

py_convrot: rotation conventions, to understand and to apply

Ludmila Urzhumtseva^{a,b*} and Alexandre Urzhumtsev^{c,d*}

^aArchitecture et Réactivité de l'ARN, UPR 9002 CNRS, IBMC (Institute of Molecular and Cellular Biology), 15 rue R. Descartes, 67084 Strasbourg, France, ^bUniversité de Strasbourg, 67000 Strasbourg, France, ^cCentre of Integrative Biology, Institut de Génétique et de Biologie Moléculaire et Cellulaire, CNRS-INSERM-UdS, 1 rue Laurent Fries, BP 10142, 67404 Illkirch, France, and ^dUniversité de Lorraine, Faculté des Sciences et Technologies, BP 239, 54506 Vandoeuvre-les-Nancy, France. *Correspondence e-mail: l.ourjountseva@unistra.fr, sacha@igbmc.fr

Received 15 February 2019

Accepted 20 May 2019

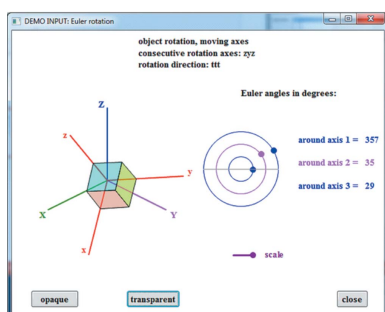
Edited by J. M. García-Ruiz, Instituto Andaluz de Ciencias de la Tierra, Granada, Spain

Keywords: rigid-body rotation; rotation matrices; rotation conventions; teaching tools; interactive illustrations; Euler angles; polar angles; direction cosines.

Rotation is a core crystallographic operation. Two sets of Cartesian coordinates of each point of a rotated object, those before and after rotation, are linearly related, and the coefficients of these linear combinations can be represented in matrix form. This 3×3 matrix is unique for all points and thus describes unambiguously a particular rotation. However, its nine elements are mutually dependent and are not interpretable in a straightforward way. To describe rotations by independent and comprehensible parameters, crystallographic software usually refers to Euler or to polar angles. In crystallography and cryo-electron microscopy, there exists a large choice of conventions, making direct comparison of rotation parameters difficult and sometimes confusing. The program *py_convrot*, written in Python, is a converter of parameters describing rotations. In particular, it deals with all possible choices of polar angles and with all kinds of Euler angles, including all choices of rotation axes and rotation directions. Using a menu, a user can build their own rotation parameterization; its action can be viewed with an interactive graphical tool, *Demo*. The tables in this article and the extended help pages of the program describe details of these parameterizations and the decomposition of rotation matrices into all types of parameters. The program allows orthogonalization conventions and symmetry operations to be taken into account. This makes the program and its supporting materials both an illustrative teaching material, especially for non-specialists in mathematics and computing, and a tool for practical use.

1. Introduction

Rotation in three-dimensional space is one of the principal geometric operations in crystallography and electron microscopy. By definition, rotation is an operation that conserves all distances and has a point fixed, differently from translation. This operation is linear, meaning that for each point its Cartesian coordinates after rotation are linear combinations of its coordinates before rotation. The coefficients of these linear combinations are the same for all points and can be represented in matrix form. Such a rotation matrix describes the corresponding rotation unambiguously. However, the elements of such 3×3 matrices are mutually dependent; not every matrix describes a rotation. Also, interpreting a rotation matrix in terms of motions is not easy, except for trivial cases. For these reasons, in order to manipulate different objects and models and to define their spatial positions, it is convenient to describe rotations in terms of 'physical' parameters, independent if possible. Euler (1776; translation in English by J. Sten, <http://www.17centurymaths.com/contents/euler/e478tr.pdf>) showed that



© 2019 International Union of Crystallography

(1) 'in whatever way a sphere is turned about its centre, it is always possible to assign a diameter, whose direction in the translated state agrees with that of the initial state', and

(2) 'everything [rigid-body rotation] can be defined using three angles'.

In other words, each rotation in three-dimensional space keeps not only a point but a whole line fixed, and each rotation can be realized as a suite of three consecutive elemental rotations about coordinate axes. These two results are at the origin of the rotation descriptions given by so-called polar angles¹ and Euler angles, respectively. The large choice of ways to define the position of the rotation axis (for the former description) and to define the three rotation axes and relevant rotation angles (for the latter description) is the principal reason for frequent misunderstanding and confusion. This variability is not only theoretical; in particular, a very large list of conventions is used in structural biology. The reason for this is that different problems may be better solved using different sets of parameters; there is also the matter of habits in a given research field and its historical development. This large choice requires understanding and conversion of one group of parameters to another.

The literature describing rotation theory is vast and often matches to the features of particular fields and problems. In structural biology, it varies from formal general descriptions, e.g. Diamond & Cranswick (2010), barely accessible to non-mathematicians, to textbooks and fundamental articles describing details of a particular choice of parameters, for example Giacovazzo *et al.* (1992), Heymann *et al.* (2005, 2006) and Rupp (2010). At the same time, comprehensive material explaining the variety of rotation conventions and their relations (e.g. Goldstein, 1950) is practically nonexistent to our knowledge. This leads not only to insufficient understanding of the particular procedures but also to a number of errors in the existing descriptions that sometimes disagree with actual program code.

Previously, a program *convrot* (Urzhumtseva & Urzhumtsev, 1997) was developed to convert one set of rotation parameters to another for a large list of crystallographic conventions known at that time. This program has now become obsolete. Tulloue (2001) has created an excellent illustrative site of rotations defined by Euler angles, but only for a single choice of axes. *EulerAngles* (Bigras & Bonev, 2012) uses all possible sets of Euler angles for rotations about moving coordinate axes and illustrates them. However, it refers neither to rotation conventions in polar angles nor to rotations about fixed coordinate axes. *Pan* (Morawiec, 2015) defines a given rotation simultaneously by quaternions, direction cosines and some particular set of Euler angles. The program *3D Rotation Converter* (Gaschler, 2016) is more general but also does not address some major rotation conventions. The site <https://www.geometrictools.com/Documentation/EulerAngles.pdf> by D. Eberly gives rotation

matrices for different sets of Euler angles and expressions for the respective angles.

In summary, these programs and sites cover a limited range of conventions while missing some used in structural biology. Only a few of them are illustrative; practically none of them deal with symmetries and orthogonalization conventions that are important in crystallographic projects.

To address these problems, we have developed a new rotation converter, *py_convrot*, and describe it herein, with the following considerations:

(a) The program should not simply convert one set of numbers into another but be capable of graphically illustrating what the respective numbers mean; it should have an interactive illustrator of different rotation conventions.

(b) The program needs to be in an open form and not to only refer to conventions used nowadays.

(c) The program and the complementary material should be self-contained, so that users can reproduce all calculations without requiring external information.

(d) The material should be accessible to readers without advanced training in mathematics and informatics.

(e) The description of parameters should be non-confusing, especially in cases of contradicting notations and comments in existing literature and software descriptions. The main text should contain only the principal relations while appendices give all minimally necessary technical details and demonstrations.

(f) The software should be as compatible as possible with widely used crystallographic packages and program suites, such as *CCP4* (Collaborative Computational Project, Number 4, 1994), *PyMol* (DeLano, 2002; <http://www.pymol.org>), *Phenix* (Adams *et al.*, 2010), *Coot* (Emsley *et al.*, 2010) and others, and be principally aimed at projects in crystallography and electron microscopy.

2. Converter purpose

2.1. Conversion of rotation parameters

The converter addresses the problem that the rotation of an object (a rigid body) in three-dimensional space can be described by different types of parameters. This variety of descriptions is large since

(a) one can describe a rotation as a single operation about an axis in a general orientation (polar angles) or by three consecutive rotations about coordinate axes (Euler angles),

(b) both polar and Euler angles may be defined with respect to different coordinate axes,

(c) Euler angles may be defined with respect to both fixed and moving axes,

(d) a positive direction of each rotation may be defined in two opposite ways, and

(e) some conventions consider the object rotating with respect to a fixed coordinate system, while some do it inversely.

Note that (c) and (e) are different. In (c) there are two different choices of the rotation axes, but the object is always

¹ The term 'polar angles', widely used in crystallography and especially in macromolecular crystallography, sometimes is considered as formally inappropriate; see discussion in §3.3 and in Appendix C.

moved and the system is kept fixed, while in (e) there are two different choices for the fixed and rotating items.

The program *py_convrot* considers all possible choices listed above. For each of these conventions, the program first calculates the rotation matrix corresponding to the given values of parameters. At the next step this matrix is interpreted in terms of parameters corresponding to any other convention, according to the user's choice. If requested, this matrix can be applied to a list of atomic coordinates in order to get a list of respective coordinates after rotation (rotated atomic model). The program can operate either with a single set of parameter values or with a list of sets.

The program also considers the rotation description by the direction cosines defining the orientation of the rotation axis and that by quaternions. Differently from *convrot*, it no longer considers the Lattman (1972) parameterization of the Euler angles used in *CNS* (Brünger *et al.*, 1998).

Specifically for crystallographic projects, orthogonalization conventions may be taken into account when defining the rotations. If crystallographic symmetry operations (space group) are known and selected from the menu, the program gives a full list of symmetry-related angles after rotation; this option may be useful for molecular replacement to compare results of several rotation functions (Urzhumtsev & Urzhumtseva, 2002; Urzhumtseva & Urzhumtsev, 2002).

