsymbol{\rho}|)$, which always results in an angle in the range $[0, \pi]$, in agreement with convention 5. Note that in source code, a Rodrigues–Frank vector should be stored as a four-component vector, first the unit axis vector $\hat{\mathbf{n}}$, then the vector length $\tan(\omega/2)$ in the fourth position. This is necessary so that a 180° rotation around an arbitrary axis can be correctly represented without causing numerical overflows. The ANSI/IEEE Standard 754-1985 (Standard for Binary Floating Point Arithmetic) defines how the number $+\infty$ is represented digitally; proper implementations of the Rodrigues–Frank representation should adhere to this standard.

2.7.2. Homochoric vector. The homochoric representation is defined as a generalization of the standard Lambert equal-area mapping, which projects the 2D sphere $\mathcal{S}^2$ (i.e. the collection of all 3D vectors with unit norm) onto a 2D disk. The generalization takes the 3D unit quaternion hemisphere $\mathcal{S}^3_+$, and maps it onto a 3D ball with a radius selected so that the ball volume is equal to the surface area of the Northern hemisphere of the unit quaternion sphere, π^2.

The homochoric representation of rotations is defined by the following form of the function $f(\omega)$:

$$f(\omega) = \left[\frac{3}{4}(\omega - \sin\omega)\right]^{\frac{1}{3}} \qquad \text{for} \quad \omega \in [0, \pi]. \tag{13}$$

The maximum value of $f(\omega)$ is given by $f(\pi) = (3\pi/4)^{1/3}$, which is the radius of the homochoric ball; hence its volume equals π^2, as required for an equal-volume mapping. Note that for $\omega \in]\pi, 2\pi[$, one would have to use the following definition for the function $f(\omega)$:

$$f(\omega) = \left[\frac{3}{4}(2\pi - \omega + \sin\omega)\right]^{\frac{1}{3}} \qquad \text{for} \quad \omega \in]\pi, 2\pi[.$$

The homochoric vector, $\mathbf{h}$, is defined as:

$$\mathbf{h} = \hat{\mathbf{n}}\left[\frac{3}{4}(\omega - \sin\omega)\right]^{\frac{1}{3}}.$$

Note that the sign limitations on the axis–angle pair are sufficient to guarantee that the above definition is all that is needed. The homochoric representation is an equal-volume representation, which, at least in principle, makes it easy to sample orientation space uniformly. The homochoric ball is isomorphic with the set SO(3).

The homochoric representation can be transformed into the cubochoric representation, via an equal-volume transformation between a ball and a cube [10]. The cubochoric representation allows for a straightforward uniform sampling of SO(3), starting from a uniform 3D cubical grid, which is trivially constructed in numerical implementations. Since the cubochoric representation is essentially a convenient transformation of the homochoric representation, we will not consider it any further in this paper.

3. Relations between rotation parameterizations

3.1. Introductory comments

The conventions listed in the previous section become important when one frequently needs to transform between various rotation representations and parameterizations. If any one of the rotation representations is used by itself, then there is no need to apply any of the conventions, as long as the selected representation is applied in an internally consistent way. To consistently implement transformations between pairs of rotation representations, however, requires the introduction of one additional convention.

As stated in the Introduction, a consistent implementation of conversion routines between rotation representations is made difficult by the fact that one can, in principle, make a sign selection for each individual representation. Consistency between representations can be achieved by a careful consideration of the relations between the representations, which we will describe in detail in the remainder of this section; however, achieving such consistency may come at the cost of having to give up long held understandings or intuitive insights as to how 3D rotations ought to work. At the end of section 2.6, we alluded to the fact that the left-handed quaternion product can be written as a right-handed product, provided the definition of the permutation symbol is changed. It turns out that a proper handling of this sign choice is the key to a consistent implementation of rotation conversions. Instead of changing the definition of the permutation symbol ϵ_{ijk}, we opt to replace the permutation symbol, wherever it occurs, by the following notation:

$$\epsilon_{ijk} \rightarrow P\epsilon_{ijk}; \tag{14}$$

the integer P can take on the values ± 1, and we will see that each choice gives rise to an internally consistent set of rotation conversion relations. One choice, $P = +1$, produces conversion routines that are fully consistent with all standard mathematical definitions and concepts (e.g. the quaternion product) but produces some counter-intuitive results; the opposite choice, $P = -1$, requires us to abandon common mathematical definitions (e.g. the redefinition of the quaternion product discussed in section 2.6), but produces results that agree with our typical understanding of how rotations ought to work. Since both choices for P produce a fully consistent set of conversion algorithms, it is up to the user to select which approach should be followed, either the one that follows traditional mathematical definitions, but results in counter-intuitive conversions, or the one that uses non-standard mathematical definitions, but agrees better with our intuition. As we will see, the selection of P is also closely related to the difference between the active and passive rotation interpretations.

In the following section, we list an overview of the most important analytical conversion relations between rotation representations. In a numerical implementation, such direct conversions are useful, because they can be concatenated to produce conversions between representations for which a direct conversion either does not exist in simple analytical form, or would be cumbersome to implement. In table 2 we list the conversion routines (a checkmark indicates a direct conversion, a letter or letter sequence indicates an indirect conversion via intermediate representations). The cubochoric representation, while not discussed in detail in this paper, is included for completeness.