2.2. Teaching goals

Teaching is another principal purpose of the program.

Demo is an interactive graphical tool especially useful for understanding and comparing different rotation conventions. For each choice of rotation parameters, *Demo* shows how the position of an object changes with a variation of values of the chosen parameters. A user can choose their own combination of rotation parameters, creating and trying new conventions.

Help gives principal formulae for rotation conventions and illustrates them one by one. *Help* pages are available in pdf format; they are provided together with the program and can be used separately.

Section 3 and Appendices A–D provide reminders of principal definitions, basic notions and results concerning rotations. Tables 1–3 summarize basic information relevant to the choice of the rotation matrices and their parameters when

Table 1

Rotation matrices for the direction cosines and for quaternions and their values extracted from the matrix.

For $\kappa = \pi$ or $q_0 = 0$ the component without the sign function is usually that with the largest absolute value among the diagonal elements, and the other two components are defined using sign. The table gives an example when l_z is the largest component. $c\kappa$ and $s\kappa$ stand for the cosine and sine of the rotation angle κ . For other definitions see Section 3.3.1.

Matrix	Matrix solution
	$\kappa = \arccos[\frac{1}{2}(r_{11} + r_{22} + r_{33} - 1)]$
	$\kappa = 0: l_x = l_y = 0, l_z = 1$
$\begin{bmatrix} l_x^2(1 - c\kappa) + c\kappa & l_x l_y(1 - c\kappa) - l_z s\kappa & l_x l_z(1 - c\kappa) + l_y s\kappa \\ l_x l_y(1 - c\kappa) + l_z s\kappa & l_y^2(1 - c\kappa) + c\kappa & l_y l_z(1 - c\kappa) - l_x s\kappa \\ l_x l_z(1 - c\kappa) - l_y s\kappa & l_y l_z(1 - c\kappa) + l_x s\kappa & l_z^2(1 - c\kappa) + c\kappa \end{bmatrix}$	$0 < \kappa < \pi:$ $l_x = [(r_{32} - r_{23})/(2s\kappa)]$ $l_y = [(r_{13} - r_{31})/(2s\kappa)]$ $l_z = [(r_{21} - r_{12})/(2s\kappa)]$
	$\kappa = \pi:$ $l_x = \text{sign}(r_{13}) \cos[\frac{1}{2} \arccos(r_{11})]$ $l_y = \text{sign}(r_{23}) \cos[\frac{1}{2} \arccos(r_{22})]$ $l_z = \cos[\frac{1}{2} \arccos(r_{33})]$
	$q_0 = \frac{1}{2}[(r_{11} + r_{22} + r_{33} + 1)]^{1/2}$
$\begin{bmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_z q_0) & 2(q_x q_z + q_y q_0) \\ 2(q_x q_y + q_z q_0) & q_0^2 + q_y^2 - q_x^2 - q_z^2 & 2(q_y q_z - q_x q_0) \\ 2(q_x q_z - q_y q_0) & 2(q_y q_z + q_x q_0) & q_0^2 + q_z^2 - q_x^2 - q_y^2 \end{bmatrix}$	$q_0 > 0:$ $q_x = [(r_{32} - r_{23})/(4q_0)]$ $q_y = [(r_{13} - r_{31})/(4q_0)]$ $q_z = [(r_{21} - r_{12})/(4q_0)]$
	$q_0 = 0:$ $q_x = \text{sign}(r_{13}) \cos[\frac{1}{2} \arccos(r_{11})]$ $q_y = \text{sign}(r_{23}) \cos[\frac{1}{2} \arccos(r_{22})]$ $q_z = \cos[\frac{1}{2} \arccos(r_{33})]$

using direction cosines and quaternions (Table 1) or Euler (Table 2) or polar angles (Table 3).

All of this information is classic and most of it can be found in the literature. However it is much dispersed between different sources and the notations used are very different, sometimes opposite to each other and confusing. For this reason we have tried to produce a single document presenting this variety of options in a consistent, complete and detailed way.

3. Method

3.1. Two basic options to describe rotations

Let an object be defined in some Cartesian coordinate system with unit and mutually orthogonal basis vectors \mathbf{e}_x , \mathbf{e}_y , \mathbf{e}_z (orthonormal basis), making a right-hand frame (called also right-handed or right-chiral), and with the origin at \mathbf{O} , the point around which the object is rotated. In crystallography, the default choice of the basis vectors, also referred to as

Table 2

Euler angles: rotation matrices and angles extracted from the matrices.

c1 and s1 stand for $\cos(\kappa_1)$ and $\sin(\kappa_1)$; similarly for c2, s2, c3 and s3. The function $\text{atan2}(y, x)$ defines the angle κ such that $\tan \kappa = y/x$ and that the signs of $\sin(\kappa)$ and $\cos(\kappa)$ coincide with the signs of y and x , respectively.

Matrix product	Resulting rotation matrix	Matrix solution
$R_x(\kappa_3)R_y(\kappa_2)R_x(\kappa_1)$	$\begin{pmatrix} c2 & s1 s2 & c1 s2 \\ s2 s3 & -s1 c2 s3 + c1 c3 & -c1 c2 s3 - s1 c3 \\ -s2 c3 & s1 c2 c3 + c1 s3 & c1 c2 c3 - s1 s3 \end{pmatrix}$	$\kappa_2 = \arccos(r_{11})$ s2 = 0: $\kappa_1 = \text{atan2}(-r_{23}, r_{22}), \kappa_3 = 0$ s2 > 0: $\kappa_1 = \text{atan2}(r_{12}, r_{13}), \kappa_3 = \text{atan2}(r_{21}, -r_{31})$
$R_y(\kappa_3)R_z(\kappa_2)R_y(\kappa_1)$	$\begin{pmatrix} c1 c2 c3 - s1 s3 & -s2 c3 & s1 c2 c3 + c1 s3 \\ c1 s2 & c2 & s1 s2 \\ -c1 c2 s3 - s1 c3 & s2 s3 & -s1 c2 s3 + c1 c3 \end{pmatrix}$	$\kappa_2 = \arccos(r_{22})$ s2 = 0: $\kappa_1 = \text{atan2}(-r_{31}, r_{33}), \kappa_3 = 0$ s2 > 0: $\kappa_1 = \text{atan2}(r_{23}, r_{21}), \kappa_3 = \text{atan2}(r_{32}, -r_{12})$
$R_z(\kappa_3)R_x(\kappa_2)R_z(\kappa_1)$	$\begin{pmatrix} -s1 c2 s3 + c1 c3 & -c1 c2 s3 - s1 c3 & s2 s3 \\ s1 c2 c3 + c1 s3 & c1 c2 c3 - s1 s3 & -s2 c3 \\ s1 s2 & c1 s2 & c2 \end{pmatrix}$	$\kappa_2 = \arccos(r_{33})$ s2 = 0: $\kappa_1 = \text{atan2}(-r_{12}, r_{11}), \kappa_3 = 0$ s2 > 0: $\kappa_1 = \text{atan2}(r_{31}, r_{32}), \kappa_3 = \text{atan2}(r_{13}, -r_{23})$
$R_x(\kappa_3)R_z(\kappa_2)R_x(\kappa_1)$	$\begin{pmatrix} c2 & -c1 s2 & s1 s2 \\ s2 c3 & c1 c2 c3 - s1 s3 & -s1 c2 c3 - c1 s3 \\ s2 s3 & c1 c2 s3 + s1 c3 & -s1 c2 s3 + c1 c3 \end{pmatrix}$	$\kappa_2 = \arccos(r_{11})$ s2 = 0: $\kappa_1 = \text{atan2}(r_{32}, r_{33}), \kappa_3 = 0$ s2 > 0: $\kappa_1 = \text{atan2}(r_{13}, -r_{12}), \kappa_3 = \text{atan2}(r_{31}, r_{21})$
$R_y(\kappa_3)R_x(\kappa_2)R_y(\kappa_1)$	$\begin{pmatrix} -s1 c2 s3 + c1 c3 & s2 s3 & c1 c2 s3 + s1 c3 \\ s1 s2 & c2 & -c1 s2 \\ -s1 c2 c3 - c1 s3 & s2 c3 & c1 c2 c3 - s1 s3 \end{pmatrix}$	$\kappa_2 = \arccos(r_{22})$ s2 = 0: $\kappa_1 = \text{atan2}(r_{13}, r_{11}), \kappa_3 = 0$ s2 > 0: $\kappa_1 = \text{atan2}(r_{21}, -r_{23}), \kappa_3 = \text{atan2}(r_{12}, r_{32})$
$R_z(\kappa_3)R_y(\kappa_2)R_z(\kappa_1)$	$\begin{pmatrix} c1 c2 c3 - s1 s3 & -s1 c2 c3 - c1 s3 & s2 c3 \\ c1 c2 s3 + s1 c3 & -s1 c2 s3 + c1 c3 & s2 s3 \\ -c1 s2 & s1 s2 & c2 \end{pmatrix}$	$\kappa_2 = \arccos(r_{33})$ s2 = 0: $\kappa_1 = \text{atan2}(r_{21}, r_{22}), \kappa_3 = 0$ s2 > 0: $\kappa_1 = \text{atan2}(r_{32}, -r_{31}), \kappa_3 = \text{atan2}(r_{23}, r_{13})$
$R_z(\kappa_3)R_y(\kappa_2)R_x(\kappa_1)$	$\begin{pmatrix} c2 c3 & s1 s2 c3 - c1 s3 & c1 s2 c3 + s1 s3 \\ c2 s3 & s1 s2 s3 + c1 c3 & c1 s2 s3 - s1 c3 \\ -s2 & s1 c2 & c1 c2 \end{pmatrix}$	$\kappa_2 = -\arcsin(r_{31})$ c2 = 0: $\kappa_1 = \text{atan2}(-r_{23}, r_{22}), \kappa_3 = 0$ c2 > 0: $\kappa_1 = \text{atan2}(r_{32}, r_{33}), \kappa_3 = \text{atan2}(r_{21}, r_{11})$
$R_x(\kappa_3)R_z(\kappa_2)R_y(\kappa_1)$	$\begin{pmatrix} c1 c2 & -s2 & s1 c2 \\ c1 s2 c3 + s1 s3 & c2 c3 & s1 s2 c3 - c1 s3 \\ c1 s2 s3 - s1 c3 & c2 s3 & s1 s2 s3 + c1 c3 \end{pmatrix}$	$\kappa_2 = -\arcsin(r_{12})$ c2 = 0: $\kappa_1 = \text{atan2}(-r_{31}, r_{33}), \kappa_3 = 0$ c2 > 0: $\kappa_1 = \text{atan2}(r_{13}, r_{11}), \kappa_3 = \text{atan2}(r_{32}, r_{22})$
$R_y(\kappa_3)R_x(\kappa_2)R_z(\kappa_1)$	$\begin{pmatrix} s1 s2 s3 + c1 c3 & c1 s2 s3 - s1 c3 & c2 s3 \\ s1 c2 & c1 c2 & -s2 \\ s1 s2 c3 - c1 s3 & c1 s2 c3 + s1 s3 & c2 c3 \end{pmatrix}$	$\kappa_2 = -\arcsin(r_{23})$ c2 = 0: $\kappa_1 = \text{atan2}(-r_{12}, r_{11}), \kappa_3 = 0$ c2 > 0: $\kappa_1 = \text{atan2}(r_{21}, r_{22}), \kappa_3 = \text{atan2}(r_{13}, r_{33})$
$R_y(\kappa_3)R_z(\kappa_2)R_x(\kappa_1)$	$\begin{pmatrix} c2 c3 & -c1 s2 c3 + s1 s3 & s1 s2 c3 + c1 s3 \\ s2 & c1 c2 & -s1 c2 \\ -c2 s3 & c1 s2 s3 + s1 c3 & -s1 s2 s3 + c1 c3 \end{pmatrix}$	$\kappa_2 = \arcsin(r_{21})$ c2 = 0: $\kappa_1 = \text{atan2}(r_{32}, r_{33}), \kappa_3 = 0$ c2 > 0: $\kappa_1 = \text{atan2}(-r_{23}, r_{22}), \kappa_3 = \text{atan2}(-r_{31}, r_{11})$
$R_z(\kappa_3)R_x(\kappa_2)R_y(\kappa_1)$	$\begin{pmatrix} -s1 s2 s3 + c1 c3 & -c2 s3 & c1 s2 s3 + s1 c3 \\ s1 s2 c3 + c1 s3 & c2 c3 & -c1 s2 c3 + s1 s3 \\ -s1 c2 & s2 & c1 c2 \end{pmatrix}$	$\kappa_2 = \arcsin(r_{32})$ c2 = 0: $\kappa_1 = \text{atan2}(r_{13}, r_{11}), \kappa_3 = 0$ c2 > 0: $\kappa_1 = \text{atan2}(-r_{31}, r_{33}), \kappa_3 = \text{atan2}(-r_{12}, r_{22})$
$R_x(\kappa_3)R_y(\kappa_2)R_z(\kappa_1)$	$\begin{pmatrix} c1 c2 & -s1 c2 & s2 \\ c1 s2 s3 + s1 c3 & -s1 s2 s3 + c1 c3 & -c2 s3 \\ -c1 s2 c3 + s1 s3 & s1 s2 c3 + c1 s3 & c2 c3 \end{pmatrix}$	$\kappa_2 = \arcsin(r_{13})$ c2 = 0: $\kappa_1 = \text{atan2}(r_{21}, r_{22}), \kappa_3 = 0$ c2 > 0: $\kappa_1 = \text{atan2}(-r_{12}, r_{11}), \kappa_3 = \text{atan2}(-r_{23}, r_{33})$

coordinate axes, is defined by one of the orthogonalization agreements (Appendix D).

Euler (1776) showed that any rotation, *i.e.* a motion with one immobile point, **O** in our case, may be represented as a single rotation about some axis in a general position. Alternatively, it can be represented by rotations about particular coordinate axes, but this requires three such consecutive elemental rotations. The Cartesian coordinates of a point after rotation are expressed through its coordinates before rotation (both described in the same basis **{O, e_x, e_y, e_z}**) by three linear equations that can be represented in matrix form. Note that this rotation matrix *R* is the same for all points of the object and is unique for each rotation.

3.2. Three consecutive rotations about the coordinate axes

3.2.1. Euler angles and rotation axes. Let us consider three consecutive rotations about coordinate axes **axis1**, **axis2**, **axis3** by angles κ_1 , κ_2 , κ_3 , respectively; these angles are known as Euler angles. The order in which the rotations are applied is crucial. While in the literature one can find alternative notations for these angles (α , β , γ or ϕ , θ , ψ or other), our notation has the advantage of showing explicitly the order of rotations: first by κ_1 about **axis1**, then by κ_2 about **axis2**, then by κ_3 about **axis3**, whatever the axes are. Let us denote the respective rotation matrices by $R_{\text{axis1}}(\kappa_1)$, $R_{\text{axis2}}(\kappa_2)$ and $R_{\text{axis3}}(\kappa_3)$; particular expressions of matrices for rotations about **e_x**, **e_y**, **e_z** are given in Appendix B.

If the three rotation axes are selected from the set **e_x**, **e_y**, **e_z** we are talking about ‘extrinsic rotations’ or ‘active rotations’; for simplicity we denote the corresponding choice by **x**, **y** or **z**, respectively. Two consecutive rotations cannot be about the same axis, but the first and the third axes can be the same (referred to as proper Euler angles; when all three axes are different, the corresponding angles are also called Tait–Bryan angles). For example, **xyz** and **zyz** are acceptable, but not **yyz**. The overall rotation matrix is a product of matrices *from right to left* (Appendix A):

$$R = R_{\text{axis3}}(\kappa_3) R_{\text{axis2}}(\kappa_2) R_{\text{axis1}}(\kappa_1). \quad (1)$$

For example, for **axis1 = x**, **axis2 = y**, **axis3 = z** the matrix (1) is $R_z(\kappa_3)R_y(\kappa_2)R_x(\kappa_1)$.

Often **axis1**, **axis2**, **axis3** correspond to a different, mobile coordinate system. Initially its basis vectors **u_x**, **u_y**, **u_z** coincide with the respective vectors **e_x**, **e_y**, **e_z**. However, they are associated with the rotating object and move with it. In this case we are talking about ‘intrinsic rotations’ or ‘passive rotations’. Referring to the same three basic matrices as above, the rotation matrix becomes their product *from left to right* (Appendix B):

$$R = R_{\text{axis1}}(\kappa_1) R_{\text{axis2}}(\kappa_2) R_{\text{axis3}}(\kappa_3). \quad (2)$$

3.2.2. Direction of the rotations. The elemental matrices $R_x(\kappa)$, $R_y(\kappa)$, $R_z(\kappa)$ given in Appendix B correspond to rotations in the trigonometrical sense, *i.e.* counterclockwise when looking against the rotation axis [Fig. 1(a)]. Working in three-

dimensional space, the term ‘right-hand rotation’ is frequently used. While this is the default direction in mathematics and physics, one may define a positive rotation direction in the opposite sense, *i.e.* clockwise when looking against the rotation axis (left-hand rotation). This does not change the form of the rotation matrix but simply reverses the sign of its argument. (This is equivalent to inversion of the corresponding matrix.) For example, inversion of the rotation direction for all three extrinsic rotations [equation (1)] simultaneously makes matrix *R* equal to

$$R = R_{\text{axis3}}(-\kappa_3) R_{\text{axis2}}(-\kappa_2) R_{\text{axis1}}(-\kappa_1). \quad (3)$$

For intrinsic rotations [equation (2)] it becomes

$$R = R_{\text{axis1}}(-\kappa_1) R_{\text{axis2}}(-\kappa_2) R_{\text{axis3}}(-\kappa_3). \quad (4)$$

3.2.3. System or object rotation. Owing to the relative character of a motion, rotation of an object with respect to a given basis may also be seen as a rotation of this basis keeping the object fixed. Such rotation automatically means that the coordinate axes (of the system) are moving ones.