3.2. Conversions between rotation representations

In this section, we list several analytical conversion formulae between a number of rotation representations. More explicit conversion algorithms are described in Appendix A for all of the checkmarks in table 2. Each conversion is labeled by a string of the form '*xx*2*yy*', where *xx* and *yy* are two-letter abbreviations for the representations: Euler angles (eu); rotation/orientation matrix (om); axis–angle pair (ax); Rodrigues–Frank vector (ro); unit quaternion (qu); and homochoric vector (ho). In all cases, it is assumed that all five conventions introduced before are followed. In addition, we introduce the modified symbol $P\epsilon_{ijk}$ for the permutation symbol wherever it appears. In most cases, the equations listed are consistent with those in reference [11], although the notation may be slightly different. The two routines involving the cubochoric representation are not discussed in the appendix and can be found in [10].

Table 2. Table of direct (✓) and indirect transformation routines between rotation representations, which are identified by single letters: Euler angles (e); rotation/orientation matrix (o); axis–angle pair (a); Rodrigues–Frank vector (r); unit quaternion (q); homochoric vector (h); and cubochoric vector (c).

↓ from\to →	e	o	a	r	q	h	c
e	—	✓	✓	✓	✓	a	ah
o	✓	—	✓	e	✓	a	ah
a	o	✓	—	✓	✓	✓	h
r	o	a	✓	—	a	✓	h
q	✓	✓	✓	✓	—	✓	h
h	ao	a	✓	a	a	—	✓
c	hao	ha	h	ha	ha	✓	—

There are several transformation relations that involve the use of the permutation symbol. The first one is known as the Rodrigues formula and relates the components of the axis–angle pair, $(\hat{\mathbf{n}}, \omega)$, to the orientation matrix α_{ij}:

$$\alpha_{ij} = \delta_{ij} \cos \omega + (1 - \cos \omega) n_i n_j - \sin \omega \; \epsilon_{ijk} n_k. \tag{15}$$

The second relation involves the transformation from quaternion components q_i to the rotation matrix α_{ij}:

$$\alpha_{ij} = (q_0^2 - q_k q_k)\delta_{ij} + 2q_i q_j - 2\epsilon_{ijk} q_0 q_k. \tag{16}$$

In both cases, the permutation symbol must be replaced by the modified symbol so that the transformation relations become:

$$\alpha_{ij} = \delta_{ij} \cos \omega + (1 - \cos \omega) n_i n_j - P \sin \omega \; \epsilon_{ijk} n_k; \tag{17}$$

$$\alpha_{ij} = (q_0^2 - q_k q_k)\delta_{ij} + 2q_i q_j - 2P\epsilon_{ijk} q_0 q_k. \tag{18}$$

4. Discussion

4.1. Revisiting active versus passive interpretations

The 19 conversion relations presented in Appendix A provide efficient numerical pathways to switch from one rotation representation to another. They are fully consistent with each other, regardless of the value of the constant P. The routines have been implemented in an open source fortran-90 library, which is made available to the community via the source code repository `github.com` [12].

In this section, we take a closer look at the active versus passive interpretation, in particular in the context of the conversion routines. While it is clear that the rotation matrix, α_{ij}, as defined in section 2.3, represents a passive rotation (the reference frame is rotated from the old vectors $\mathbf{e}_j$ to the new vectors $\mathbf{e}'_i$), it is not straightforward to recognize the active or passive nature of a rotation once it has been cast into one of the other representations. As an example, consider the Bunge Euler angle triplet $\boldsymbol{\theta} = (\pi/2, 0, 0)$, i.e. a counterclockwise rotation by 90° around the $\mathbf{e}_3$ axis. If we set $P = +1$ in the conversion routines, then we obtain the following representations (the subscript $\boldsymbol{\theta}$ indicates that the representations were derived from the Euler angle triplet using the '*eu*2*yy*' conversion routines of appendix A):

- Passive rotation matrix:

$$\alpha_{\boldsymbol{\theta}} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix};$$

- Quaternion:

$$q_{\boldsymbol{\theta}} = \left(\frac{1}{\sqrt{2}}, 0, 0, -\frac{1}{\sqrt{2}} \right);$$

- Axis–angle pair:

$$(\hat{\mathbf{n}}, \omega)_{\boldsymbol{\theta}} = ([00\bar{1}], 90^\circ);$$

- Rodrigues–Frank vector:

$$\boldsymbol{\rho}_{\boldsymbol{\theta}} = (0, 0, -1);$$

- Homochoric vector:

$$\mathbf{h}_{\boldsymbol{\theta}} = (0, 0, -0.753\,6693).$$

However, if we use our intuition of what a 90° counterclockwise rotation around the $\mathbf{e}_3$ axis should look like in the axis–angle representation, then we would likely write:

$$(\hat{\mathbf{n}}, \omega) = ([0\,0\,1], 90^\circ);$$

note that the rotation axis unit vector here is $[0\,0\,1]$, and not $[00\bar{1}]$ as in the list above. The reason for this discrepancy is that, in determining this representation $(\hat{\mathbf{n}}, \omega) = ([0\,0\,1], 90^\circ)$, we have ignored the fact that the axis–angle pair should represent a passive rotation. All four representations in the list above (quaternion, axis–angle pair, Rodrigues–Frank vector and homochoric vector) must have the same set of plus and/or minus signs in order to be consistent with each other; the sign displayed above may disagree with our intuition, but the sign is consistent with the rotation matrix and Euler angle representations.