In this case, three consecutive rotations of the basis about its axes **axis1**, **axis2**, **axis3** by angles κ_1 , κ_2 , κ_3 result in the rotation matrix (Appendix B)

$$R = R_{\text{axis3}}(-\kappa_3) R_{\text{axis2}}(-\kappa_2) R_{\text{axis1}}(-\kappa_1), \quad (5)$$

coinciding with expression (3) above.

3.3. Single rotation about an axis in a general orientation

3.3.1. Direction cosines and quaternions. Now let us consider a single rotation of an object by some angle κ about an axis in a general orientation. This rotation axis can be defined by a unit vector **l** along it. Vector **l** can be represented by its Cartesian coordinates (l_x , l_y , l_z) in the same orthonormal basis **{O, e_x, e_y, e_z}** as above. These coordinates are cosines of the angles that **l** makes with the coordinate axes **e_x**, **e_y**, **e_z** [Fig. 1(b)]. Table 1 shows the corresponding rotation matrix.

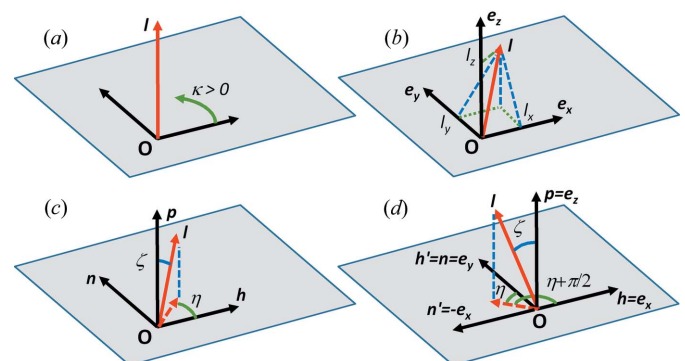


Figure 1 Rotation axis **l** (in red). (a) Definition of the positive (trigonometric, right hand) direction of rotation. (b) Definition of the rotation axes by direction cosines. (c) Definition of the rotation axis by polar coordinates. (d) Example of a standard (**h = e_x**, **n = e_y**) and alternative (**h' = e_y**, **n' = -e_x**) choice for the azimuth axis.

Table 3

Polar angles: rotation matrices and angles extracted from the matrices.

ζ is the zenith angle. $c\zeta$ and $s\zeta$ stand for $\cos(\zeta)$ and $\sin(\zeta)$; similarly for the azimuth angle η and rotation angle κ . $\text{atan2}(y, x)$ is defined as for Table 2.

Rotation matrix product	Resulting rotation matrix	Matrix solution
Zenith: Z Azimuth: X $R_z(\eta)R_y(\zeta)R_z(\kappa)R_y(-\zeta)R_z(-\eta)$	$\begin{bmatrix} (s\zeta)^2(c\eta)^2(1-c\kappa) + c\kappa & (s\zeta)^2s\eta c\eta(1-c\kappa) - c\zeta s\kappa & s\zeta c\zeta c\eta(1-c\kappa) + s\zeta s\eta s\kappa \\ (s\zeta)^2s\eta c\eta(1-c\kappa) + c\zeta s\kappa & (s\zeta)^2(s\eta)^2(1-c\kappa) + c\kappa & s\zeta c\zeta s\eta(1-c\kappa) - s\zeta c\eta s\kappa \\ s\zeta c\zeta c\eta(1-c\kappa) - s\zeta s\eta s\kappa & s\zeta c\zeta s\eta(1-c\kappa) + s\zeta c\eta s\kappa & (c\zeta)^2(1-c\kappa) + c\kappa \end{bmatrix}$	$\kappa = \arccos[(r_{11} + r_{22} + r_{33} - 1)/2]$ $\kappa = 0: \zeta = \eta = 0$ $0 < \kappa < \pi: \zeta = \arccos[(r_{21} - r_{12})/(2s\kappa)]$ $\zeta = 0$ or $\zeta = \pi: \eta = 0$ $0 < \zeta < \pi: \eta = \text{atan2}(r_{13} - r_{31}, r_{32} - r_{23})$ $\kappa = \pi: \zeta = \frac{1}{2}\arccos(r_{33})$ $\zeta = 0: \eta = 0$ $\zeta = \pi/2: \eta = \frac{1}{2}\text{atan2}(r_{12} + r_{21}, 2r_{11})$ $0 < \zeta < \pi/2: \eta = \text{atan2}(r_{23} + r_{32}, r_{13} + r_{31})$
Zenith: X Azimuth: Y $R_x(\eta)R_z(\zeta)R_x(\kappa)R_z(-\zeta)R_x(-\eta)$	$\begin{bmatrix} (c\zeta)^2(1-c\kappa) + c\kappa & s\zeta c\zeta c\eta(1-c\kappa) - s\zeta s\eta s\kappa & s\zeta c\zeta s\eta(1-c\kappa) + s\zeta c\eta s\kappa \\ s\zeta c\zeta c\eta(1-c\kappa) + s\zeta s\eta s\kappa & (s\zeta)^2(c\eta)^2(1-c\kappa) + c\kappa & (s\zeta)^2s\eta c\eta(1-c\kappa) - c\zeta s\kappa \\ s\zeta c\zeta s\eta(1-c\kappa) - s\zeta c\eta s\kappa & (s\zeta)^2s\eta c\eta(1-c\kappa) + c\zeta s\kappa & (s\zeta)^2(s\eta)^2(1-c\kappa) + c\kappa \end{bmatrix}$	$\kappa = \arccos[(r_{11} + r_{22} + r_{33} - 1)/2]$ $\kappa = 0: \zeta = \eta = 0$ $0 < \kappa < \pi: \zeta = \arccos[(r_{32} - r_{23})/(2s\kappa)]$ $\zeta = 0$ or $\zeta = \pi: \eta = 0$ $0 < \zeta < \pi: \eta = \text{atan2}(r_{21} - r_{12}, r_{13} - r_{31})$ $\kappa = \pi: \zeta = \frac{1}{2}\arccos(r_{11})$ $\zeta = 0: \eta = 0$ $\zeta = \pi/2: \eta = \frac{1}{2}\text{atan2}(r_{23} + r_{32}, 2r_{22})$ $0 < \zeta < \pi/2: \eta = \text{atan2}(r_{31} + r_{13}, r_{21} + r_{12})$
Zenith: Y Azimuth: Z $R_y(\eta)R_x(\zeta)R_y(\kappa)R_x(-\zeta)R_y(-\eta)$	$\begin{bmatrix} (s\zeta)^2(s\eta)^2(1-c\kappa) + c\kappa & s\zeta c\zeta s\eta(1-c\kappa) - s\zeta c\eta s\kappa & (s\zeta)^2s\eta c\eta(1-c\kappa) + c\zeta s\kappa \\ s\zeta c\zeta s\eta(1-c\kappa) + s\zeta c\eta s\kappa & (c\zeta)^2(1-c\kappa) + c\kappa & s\zeta c\zeta c\eta(1-c\kappa) - s\zeta s\eta s\kappa \\ (s\zeta)^2s\eta c\eta(1-c\kappa) - c\zeta s\kappa & s\zeta c\zeta c\eta(1-c\kappa) + s\zeta s\eta s\kappa & (s\zeta)^2(c\eta)^2(1-c\kappa) + c\kappa \end{bmatrix}$	$\kappa = \arccos[(r_{11} + r_{22} + r_{33} - 1)/2]$ $\kappa = 0: \zeta = \eta = 0$ $0 < \kappa < \pi: \zeta = \arccos[(r_{13} - r_{31})/(2s\kappa)]$ $\zeta = 0$ or $\zeta = \pi: \eta = 0$ $0 < \zeta < \pi: \eta = \text{atan2}(r_{32} - r_{23}, r_{21} - r_{12})$ $\kappa = \pi: \zeta = \frac{1}{2}\arccos(r_{22})$ $\zeta = 0: \eta = 0$ $\zeta = \pi/2: \eta = \frac{1}{2}\text{atan2}(r_{31} + r_{13}, 2r_{33})$ $0 < \zeta < \pi/2: \eta = \text{atan2}(r_{12} + r_{21}, r_{32} + r_{23})$

A close description is that by quaternions. Here, the coordinates of the rotation axis are ‘merged’ with the rotation angle, giving a set of four quaternion parameters $\{q_0, q_x, q_y, q_z\}$ that are $\{\cos(\kappa/2), l_x \sin(\kappa/2), l_y \sin(\kappa/2), l_z \sin(\kappa/2)\}$. These quartets have a number of important mathematical features and are convenient for formal mathematical manipulations. The corresponding rotation matrix is also given in Table 1.

3.3.2. Polar angles and rotation axes. The normalization condition $l_x^2 + l_y^2 + l_z^2 = 1$ on vector \mathbf{l} indicates that the parameters of direction cosines are mutually dependent; the same

is true for quaternions. Instead of these parameters, one can define \mathbf{l} by its spherical coordinates. Here one angle describes the inclination of \mathbf{l} , i.e. the angle that the vector \mathbf{l} makes with a chosen coordinate axis, called the ‘zenith’ or ‘polar’ axis and which we denote by \mathbf{p} . By definition this angle varies from 0 to π . The plane normal to \mathbf{p} contains two other vectors of the Cartesian basis. One of them is chosen as the ‘azimuth’ or ‘horizontal’ axis, and we denote it by \mathbf{h} . Vectors \mathbf{p} and \mathbf{h} are completed by the third unit vector \mathbf{n} (for normal), making a right-hand orthogonal frame \mathbf{pnh} (Appendix C).

The angle that the orthogonal projection of \mathbf{l} into the plane \mathbf{hOn} makes with the azimuth axis \mathbf{h} is the second angle defining the orientation of the rotation axis. This angle is considered in the trigonometric sense looking against \mathbf{p} . The two angles introduced above fully define the orientation of the rotation axis \mathbf{l} [Fig. 1(c)]. We call these two angles the ‘zenith angle’, ζ , and the ‘azimuth angle’, η , reserving the term ‘polar angles’ for the ensemble of three angles (ζ, η, κ) as routinely used in macromolecular crystallography. (Some sources reserve the term ‘polar angle’ only for a rotation in a plane.) For $\zeta = 0$ and for $\zeta = \pi$ the azimuth angle η is undefined and we arbitrarily assign $\eta = 0$. With this choice, the rotation matrix can be expressed as a product of the elemental matrices (Appendix C):

$$R = R_p(\eta) R_n(\zeta) R_p(\kappa) R_n(-\zeta) R_p(-\eta). \quad (6)$$

All rotations in equation (6) are considered in the trigonometrical sense.

3.3.3. Rotation direction and system rotation. Both inversion of the rotation direction and consideration of the basis rotating about a fixed object result in inversion of the sign of the argument of the central matrix in (6), *i.e.* replacing $R_p(\kappa)$ by $R_p(-\kappa)$, and keeping the rest.

3.4. Conversion of rotation matrices into parameters

Previous sections have shown how to obtain the rotation matrix from different kinds of rotation parameters. Conversely, given the rotation matrix, one may wish to recover the independent rotation parameters of one’s choice. Tables 1–3 provide the corresponding expressions considering a fixed system and the object rotated about fixed axes in the trigonometric sense. For alternative conventions, the final assignment is done according to the remark in §3.3.3 or by comparison of equation (1) with the respective expressions (2)–(5). These tables show that the extraction of the rotation parameters from the rotation matrix can always be done unambiguously, except for the equivalent solution. Several remarks have to be made.

First, the expressions for parameters are different for general and for some particular situations, *e.g.* those when $\kappa = \pi$ for polar angles or when $\kappa_2 = 0$ for proper Euler angles *etc.* For these situations, usually the value of one of the parameters is assigned arbitrarily. Second, in practice these particular situations are identified with some precision and it is crucial to keep rounding errors as small as possible. Third, each solution has its equivalent; the reason for this is that each rotation can be considered also as a rotation by the complementary (or opposite) angle about the axis directed in the opposite way, *i.e.* inverting the vector \mathbf{l} .

Finally, a cyclic permutation of rotation axes in the definition of the Euler or polar angles results in a cyclic permutation of rows and columns of the rotation matrices (Tables 2 and 3). In turn, this corresponds to a cyclic permutation of indices of the matrix elements in the expressions for Euler or polar angles extracted from the rotation matrix.

4. Program

4.1. Overall structure

The program core is represented by 12 modules corresponding to all possible independent assignments of rotation axes for Euler angles [see, for example, Kuipers (1999) or <https://www.mecademic.com/resources/Euler-angles/Euler-angles>] and by three modules for all possible standard choices of the zenith and azimuth axes. Each of these modules contains the part that interprets the matrices in terms of the relevant Euler or polar angles. In fact, only three modules for $R_x(\kappa)$, $R_y(\kappa)$ and $R_z(\kappa)$ are sufficient to calculate any rotation matrix by combining them appropriately, but the interpretation of the matrix in terms of parameters would be less straightforward. There exists also a module for matrix calculation and interpretation in terms of direction cosines.

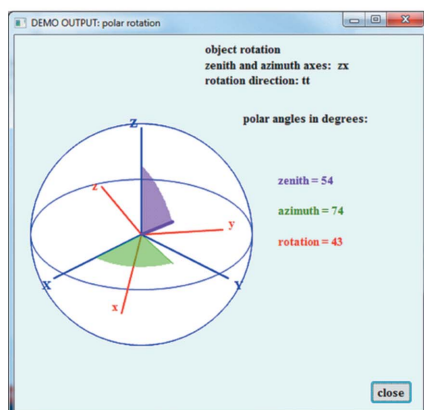
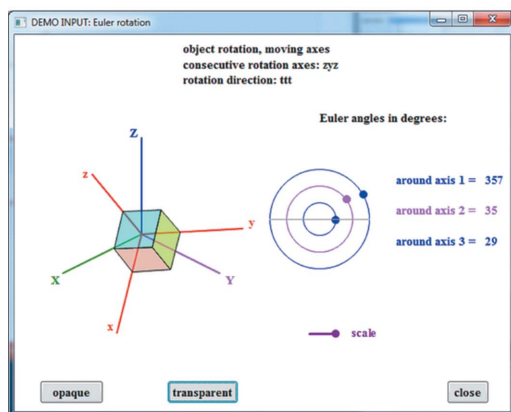
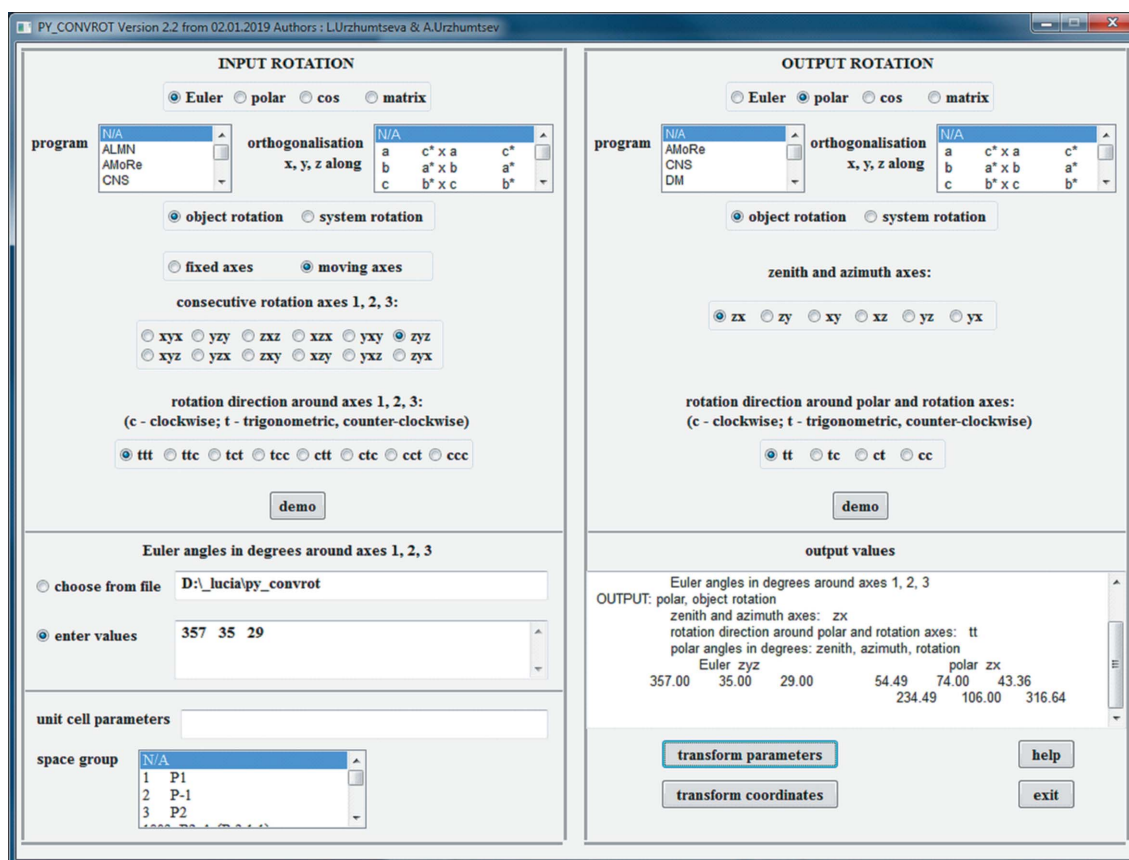
The program calls the appropriate module according to the particular choice of parameters made by the user. If necessary, the angle values are adjusted internally, for example are taken in the inverse order, taken with the inverse signs (Appendices B and C) or increased by $\pi/2$ for a nonstandard choice of azimuth axis (Appendix C). A corresponding adjustment is required when extracting the parameters from the matrix.

The program has a module to generate the parameters corresponding to symmetrically equivalent orientations for a selected space group given the rotation matrix defining the principal position of the rotated object. The orthogonalization and deorthogonalization module is included for use of the program in a crystallographic environment. Finally, there is a module to apply the rotation matrix (and the translation vector) to a list of atomic coordinates. Note that the respective anisotropic atomic displacement parameters and the TLS parameters (Schomaker & Trueblood, 1968) are not recalculated (Kronenburg, 2004).