The rotation conversions obtained by setting $P = +1$ are mathematically consistent with each other, and they give rise to the expected expressions for the rotation of a vector by means of quaternions (see next section). However, they do not necessarily agree with our expectation or intuition, likely due to the fact that we do not usually think of the distinction between active and passive interpretations in the vector-based rotation representations (quaternion, Rodrigues–Frank, axis–angle, and homochoric). If we ignore the passive nature of the rotation in the vector-based representations, then a conversion to the rotation matrix will result in an active rotation matrix, not a passive one. By selecting the value of P and consistently using this value, we obtain a fully consistent set of rotation transformations; the choice $P = +1$ results in conversions that follow traditional mathematical relations, but produce a counterintuitive result for all vector-based representations, whereas the choice $P = -1$ produces relations that are in agreement with intuition, but require a redefinition of the quaternion product, as we will discuss in detail in the following section.

To obtain a consistent set of transformation routines between rotation representations, we must re-define all representations that depend on the unit rotation axis vector $\hat{\mathbf{n}}$. For a counterclockwise rotation by an angle ω about the $\hat{\mathbf{n}}$ axis, and assuming that conventions 1–5 are satisfied, we have the following definitions for passive rotation representations:

$$\text{axis} - \text{angle pair} : (-P\hat{\mathbf{n}}, \omega); \tag{19}$$

$$\text{Rodrigues} - \text{Frank vector}: \boldsymbol{\rho} = -P\hat{\mathbf{n}}\tan\frac{\omega}{2}; \tag{20}$$

$$\text{homochoric vector}: \mathbf{h} = -P\hat{\mathbf{n}}\left(\frac{3}{4}(\omega - \sin\omega)\right)^{\frac{1}{3}}; \tag{21}$$

$$\text{quaternion}: q = \left(\cos\frac{\omega}{2}, -P\hat{\mathbf{n}}\sin\frac{\omega}{2}\right). \tag{22}$$

In addition to these definitions, we also need to redefine the quaternion multiplication operation:

$$pq \equiv (p_0q_0 - \mathbf{p}\cdot\mathbf{q}, q_0\mathbf{p} + p_0\mathbf{q} + P\mathbf{p}\times\mathbf{q}), \tag{23}$$

As a consequence, the rotation operator $L_p(\mathbf{r})$ must also be modified as follows:

$$L_p(\mathbf{r}) \equiv (p_0^2 - \|\mathbf{p}\|^2)\mathbf{r} + 2(\mathbf{p}\cdot\mathbf{r})\mathbf{p} + 2Pp_0(\mathbf{p}\times\mathbf{r}). \tag{24}$$

The introduction of the factor P in the quaternion rotation operator guarantees that the rotation representations will be fully consistent for all the transformation relations listed in appendix A, regardless of the choice of P. Since rotations are considered to be passive (convention 3), the operator $L_p(\mathbf{r})$ will produce the components of the passively rotated vector $\mathbf{r}$ with respect to the rotated basis vectors; we will illustrate below that this remains true for the concatenation of multiple rotations.

4.2. Vector transformations and quaternion-based consecutive rotations

4.2.1. Illustration of the use of the redefined quaternion product. As an example of the transformation relations and redefined quaternion operations introduced in the previous section, we consider here a counter-clockwise rotation of 120° around the [1 1 1] direction, illustrated in figure 1. In the active interpretation, this rotation takes the vector $\mathbf{r} = (0, 0, z)$ and transforms it to components $(z, 0, 0)$; in the passive interpretation, shown in figure 1(b), the original vector $\mathbf{r}$ now lies along the new $\mathbf{e}'_y$ axis so that the passive components are given by $(0, z, 0)$. We will now show that either choice of P produces the correct results for the transformed coordinates.

We begin by setting $P = -1$. The quaternion q is then given by $\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]$, using equation (22). The resulting rotation matrix, α_{ij} is derived from:

$$\alpha_q = \begin{pmatrix} \bar{q} + 2q_1^2 & 2(q_1q_2 - Pq_0q_3) & 2(q_1q_3 + Pq_0q_2) \\ 2(q_1q_2 + Pq_0q_3) & \bar{q} + 2q_2^2 & 2(q_2q_3 - Pq_0q_1) \\ 2(q_1q_3 - Pq_0q_2) & 2(q_2q_3 + Pq_0q_1) & \bar{q} + 2q_3^2 \end{pmatrix}, \tag{25}$$

with $\bar{q} = q_0^2 - (q_1^2 + q_2^2 + q_3^2)$, which produces the rotation matrix:

$$\alpha_q = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

This is a passive rotation matrix, by convention 3, and operation on the vector components $(0, 0, z)$ produces the components $(0, z, 0)$ with respect to the basis vectors $\mathbf{e}'_i$, as expected. The quaternion operator $L_q(\mathbf{r})$ can be written as:

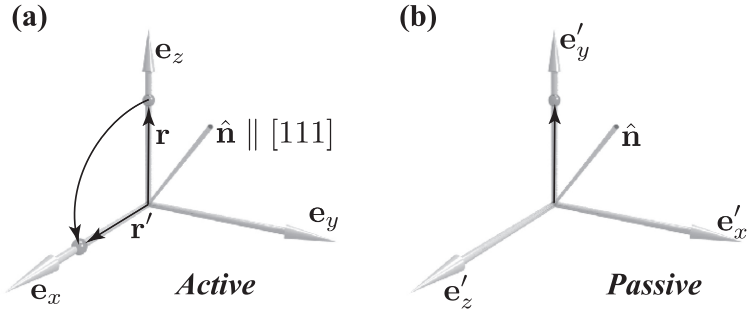

Figure 1. Illustration of active (a) and passive (b) rotations corresponding to 120° @ [1 1 1].

$$L_q(\mathbf{r}) = (q_0^2 - \|\mathbf{q}\|^2)\mathbf{r} + 2(\mathbf{q}\cdot\mathbf{r})\mathbf{q} - 2q_0(\mathbf{q}\times\mathbf{r});$$
$$= -\frac{1}{2}\begin{pmatrix}0\\0\\z\end{pmatrix} + z\begin{pmatrix}1/2\\1/2\\1/2\end{pmatrix} - \begin{pmatrix}z/2\\-z/2\\0\end{pmatrix} = \begin{pmatrix}0\\z\\0\end{pmatrix},$$

which is the correct result for a passive rotation, as shown in figure 1(b).