4.2. Protocol

The program can be used in a crystallographic environment but not necessarily. The choice of a crystallographic environment means that either crystallographic symmetries will be used or the rotation elements (*e.g.* the polar axis) will be defined with respect to the crystallographic axes. In this optional case the user needs to first select the orthogonalization conventions; they may be different for the input and output parameters. Then, independently of this step, the user needs to choose the particular sets of the input and output rotation parameters. In a simplified mode of conversion from one known program to another, the settings above can be done automatically by selecting the program names in the list. This list is very short in comparison with that given by Urzhumtseva & Urzhumtsev (1997). One reason is that many programs have become obsolete since 1997. The second reason is that we wanted to avoid an eventual disagreement between the formal program descriptions, available comments on the choice of parameters and the actual program source.

The values of the chosen input parameters can be either entered directly by typing them or using copy/paste options or using *Demo*, or provided by indicating the file containing such



Euler: $X \rightarrow Y \rightarrow Z$ (fixed axes) Matrix product: $R = R_z(\kappa_3) R_y(\kappa_2) R_x(\kappa_1)$

Interpretation (rotating the object) :

Fixed axes :

- first rotation by κ_1 about X
- second rotation by κ_2 about Y
- third rotation by κ_3 about Z

Moving axes :

- first rotation by κ_3 about Z
- second rotation by κ_2 about new Y
- third rotation by κ_1 about new X

Rotations in terms of moving axes associated with the object :

Rotation 1 Rotation 2 Rotation 3

Rotation matrix R :

$$\begin{pmatrix} \cos \kappa_2 \cos \kappa_3 & \sin \kappa_1 \sin \kappa_2 \cos \kappa_3 - \cos \kappa_1 \sin \kappa_3 & \cos \kappa_1 \sin \kappa_2 \cos \kappa_3 + \sin \kappa_1 \sin \kappa_3 \\ \cos \kappa_2 \sin \kappa_3 & \sin \kappa_1 \sin \kappa_2 \sin \kappa_3 + \cos \kappa_1 \cos \kappa_3 & \cos \kappa_1 \sin \kappa_2 \sin \kappa_3 - \sin \kappa_1 \cos \kappa_3 \\ -\sin \kappa_2 & \sin \kappa_1 \cos \kappa_2 & \cos \kappa_1 \cos \kappa_2 \end{pmatrix}$$

Getting angles from the matrix

$$\begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \quad \begin{matrix} 0 \leq \kappa_1 < 2\pi \\ -\frac{1}{2}\pi \leq \kappa_2 \leq \frac{1}{2}\pi \\ 0 \leq \kappa_3 < 2\pi \end{matrix}$$

$\kappa_2 = -\arcsin(R_{31})$

If $\cos \kappa_2 = 0$: $\kappa_1 = \text{atan2}(-R_{23}, R_{22})$, $\kappa_3 = 0$

If $\cos \kappa_2 > 0$: $\kappa_1 = \text{atan2}(R_{32}, R_{33})$, $\kappa_3 = \text{atan2}(R_{21}, R_{11})$

Equivalent angles : $\kappa_1' = \kappa_1 + \pi$; $\kappa_2' = \pi - \kappa_2$; $\kappa_3' = \kappa_3 + \pi$

Relevant software : n/a

Figure 2 `py_convrot` session. The principal window (top), two snapshots of the interactive *Demo* (lower left) and an example of a *Help* page (lower right).

values. The last option is convenient when converting lists of parameters, e.g. Euler angles after a molecular replacement run. The recalculation of the parameters can be followed by recalculation of coordinates; input and output files with lists of atomic coordinates and an eventual model shift have to be defined.

Any choice of the rotation parameters can be illustrated by *Demo*; this can be done separately for the input and output parameter selection. For input parameters, *Demo* is interactive; the user can vary parameter values and observe respective changes in the axes and object positions. At any time, *Help* is available, explaining the rotation matrices and their interpretation in terms of angles, as well as illustrating the orthogonalization conventions.

Fig. 2 shows an example of a *py_convrot* session.

5. Software environment and program distribution

The GUI program *py_convrot* is written in Python 2 and requires version 2.6 or 2.7 and the wxPython toolkit. *py_convrot* has been tested under Windows, Linux and MacOS systems. If the Python libraries are installed in the frame of other program packages such as *PHENIX* (Adams *et al.*, 2010), the program may be started as

```
phenix.wxpython py_convrot.py
```

The program can be downloaded from the site <http://www-ibmc.u-strasbg.fr/spip-arn/spip.php?rubrique2&lang=en>. It is also available on request from the authors. A Python 3 version is under development.

6. Conclusions

Description of rotations in space is an old task widely addressed in the literature. The algebraic simplicity of its solution allows developers to select and adapt the description that is most appropriate to their particular project. At the same time, the variety of possible descriptions causes a lot of confusion. The present article and associated teaching program *py_convrot*, addressed to researchers working in crystallography, cryoEM or similar fields, are an attempt to review these descriptions and relate them to each other.

APPENDIX A

Rotation matrices in Cartesian coordinate systems

Let an object be defined in some Cartesian coordinate system with basis vectors $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ (right-hand orthonormal basis) and the origin around which the object is rotated at \mathbf{O} . Let $\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z$ be three other unit vectors, rigidly associated with the rotated object and such that for the initial position of the object they coincide with $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$, respectively. After rotation, these vectors can be expressed in the basis $\{\mathbf{O}, \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ as

$$\begin{aligned}\mathbf{u}'_x &= r_{11}\mathbf{e}_x + r_{21}\mathbf{e}_y + r_{31}\mathbf{e}_z, & \mathbf{u}'_y &= r_{12}\mathbf{e}_x + r_{22}\mathbf{e}_y + r_{32}\mathbf{e}_z, \\ \mathbf{u}'_z &= r_{13}\mathbf{e}_x + r_{23}\mathbf{e}_y + r_{33}\mathbf{e}_z.\end{aligned}\quad (7)$$

If we compose a matrix R with these coordinates in columns as

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}\quad (8)$$

then the coordinates (x'_e, y'_e, z'_e) of any point of the object after its rotation are related to its coordinates (x_e, y_e, z_e) in the initial position by

$$\begin{pmatrix} x'_e \\ y'_e \\ z'_e \end{pmatrix} = R \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}.\quad (9)$$

The rotation matrix R , called also the matrix of the rotation operator, is the same for all points. As a consequence, it can be considered as the reference entity for any kind of parameterization of the rotation. We should distinguish between equation (9) and the equations that relate the coordinates of a fixed point in the initial basis and in the rotated basis, (x_u, y_u, z_u) , namely²

$$\begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix} = R^{-1} \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} = R \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix}.\quad (10)$$

It follows from equation (9) that two consecutive rotations, first by R_1 and then by R_2 , are equivalent to a rotation described by the matrix $R = R_2 R_1$ with the matrices taken from right to left (all matrices are in the same Cartesian basis):

$$\begin{pmatrix} x''_e \\ y''_e \\ z''_e \end{pmatrix} = R_2 \begin{pmatrix} x'_e \\ y'_e \\ z'_e \end{pmatrix} = R_2 \left[R_1 \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} \right] = (R_2 R_1) \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}.\quad (11)$$

APPENDIX B

Euler angles

B1. Extrinsic rotations (fixed coordinate axes)

As shown by Euler (1776), a rigid-body rotation can be described by three consecutive rotations about the coordinate axes **axis1**, **axis2**, **axis3** by angles $\kappa_1, \kappa_2, \kappa_3$, respectively. In the case of extrinsic rotations (rotations about fixed coordinate axes), **axis1**, **axis2**, **axis3** are selected from $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$. The same axis cannot be taken for two consecutive elemental rotations.

For elemental rotations by angle κ about axes $\mathbf{e}_x, \mathbf{e}_y$ or \mathbf{e}_z the corresponding rotation matrices [equation (8)] are

² In an orthonormal basis, the inverse matrix coincides with the transposed one, $R^{-1} = R^*$.

$$\begin{aligned}
 R_x(\kappa) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \kappa & -\sin \kappa \\ 0 & \sin \kappa & \cos \kappa \end{pmatrix}, \\
 R_y(\kappa) &= \begin{pmatrix} \cos \kappa & 0 & \sin \kappa \\ 0 & 1 & 0 \\ -\sin \kappa & 0 & \cos \kappa \end{pmatrix}, \\
 R_z(\kappa) &= \begin{pmatrix} \cos \kappa & -\sin \kappa & 0 \\ \sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{pmatrix}.
 \end{aligned} \tag{12}$$

Here each rotation is considered in its trigonometrical sense, *i.e.* counterclockwise when looking against the rotation axis [Fig. 1(a)]. According to equation (11) the total rotation matrix is calculated as a product [equation (1)] of the individual matrices.

B2. Intrinsic rotations (moving coordinate axes)

Alternatively, **axis1**, **axis2**, **axis3** may be the axes of a coordinate system rigidly associated with the rotating object and moving with it. This is an intrinsic rotation. We denote this rotating basis $\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z$ by U, distinguishing it from the original immobile basis $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ which we denote by E. The immobile point **O** is the origin for both of them. Note that for both extrinsic and intrinsic rotations, ultimately, we are interested in the coordinates of a point with respect to the initial, fixed coordinate system $\{\mathbf{O}, \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$, while $\{\mathbf{O}, \mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z\}$ is simply an intermediate tool to describe the rotation.