For the opposite sign choice, $P = +1$, the quaternion becomes $q = \left[\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right]$, following equation (22). The rotation matrix α_q remains the same, due to the fact that the factor P in equation (25) compensates for the sign change. The quaternion rotation operator now becomes:

$$L_q(\mathbf{r}) = (q_0^2 - \|\mathbf{q}\|^2)\mathbf{r} + 2(\mathbf{q}\cdot\mathbf{r})\mathbf{q} + 2q_0(\mathbf{q}\times\mathbf{r});$$
$$= -\frac{1}{2}\begin{pmatrix}0\\0\\z\end{pmatrix} + z\begin{pmatrix}1/2\\1/2\\1/2\end{pmatrix} + \begin{pmatrix}-z/2\\z/2\\0\end{pmatrix} = \begin{pmatrix}0\\z\\0\end{pmatrix},$$

which is once again the correct result. Note that the sign change of the last term with respect to the $P = -1$ case illustrated before is compensated by the sign change in the vector part of the quaternion.

4.2.2. Illustration of rotation combinations. Combinations of rotations are usually described by matrix products, or by quaternion products. We must therefore verify that the modified definitions introduced before remain consistent when two or more rotations are combined. Throughout this section, we will use the following two rotations: rotation A by 120° around [1 1 1], and rotation B by 180° around [1 1 0]; rotation A will be carried out first, followed by rotation B. Figure 2 illustrates the consecutive action of the two rotations, in the active mode (a) and in the passive mode (b). In terms of rotation matrices, we have the following relations:

$$\mathbf{e}''_i = \beta_{ij}\mathbf{e}'_j,$$

and

$$\mathbf{e}'_j = \alpha_{jk}\mathbf{e}_k,$$

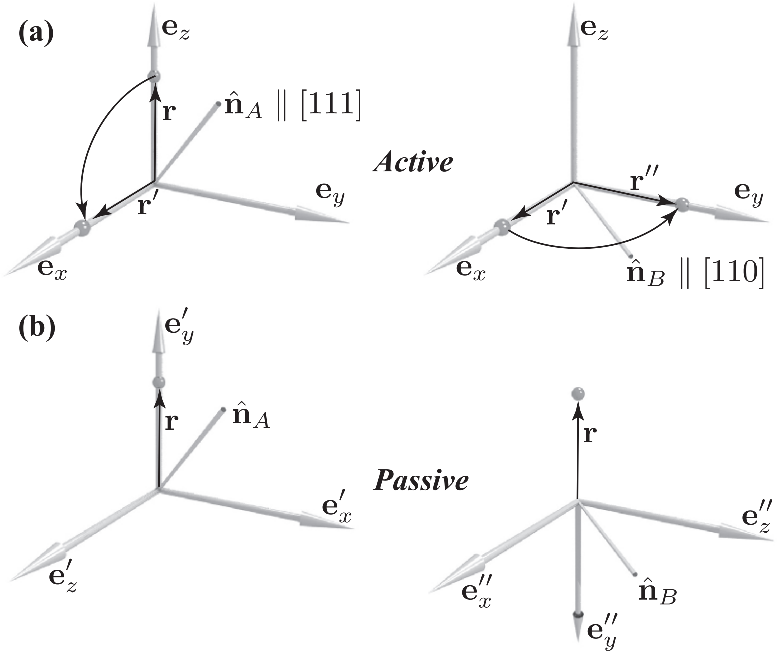

Figure 2. Illustration of the active (a) and passive (b) combination of two rotations.

with α_{jk} representing the passive rotation A, and β_{ij} the passive rotation B. Combining the transformations, keeping in mind the passive nature of the rotation matrices, we have:

$$\mathbf{e}''_i = \alpha_{ij}\beta_{jk}\mathbf{e}_k = \gamma_{ik}\mathbf{e}_k,$$

where the passive rotation matrix $\gamma_{ik} = \alpha_{ij}\beta_{jk}$ represents the combined rotation.

Selecting $P = +1$, the two rotations are represented by the quaternions (using equation (22))

$$q_A = \left[\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\right], \qquad q_B = \left[0, \frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0\right],$$

and the resulting rotation matrices (using equation (25)):

$$\alpha = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \beta = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

The combined rotation is then represented by

$$\gamma = \alpha\beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}.$$

It is easy to verify that this matrix transforms the coordinates of the point (0, 0, z) to (0,$-z$, 0) with respect to the double-primed reference frame. The active matrix γ^T transforms the point (0, 0, z) to (0, z, 0), in agreement with the graphical representation in figure 2(a).

In quaternion form, the combined rotation C can be derived by consecutive application of the quaternion operators $L_{q_A}(\mathbf{r})$ and $L_{q_B}(\mathbf{r})$ in the correct order; since these quaternions, according to convention 3, represent passive rotations, we must combine them in the order q_Aq_B, similar to the order of the passive matrices above. This results in:

$$L_{q_A}(L_{q_B}(\mathbf{r})) = L_{q_A}(q_B(0,\mathbf{r})q_B^*) = q_Aq_B(0,\mathbf{r})q_B^*q_A^* = (q_Aq_B)(0,\mathbf{r})(q_Aq_B)^* = L_{(q_Aq_B)}(\mathbf{r}).$$

For the present case of $P = +1$, and using equation (23), this product q_Aq_B is given by:

$$\begin{aligned} q_C = q_Aq_B &= (q_{A0}q_{B0} - \mathbf{q}_A \cdot \mathbf{q}_B, q_{B0}\mathbf{q}_A + q_{A0}\mathbf{q}_B + \mathbf{q}_A \times \mathbf{q}_B); \\ &= \left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0\right). \end{aligned}$$

According to convention 5, the scalar part of the quaternion must be positive, so we change the sign of all the quaternion components. Using the quaternion rotation operator $L_{q_C}(\mathbf{r})$ (equation (24)) with $\mathbf{r} = (0, 0, z)$ we find:

$$L_{q_C}(\mathbf{r}) = (q_{C0}^2 - \|\mathbf{q}_C\|^2)\mathbf{r} + 2(\mathbf{q}_C \cdot \mathbf{r})\mathbf{q}_C + 2q_{C0}(\mathbf{q}_C \times \mathbf{r}) = (0, -z, 0),$$

in agreement with figure 2(b).

For the case $P = -1$, the quaternions are given by (using equation (22)):

$$q_A = \left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right], \qquad q_B = \left[0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right],$$

and the resulting rotation matrices (using equation (25)) are identical to the matrices listed before. Using equation (23), we have for the quaternion product:

$$q_C = q_Aq_B = \left(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0\right),$$

where convention 5 has been applied. Using the quaternion operator $L_{q_C}(\mathbf{r})$ (equation (24)) with $\mathbf{r} = (0, 0, z)$ we find:

$$L_{q_C}(\mathbf{r}) = (q_{C0}^2 - \|\mathbf{q}_C\|^2)\mathbf{r} + 2(\mathbf{q}_C \cdot \mathbf{r})\mathbf{q}_C - 2q_{C0}(\mathbf{q}_C \times \mathbf{r}) = (0, -z, 0),$$

once again in agreement with figure 2(b). To achieve active rotations, the quaternion q_C must be conjugated before the operator $L_{q_C}(\mathbf{r})$ is applied, i.e. $L_{q_c^*}(\mathbf{r})$.

The conclusion from this explicit example is that the revised definitions of the quaternion operators found in section 4.1 provide a consistent framework for transformations between rotations, as well as for the concatenation of multiple rotations. Either sign choice for P leads to a fully internally consistent set of transformation and composition rules, valid across all the rotation representations described in this paper.

5. Summary

In this tutorial paper, we have described a consistent approach to conversions between different rotation representations and parameterizations commonly used in the materials community. We have stated six conventions (mostly sign conventions) that need to be followed to achieve an internally consistent set of transformation routines; 19 explicit transformation routines between representations are described in the appendix. This paper is the result of a small scale 'rotation round-robin', carried out amongst the authors, which revealed several sign differences between individual author's implementations. The recent availability of open source software for 3-D materials science (e.g. DREAM.3D), for which many researchers are contributing code, has made it clear that there is a need for agreed-upon rotation conventions so that, from the start, source code generated by different research groups can be written in a way that is readily compatible with the larger project code. When dealing with 3-D rotation representations, there are too many opportunities for sign errors and confusion, and it is hoped that the present paper will provide a guide to researchers in the field.

The main message of this tutorial paper is that there are two alternative ways of implementing rotation transformation routines. The two approaches can be selected by means of a simple parameter, P, which takes on the values ±1, and are readily implemented in any modern programming language; a Fortran-90 version, an Interactive Data Language (IDL, [13]) version, and a MatLab version are available from the authors, and a C/C++ version has been implemented as part of the DREAM.3D open source package. Effectively, the introduction of the parameter P as part of a redefined permutation symbol, $\epsilon_{ijk} \rightarrow P\epsilon_{ijk}$, must be carried through in all rotation transformation expressions, as well as in the definition of the quaternion product. All neo-Eulerian rotation representations are also redefined in terms of the parameter P. It is suggested that the selection of P be made at compilation time and that the selection be described explicitly in the package documentation. An extensive test program that exercises all the transformations through a series of general and special rotation cases is also available from the authors; such a test program may prove to be valuable when implementations in other programming languages are undertaken.

Acknowledgments

DJR would like to acknowledge the financial support of the Naval Research Laboratory and the Structural Metallics Program of Office of Naval Research under contract # N0001414WX20779. ADR acknowledges the support of the National Science Foundation under contract # DMR 1435544. MDG would like to acknowledge the Air Force Office of Scientific Research, MURI contract # FA9550-12-1-0458, for financial support.

Appendix A. Conversions between rotation representations

In the following subsections, nineteen explicit rotation conversion algorithms are described. For each conversion, the section title provides a shorthand indicator of the conversion, with the first two letters describing the input representation, and the last two characters the output representation; thus, *eu2ro* refers to the algorithm that converts an Euler angle triplet into a Rodrigues–Frank vector, and *ho2ax* takes a homochoric vector as input and converts it to an axis–angle pair. All algorithms below strictly adhere to the sign conventions described in the main body of this paper. Upon implementing these relations, the reader should exercise caution when using numerical implementations of the arc-tangent function, for which both single argument and two-argument versions exist in many programming languages; it is recommended that the two argument version be used in order for the angular range to be $[0, 2\pi]$, taking care to note which of the arguments is the abscissa (denominator) and which the ordinate (numerator). Care must then be taken to ensure that the resulting angle satisfies convention 5.