Let R_1 correspond to a rotation of the object, together with the basis U, from its initial position. For any point of a rotating object, its coordinates in U do not change and are the same as initially in basis E: $(x_u, y_u, z_u) = (x_e, y_e, z_e)$. If R_2 is the matrix describing a following rotation with respect to the rotated basis U,

$$\begin{pmatrix} x'_u \\ y'_u \\ z'_u \end{pmatrix} = R_2 \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix}, \tag{13}$$

the coordinates of the rotated point in E, according to equation (10), become

$$\begin{aligned}
 \begin{pmatrix} x'_e \\ y'_e \\ z'_e \end{pmatrix} &= R_1 \begin{pmatrix} x'_u \\ y'_u \\ z'_u \end{pmatrix} = R_1 \left[R_2 \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix} \right] = R_1 \left[R_2 \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} \right] \\
 &= (R_1 R_2) \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}.
 \end{aligned} \tag{14}$$

In other words, for intrinsic rotations the matrix product is calculated in the reverse order to that for extrinsic ones: matrices are taken from left to right. Generalizing this, the matrix to calculate Cartesian coordinates after three consecutive rotations about moving axes **axis1**, **axis2**, **axis3** is equation (2).

B3. Extrinsic versus intrinsic conventions

Table 2 shows the resulting matrix [equation (8)] for rotations about all possible triplets of fixed axes. A direct comparison of formula (2) with (1) shows that each extrinsic rotation can be easily interpreted as an intrinsic rotation and *vice versa*. In other words, each rotation matrix can be interpreted in terms of both conventions.

The convention most frequently used in the past few decades in macromolecular crystallography (*e.g.* Collaborative Computational Project, Number 4, 1994) is that described by the matrix

$$R = R_z(\kappa_1)R_y(\kappa_2)R_z(\kappa_3). \tag{15}$$

It can be interpreted³ as an intrinsic rotation **zyz** (moving, mobile axes) about **axis1** = \mathbf{u}_z , **axis2** = \mathbf{u}_y , **axis3** = \mathbf{u}_z by angles $\kappa_1, \kappa_2, \kappa_3$, respectively. This is equivalent to the consecutive rotations about the fixed axes (extrinsic rotations) **axis1** = \mathbf{e}_z , **axis2** = \mathbf{e}_y , **axis3** = \mathbf{e}_z by angles $\kappa_3, \kappa_2, \kappa_1$. Note that, similarly to other proper Euler angles, this convention is not optimal for rigid-body refinement since in that case the original orientation corresponding to $\kappa_2 = 0$ is singular, defining the sum $\kappa_1 + \kappa_3$ and not each of these angles individually (Urzhumtsev *et al.*, 1989; Afonine *et al.*, 2009).

B4. Other rotation conventions

Inversion of the rotation direction corresponds to taking the opposite sign for the corresponding rotation angle in equation (1) or (2). In general, for two angles from the same triplet of Euler angles, opposite directions can be considered as positive, *e.g.* https://spider.wadsworth.org/spider_doc/spider/docs/euler.html.

Rotating a system (basis) about the fixed object is opposite to an equivalent rotation of the object with respect to a fixed basis. Three consecutive rotations of the basis about the axes **axis1**, **axis2**, **axis3** (which by construction are moving axes) by angles $\alpha_1, \alpha_2, \alpha_3$ are described by matrix (2). However, since an opposite rotation is described by the inverse matrix, matrix (8) to recalculate the Cartesian coordinates of a fixed point with respect to the rotated basis, which is the final goal, becomes

$$\begin{aligned}
 R &= [R_{\text{axis1}}(\kappa_1)R_{\text{axis2}}(\kappa_2)R_{\text{axis3}}(\kappa_3)]^{-1} \\
 &= R_{\text{axis3}}^{-1}(\kappa_3)R_{\text{axis2}}^{-1}(\kappa_2)R_{\text{axis1}}^{-1}(\kappa_1) \\
 &= R_{\text{axis3}}(-\kappa_3)R_{\text{axis2}}(-\kappa_2)R_{\text{axis1}}(-\kappa_1).
 \end{aligned} \tag{16}$$

APPENDIX C

Polar angles

C1. Definition of the axis direction

If we characterize a rotation in three-dimensional space by a matrix R , a unit vector **I** along the rotation axis is its eigenvector corresponding to its eigenvalue equal to 1. Such an eigenvalue always exists and is unique except for the

³ See comments in http://www.ccp4.ac.uk/html/rotationmatrices.html#eulerian_angles, which show once more the usefulness of the matrix as a reference object to describe rotations and to avoid confusion.

Table 4
Orthogonalization conventions.

$a, b, c, \alpha, \beta, \gamma$ are cell parameters; $\alpha^*, \beta^*, \gamma^*$ are angles between the vectors of the dual basis calculated as $\alpha^* = \arccos[(\cos \beta \cos \gamma - \cos \alpha)/(\sin \beta \sin \gamma)]$, $\beta^* = \arccos[(\cos \gamma \cos \alpha - \cos \beta)/(\sin \gamma \sin \alpha)]$, $\gamma^* = \arccos[(\cos \alpha \cos \beta - \cos \gamma)/(\sin \alpha \sin \beta)]$. For the fourth orthogonalization convention, $a' = \|\mathbf{a} + \mathbf{b}\| = (a^2 + b^2 + 2ab \cos \gamma)^{1/2}$, $\beta' = \arccos[(a \cos \beta + b \cos \alpha)/a']$, $\gamma' = \arccos[(a \cos \gamma + b)/a']$, $\cos \beta'^* = (\cos \gamma' \cos \alpha - \cos \beta')/(\sin \gamma' \sin \alpha)$.

$\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ along vectors	Orthogonalization matrix B	Deorthogonalization matrix A
\mathbf{a} $\mathbf{c}^* \times \mathbf{a}$ \mathbf{c}^*	$\begin{pmatrix} a & b \cos \gamma & c \cos \beta \\ 0 & b \sin \gamma & -c \sin \beta \cos \alpha^* \\ 0 & 0 & c \sin \beta \sin \alpha^* \end{pmatrix}$	$\begin{bmatrix} 1/a & -1/(a \tan \gamma) & \sin \alpha \cos \beta^*/(a \sin \beta \sin \gamma \sin \alpha^*) \\ 0 & 1/(b \sin \gamma) & 1/(b \sin \gamma \tan \alpha^*) \\ 0 & 0 & 1/(c \sin \beta \sin \alpha^*) \end{bmatrix}$
\mathbf{b} $\mathbf{a}^* \times \mathbf{b}$ \mathbf{a}^*	$\begin{pmatrix} a \cos \gamma & b & c \cos \alpha \\ -a \sin \gamma \cos \beta^* & 0 & c \sin \alpha \\ a \sin \gamma \sin \beta^* & 0 & 0 \end{pmatrix}$	$\begin{bmatrix} 0 & 0 & 1/(a \sin \gamma \sin \beta^*) \\ 1/b & -1/(b \tan \alpha) & \sin \beta \cos \gamma^*/(b \sin \alpha \sin \gamma \sin \beta^*) \\ 0 & 1/(c \sin \alpha) & 1/(c \sin \alpha \tan \beta^*) \end{bmatrix}$
\mathbf{c} $\mathbf{b}^* \times \mathbf{c}$ \mathbf{b}^*	$\begin{pmatrix} a \cos \beta & b \cos \alpha & c \\ a \sin \beta & -b \sin \alpha \cos \gamma^* & 0 \\ 0 & b \sin \alpha \sin \gamma^* & 0 \end{pmatrix}$	$\begin{bmatrix} 0 & 1/(a \sin \beta) & 1/(a \sin \beta \tan \gamma^*) \\ 0 & 0 & 1/(b \sin \alpha \sin \gamma^*) \\ 1/c & -1/(c \tan \beta) & \sin \gamma \cos \alpha^*/(c \sin \alpha \sin \beta \sin \gamma^*) \end{bmatrix}$
$(\mathbf{a} + \mathbf{b})$ $\mathbf{c}^* \times (\mathbf{a} + \mathbf{b})$ \mathbf{c}^*	$\begin{pmatrix} a' & b \cos \gamma & c \cos \beta' \\ 0 & b \sin \gamma & -c \sin \beta' \cos \alpha^* \\ 0 & 0 & c \sin \beta' \sin \alpha^* \end{pmatrix}$	$\begin{bmatrix} 1/a' & -1/(a' \tan \gamma') & \sin \alpha \cos \beta^*/(a' \sin \beta' \sin \gamma' \sin \alpha^*) \\ 0 & 1/(b \sin \gamma') & 1/(b \sin \gamma' \tan \alpha^*) \\ 0 & 0 & 1/(c \sin \beta' \sin \alpha^*) \end{bmatrix}$
\mathbf{a}^* $\mathbf{c} \times \mathbf{a}^*$ \mathbf{c}	$\begin{pmatrix} a \sin \beta \sin \gamma^* & 0 & 0 \\ -a \sin \beta \cos \gamma^* & b \sin \alpha & 0 \\ a \cos \beta & b \cos \alpha & c \end{pmatrix}$	$\begin{bmatrix} 1/(a \sin \beta \sin \gamma^*) & 0 & 0 \\ 1/(b \sin \alpha \tan \gamma^*) & 1/(b \sin \alpha) & 0 \\ \sin \gamma \cos \beta^*/(c \sin \alpha \sin \beta \sin \gamma^*) & -1/(c \tan \alpha) & 1/c \end{bmatrix}$
\mathbf{a} \mathbf{b}^* $\mathbf{a} \times \mathbf{b}^*$	$\begin{pmatrix} a & b \cos \gamma & c \cos \beta \\ 0 & b \sin \gamma \sin \alpha^* & 0 \\ 0 & -b \sin \gamma \cos \alpha^* & c \sin \beta \end{pmatrix}$	$\begin{bmatrix} 1/a & \sin \alpha \cos \gamma^*/(a \sin \beta \sin \gamma \sin \alpha^*) & -1/(a \tan \beta) \\ 0 & 1/(b \sin \gamma \sin \alpha^*) & 0 \\ 0 & 1/(c \sin \beta \tan \alpha^*) & 1/(c \sin \beta) \end{bmatrix}$
\mathbf{a}^* \mathbf{b} $\mathbf{a}^* \times \mathbf{b}$	$\begin{pmatrix} a \sin \gamma \sin \beta^* & 0 & 0 \\ a \cos \gamma & b & c \cos \alpha \\ -a \sin \gamma \cos \beta^* & 0 & c \sin \alpha \end{pmatrix}$	$\begin{bmatrix} 1/(a \sin \gamma \sin \beta^*) & 0 & 0 \\ \sin \beta \cos \gamma^*/(b \sin \alpha \sin \gamma \sin \beta^*) & 1/b & -1/(b \tan \alpha) \\ 1/(c \sin \alpha \tan \beta^*) & 0 & 1/(c \sin \alpha) \end{bmatrix}$