Table A1. Coefficients γ_i needed for the conversion from homochoric coordinates to axis angle pair, described in section A.19.

i	γ_{4i+1}	γ_{4i+2}	γ_{4i+3}	γ_{4i+4}
0	1.000 000 000 001 885	−0.500 000 000 219 485	−0.024 999 992 127 593	−0.003 928 701 544 781
1	−0.000 815 270 153 545	−0.000 200 950 042 612	−0.000 023 979 867 761	−0.000 082 028 689 266
2	0.000 124 487 150 421	−0.000 174 911 421 482	0.000 170 348 193 414	−0.000 120 620 650 041
3	0.000 059 719 705 869	−0.000 019 807 567 240	0.000 003 953 714 684	−0.000 000 365 550 014

A.1. eu2om

The Euler angle triplet $\boldsymbol{\theta} = (\varphi_1, \Phi, \varphi_2)$ can be converted into a rotation matrix by multiplication of the individual (passive) rotation matrices:

$$\begin{aligned}\alpha(\boldsymbol{\theta}) &= \alpha_{z''}(\varphi_2)\alpha_{x'}(\Phi)\alpha_z(\varphi_1);\\ &= \begin{pmatrix} c_1c_2 - s_1c\,s_2 & s_1c_2 + c_1c\,s_2 & ss_2 \\ -c_1s_2 - s_1c\,c_2 & -s_1s_2 + c_1c\,c_2 & sc_2 \\ s_1\,s & -c_1\,s & c \end{pmatrix}\end{aligned} \tag{A.1}$$

where $c_i = \cos\varphi_i$, $s_i = \sin\varphi_i$, $c = \cos\Phi$, and $s = \sin\Phi$.

A.2. eu2ax

The axis–angle pair $(\hat{\mathbf{n}}, \omega)$ can be obtained from the Bunge Euler angles by using the following relation:

$$(\hat{\mathbf{n}}, \omega) = \left(-\frac{P}{\tau}t\cos\delta, -\frac{P}{\tau}t\sin\delta, -\frac{P}{\tau}\sin\sigma, \alpha\right) \tag{A.2}$$

where

$$\begin{aligned} t &= \tan\frac{\Phi}{2}; \quad \sigma = \frac{1}{2}(\varphi_1 + \varphi_2); \quad \delta = \frac{1}{2}(\varphi_1 - \varphi_2);\\ \tau &= \sqrt{t^2 + \sin^2\sigma}; \quad \alpha = 2\arctan\frac{\tau}{\cos\sigma}.\end{aligned} \tag{A.3}$$

If $\alpha > \pi$, then the axis angle pair is given by:

$$(\hat{\mathbf{n}}, \omega) = \left(\frac{P}{\tau}t\cos\delta, \frac{P}{\tau}t\sin\delta, \frac{P}{\tau}\sin\sigma, 2\pi - \alpha\right). \tag{A.4}$$

A.3. eu2ro

This conversion involves first the conversion to the axis–angle pair, followed by application of the definition of the Rodrigues–Frank vector in equation (12).

A.4. eu2qu

Starting from Euler angles in radians, we define

$$\sigma = \frac{1}{2}(\varphi_1 + \varphi_2); \quad \delta = \frac{1}{2}(\varphi_1 - \varphi_2); \quad c = \cos\frac{\Phi}{2}; \quad s = \sin\frac{\Phi}{2}; \tag{A.5}$$

The quaternion is then given by:

$$q = (c\cos\sigma, -Ps\cos\delta, -Ps\sin\delta, -Pc\sin\sigma). \tag{A.6}$$

If q_0 becomes negative, then the sign of the entire quaternion has to be reversed, $q \to -q$ so that the resulting quaternion will lie in the Northern hemisphere of the unit quaternion sphere.

A.5. om2eu

Let the rotation matrix be given by α_{ij}, then, if $|\alpha_{33}| \neq 1$, compute ζ as

$$\zeta = \frac{1}{\sqrt{1-\alpha_{33}^2}}. \tag{A.7}$$

Then use the following relations to determine the angles:

$$\boldsymbol{\theta} = (\text{atan2}(\alpha_{31}\zeta, -\alpha_{32}\zeta), \arccos(\alpha_{33}), \text{atan2}(\alpha_{13}\zeta, \alpha_{23}\zeta)). \tag{A.8}$$

where atan2 is the standard two argument implementation of the arc-tangent function, i.e. atan2(y, x) produces the angle for which the tangent equals y/x. Note that the order of the arguments needs to be reversed for some programming languages.

If $|\alpha_{33}| = 1$, i.e. $\Phi = 0$ or π, then we can not determine a unique value for φ_1 and φ_2. We will use the convention that the entire angle is represented by φ_1, and set $\varphi_2 = 0$, and we obtain:

$$\boldsymbol{\theta} = \left(\text{atan2}(\alpha_{12}, \alpha_{11}), \frac{\pi}{2}(1-\alpha_{33}), 0\right). \tag{A.9}$$

A.6. om2ax

Starting from the rotation matrix α_{ij}, we compute the rotation angle via the trace of the matrix:

$$\omega = \arccos\left(\frac{1}{2}(\text{Tr}(\alpha) - 1)\right)$$

If the rotation angle equals zero, the axis–angle pair becomes $(\hat{\mathbf{n}}, \omega) = ([0\,0\,1], 0)$. For the direction cosines of the rotation axis, we determine the eigenvector of the rotation matrix corresponding to the eigenvalue $+1$; it is well known that every rotation matrix has three eigenvalues $(1, \mathrm{e}^{\mathrm{i}\omega}, \mathrm{e}^{-\mathrm{i}\omega})$. For each component of the (right) eigenvector, $\mathbf{E}$, we must then verify the sign as follows:

$$\begin{aligned} E_1 &= \text{sign}(E_1, P(\alpha_{32} - \alpha_{23})) \quad (\alpha_{32} \neq \alpha_{23}); \\ E_2 &= \text{sign}(E_2, P(\alpha_{13} - \alpha_{31})) \quad (\alpha_{13} \neq \alpha_{31}); \\ E_3 &= \text{sign}(E_3, P(\alpha_{21} - \alpha_{12})) \quad (\alpha_{21} \neq \alpha_{12}). \end{aligned} \tag{A.10}$$

The sign function takes two arguments, and returns the first argument with the sign of the second argument.

A.7. om2qu

Consider a rotation matrix α_{ij}. The corresponding quaternion can be determined as follows:

- compute the quaternion components:

$$\begin{aligned} q_0 &= \frac{1}{2}\sqrt{1+\alpha_{11}+\alpha_{22}+\alpha_{33}}; \\ q_1 &= \frac{P}{2}\sqrt{1+\alpha_{11}-\alpha_{22}-\alpha_{33}}; \\ q_2 &= \frac{P}{2}\sqrt{1-\alpha_{11}+\alpha_{22}-\alpha_{33}}; \\ q_3 &= \frac{P}{2}\sqrt{1-\alpha_{11}-\alpha_{22}+\alpha_{33}}. \end{aligned}$$

- Modify the component signs if necessary:

$$\begin{aligned} q_1 = -q_1 &\quad \text{if} \quad \alpha_{32} < \alpha_{23}; \\ q_2 = -q_2 &\quad \text{if} \quad \alpha_{13} < \alpha_{31}; \\ q_3 = -q_3 &\quad \text{if} \quad \alpha_{21} < \alpha_{12}. \end{aligned}$$

- normalize the quaternion:

$$q = \frac{q}{|q|} \tag{A.11}$$

A.8. ax2om

For the axis angle pair $(\hat{\mathbf{n}}, \omega)$, the orientation/rotation matrix for $P = -1$ is given by:

$$\alpha_{ij} = \begin{pmatrix} c + (1-c)n_1^2 & (1-c)n_1n_2 + s\,n_3 & (1-c)n_1n_3 - s\,n_2 \\ (1-c)n_1n_2 - s\,n_3 & c + (1-c)n_2^2 & (1-c)n_2n_3 + s\,n_1 \\ (1-c)n_1n_3 + s\,n_2 & (1-c)n_2n_3 - s\,n_1 & c + (1-c)n_3^2 \end{pmatrix} \tag{A.12}$$

with $c = \cos(\omega)$, $s = \sin(\omega)$. For $P = +1$, the matrix α_{ij} should be transposed.

A.9. ax2ro

For an axis–angle pair $(\hat{\mathbf{n}}, \omega)$, compute

$$\boldsymbol{\rho} = \hat{\mathbf{n}} \tan\frac{\omega}{2}. \tag{A.13}$$

Note that proper care must be taken of the case $\omega = \pi$.

A.10.ax2qu

Given the axis–angle pair $(\hat{\mathbf{n}}, \omega)$, the quaternion is given by:

$$q = \left(\cos\frac{\omega}{2}, \hat{\mathbf{n}}\sin\frac{\omega}{2}\right). \tag{A.14}$$

A.11. ax2ho

Given the axis–angle pair $(\hat{\mathbf{n}}, \omega)$, compute the following parameter:

$$f = \left(\frac{3}{4}(\omega - \sin\omega)\right)^{\frac{1}{3}}; \tag{A.15}$$

the homochoric vector, $\mathbf{h}$, is then given by

$$\mathbf{h} = \hat{\mathbf{n}} f. \tag{A.16}$$

A.12. ro2ax

Consider the Rodrigues–Frank vector $\boldsymbol{\rho}$. First define

$$\rho = |\boldsymbol{\rho}|. \tag{A.17}$$

Then compute the axis–angle pair as:

$$(\hat{\mathbf{n}}, \omega) = \left(\frac{\boldsymbol{\rho}}{\rho}, 2\arctan\rho\right). \tag{A.18}$$

A.13. ro2ho

For a Rodrigues–Frank vector $\boldsymbol{\rho}$, consider the vector length, ρ; if $\rho = 0$ then $\mathbf{h} = (0, 0, 0)$. If $\rho = \infty$, then set $f = 3\pi/4$, otherwise set $f = 3(\omega - \sin\omega)/4$ with $\omega = 2\arctan(\rho)$; then we have

$$\mathbf{h} = \hat{\mathbf{n}} f^{\frac{1}{3}}. \tag{A.19}$$

A.14. qu2eu

For a given unit quaternion q, compute $q_{03} = q_0^2 + q_3^2$, $q_{12} = q_1^2 + q_2^2$, and $\chi = \sqrt{q_{03}q_{12}}$. Distinguish between the following cases:

- If $\chi = 0$ and $q_{12} = 0$, then

$$\boldsymbol{\theta} = (\operatorname{atan2}(-2Pq_0q_3, q_0^2 - q_3^2), 0, 0) \tag{A.20}$$

- If $\chi = 0$ and $q_{03} = 0$, then

$$\boldsymbol{\theta} = (\operatorname{atan2}(2q_1q_2, q_1^2 - q_2^2), \pi, 0) \tag{A.21}$$

- If $\chi \neq 0$, then

$$\begin{aligned}\boldsymbol{\theta} = \Bigg(&\operatorname{atan2}\left(\frac{q_1q_3 - Pq_0q_2}{\chi}, \frac{-Pq_0q_1 - q_2q_3}{\chi}\right), \operatorname{atan2}(2\chi, q_{03} - q_{12}),\\ &\operatorname{atan2}\left(\frac{Pq_0q_2 + q_1q_3}{\chi}, \frac{q_2q_3 - Pq_0q_1}{\chi}\right)\Bigg)\end{aligned} \tag{A.22}$$

Note that all the terms that contain q_0 are multiplied by P.

A.15. qu2om

Consider a unit quaternion q and define $\bar{q} = q_0^2 - (q_1^2 + q_2^2 + q_3^2)$. Then the (passive) rotation matrix α is given by:

$$\alpha_{ij} = \begin{pmatrix} \bar{q} + 2q_1^2 & 2(q_1q_2 - Pq_0q_3) & 2(q_1q_3 + Pq_0q_2) \\ 2(q_1q_2 + Pq_0q_3) & \bar{q} + 2q_2^2 & 2(q_2q_3 - Pq_0q_1) \\ 2(q_1q_3 - Pq_0q_2) & 2(q_2q_3 + Pq_0q_1) & \bar{q} + 2q_3^2 \end{pmatrix}. \tag{A.23}$$

A.16. qu2ax

First compute the rotation angle $\omega = 2\arccos(q_0)$. If $\omega = 0$, then $(\hat{\mathbf{n}}, \omega) = ([0\,0\,1], 0)$. Otherwise:

- if $q_0 = 0$, $(\hat{\mathbf{n}}, \omega) = ([q_1, q_2, q_3], \pi)$.
- if $q_0 \neq 0$, compute

$$s = \frac{\operatorname{sgn}(q_0)}{\sqrt{q_1^2 + q_2^2 + q_3^2}}$$

where sgn returns ±1 according to the sign of its argument; then we have $(\hat{\mathbf{n}}, \omega) = ([sq_1, sq_2, sq_3], \omega)$.

A.17. qu2ro

Set $s = \sqrt{q_1^2 + q_2^2 + q_3^2}$ and $t = \tan(\arccos(q_0))$; then we have:

$$\boldsymbol{\rho} = \left[\frac{q_1}{s}, \frac{q_2}{s}, \frac{q_3}{s}, t\right]. \tag{A.24}$$

Care must be taken when s becomes very small. Recall that the Rodrigues–Frank vector is best stored as a four-component vector, with the fourth element equal to $\tan(\omega/2)$.

A.18. qu2ho

Compute $\omega = 2\arccos(q_0)$; if $\omega = 0$, then $\mathbf{h} = \mathbf{0}$, otherwise compute $s = 1/\sqrt{q_1^2 + q_2^2 + q_3^2}$, set $\hat{\mathbf{n}} = [sq_1, sq_2, sq_3]$, and $f = 3(\omega - \sin\omega)/4$; then the homochoric vector is given by:

$$\mathbf{h} = \hat{\mathbf{n}} f^{\frac{1}{3}}. \tag{A.25}$$

A.19. ho2ax

Define a 16-component constant vector $\boldsymbol{\gamma}$ with components listed in table A1; set $h = |\mathbf{h}|^2$; if $h = 0$, then $(\hat{\mathbf{n}}, \omega) = ([0, 0, 1], 0)$. Otherwise, set $\mathbf{h}' = \mathbf{h}/\sqrt{h}$ and compute the sum:

$$s = \sum_{i=1}^{16} \gamma_i\, h^{i-1};$$

The axis angle pair is then given by $(\hat{\mathbf{n}}, \omega) = (\mathbf{h}', 2\arccos(s))$. This numerical expansion is needed because there is no closed-form analytical expression for the rotation angle in terms of the components of the homochoric vector.

References

[1] Hahn T(ed) 1989 *The International Tables for Crystallography Volume A: Space-Group Symmetry* vol 1 (Dordrecht: Kluwer)

[2] Groeber M A and Jackson M A 2014 DREAM.3D: a digital representation environment for the analysis of microstructure in 3d *Integr. Mater. Manuf. Innov.* **3** 5

[3] http://mtex-toolbox.github.io/

[4] http://orilib.sourceforge.net/

[5] Giacovazzo C, Monaco H L, Artioli G, Viterbo D, Ferraris G, Gilli G, Zanotti G and Catti M 2002 *Fundamentals of Crystallography* (Oxford: Oxford University Press)

[6] Salmon R and Slater M 1987 *Computer Graphics: Systems and Concepts* (London: Addison-Wesley)

[7] Bunge H-J 1982 *Texture Analysis in Materials Science: Mathematical Methods* (London: Butterworths)

[8] Hamilton W R 1844 On quaternions; or on a new system of imaginaries in algebra *London Edinburgh Dublin Phil. Mag.* **25** 10–3

[9] Frank F 1988 Orientation mapping *Metall. Mater. Trans.* A **19** 403–8

[10] Roşca D, Morawiec A and De Graef M 2014 A new method of constructing a grid in the space of 3D rotations and its applications to texture analysis *Model. Simul. Mater. Sci. Eng.* **22** 075013

[11] Morawiec A 2004 *Orientations and Rotations: Computations in Crystallographic Textures* (New York: Springer)

[12] www.github.com/marcdegraef/3drotations

[13] www.exelisvis.com/productsservices/idl.aspx