degenerate rotation by 0 radians; this value is characteristic for a rotation. (The two other eigenvalues either are complex conjugates of modulus 1 in a general case or are both equal to 1 for a rotation by 0, or are both equal to -1 for a rotation by π radians.) Two independent parameters are required to define \mathbf{l} . Frequently, they are chosen as two angles that refer to an orthonormal basis:

(a) The zenith angle ζ that \mathbf{l} makes with a chosen basis vector \mathbf{p} that we call the polar or zenith axis; for example, but not necessarily (e.g. Rossmann & Blow, 1962), $\mathbf{p} = \mathbf{e}_z$.

(b) The azimuth angle η that the orthogonal projection of \mathbf{l} into the plane normal to \mathbf{p} makes with one of two basis vectors that belong to this plane; we call this basis vector the azimuth axis, \mathbf{h} .

It is more convenient to choose \mathbf{h} so that \mathbf{p}, \mathbf{h} and the third basis vector, \mathbf{n} , and not its opposite, form a right-hand frame (this means that $\mathbf{n} = \mathbf{p} \times \mathbf{h}$). For example, for $\mathbf{p} = \mathbf{e}_z$, this

imposes $\mathbf{h} = \mathbf{e}_x$, resulting in $\mathbf{n} = \mathbf{e}_y$. With such a convention, the three basic possible choices for \mathbf{p} and \mathbf{h} are $(\mathbf{e}_z, \mathbf{e}_x)$, $(\mathbf{e}_x, \mathbf{e}_y)$, $(\mathbf{e}_y, \mathbf{e}_z)$. In the literature, the term ‘polar angle’ is reserved sometimes for the first, inclination angle, and sometimes for the second angle, especially when working in plane. Also one may find multiple notations for these angles, including some that are opposite to each other, e.g. (θ, ϕ) and (ϕ, θ) . For these reasons we here use ζ (for ‘zenith’) and η (for ‘azimuth’), differently from common notations, to avoid any confusion and showing explicitly which angle is for which axis. We use the term ‘polar angles’ for the set (ζ, η, κ) , where κ is the angle of rotation about \mathbf{l} . Recall that the rotation angle κ is related to the trace of the rotation matrix by expression

$$2 \cos \kappa + 1 = \text{tr } R = r_{11} + r_{22} + r_{33}. \quad (17)$$

C2. Rotation about an axis in a general orientation

To find the matrix describing the rotation by angle κ about axis \mathbf{l} in a general orientation, a moving frame (basis) \mathbf{V} is defined with the vectors $\mathbf{v}_h, \mathbf{v}_n, \mathbf{v}_p$, which initially coincide with $\mathbf{h}, \mathbf{n}, \mathbf{p}$. It is rotated first by η about $\mathbf{v}_p = \mathbf{p}$ and then by ζ about \mathbf{v}_n in its new position. This superposes \mathbf{v}_p and \mathbf{l} ; the corresponding rotation matrix is

$$R_{pl} = R_p(\eta)R_n(\zeta). \quad (18)$$

According to equation (10), the coordinates of a given fixed point expressed in the rotated basis \mathbf{V} are

$$\begin{aligned} \begin{pmatrix} x_v \\ y_v \\ z_v \end{pmatrix} &= R_{pl}^{-1} \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} = [R_p(\eta)R_n(\zeta)]^{-1} \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} \\ &= R_n(-\zeta)R_p(-\eta) \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}. \end{aligned} \quad (19)$$

Rotation by κ about $\mathbf{v}_p = \mathbf{p}$ in the rotated basis is described by the product of the matrix $R_p(\kappa)$ and vector (19), where $R_p(\kappa)$ is one of the elemental matrices (12). Finally, and also according to (10), recalculating the resulting coordinates in the original coordinate system gives the total rotation matrix (8) in the form

$$\begin{aligned} R \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} &= R_{pl}R_p(\kappa) \begin{pmatrix} x_v \\ y_v \\ z_v \end{pmatrix} \\ &= R_p(\eta)R_n(\zeta)R_p(\kappa)R_n(-\zeta)R_p(-\eta) \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}. \end{aligned} \quad (20)$$

C3. Nonstandard choice for the azimuth axis

Table 3 shows the rotation matrices for all three possible standard choices for the zenith and azimuth axes. However, for the same axis \mathbf{p} one may choose axis \mathbf{h} coinciding with a different coordinate axis; in other words, the new azimuth and normal axes, \mathbf{h}' and \mathbf{n}' , may correspond to \mathbf{n} and $-\mathbf{h}$ from the previous, standard choice. One may reduce such a situation to the previous one since the angles (ζ', η') for a non-conventional choice are equivalent to the angles ($\zeta, \eta + \pi/2$) defined for the standard choice [Fig. 1(d)]. The program *py_convrot* uses this approach even when other considerations are possible.

APPENDIX D

Orthogonalization conventions and their relation to rotations

In crystallography, three kinds of basis are used and are involved in the definition of the rotation matrices in a crystallographic environment [see for example Shmueli (2010) or

Rupp (2010)]. First, crystallographic (fractional) coordinates are defined, representing each point (vector) \mathbf{r} in a right-hand basis composed of vectors defining the crystal periods $\mathbf{a}, \mathbf{b}, \mathbf{c}$:

$$\mathbf{r} = x_f\mathbf{a} + y_f\mathbf{b} + z_f\mathbf{c}. \quad (21)$$

Generally speaking, these basis vectors are not unit vectors and are not mutually orthogonal. The mixed product

$$V = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} \quad (22)$$

is the volume of the unit cell formed by the basis vectors. The second basis, dual to the previous one, is composed of the vectors

$$\mathbf{a}^* = V^{-1}\mathbf{b} \times \mathbf{c}, \quad \mathbf{b}^* = V^{-1}\mathbf{c} \times \mathbf{a}, \quad \mathbf{c}^* = V^{-1}\mathbf{a} \times \mathbf{b}. \quad (23)$$

The third is a Cartesian orthonormal basis $\{\mathbf{O}, \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ with respect to which the rotation parameters are chosen. It is defined by one of the standard conventions (Collaborative Computational Project, Number 4, 1994) through $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$ [see for example Giacovazzo *et al.* (1992) or Rupp (2010)]. The crystallographic [equation (21)] and Cartesian coordinates

$$\mathbf{r} = x_e\mathbf{e}_x + y_e\mathbf{e}_y + z_e\mathbf{e}_z \quad (24)$$

are related to each other by two mutually inverse matrices, orthogonalization matrix B and deorthogonalization matrix $A = B^{-1}$:

$$\begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix} = A \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}, \quad \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} = B \begin{pmatrix} x_f \\ y_f \\ z_f \end{pmatrix}. \quad (25)$$

Table 4 summarizes the expressions for matrices A and B for different conventions. Changing the orthogonalization choice changes the rotation matrix according to

$$A_{in}R_{in}B_{in} = A_{out}R_{out}B_{out}. \quad (26)$$

Here both left-hand and right-hand parts of the equation present the rotation matrix in the crystallographic basis.

While rotations in Euler or in polar angles are defined with respect to a Cartesian basis, crystallographic transformations are naturally defined in crystallographic coordinates. For example, let a rotation described by matrix R_{in} in some input Cartesian basis be followed by a symmetry rotation S . According to equation (26), the matrix for the resulting rotation expressed in another, output Cartesian basis is equal to

$$R_{out} = A_{out}^{-1}S(A_{in}R_{in}B_{in})B_{out}^{-1} = B_{out}S(A_{in}R_{in}B_{in})A_{out}. \quad (27)$$

Acknowledgements

The authors thank D. Tronrud, D. Kostrewa, O. Sobolev, V. Y. Lunin, R. Read and A. Brunger for useful comments and A. McEwen for his help with improving the manuscript. AU

acknowledges the support and the use of resources of the French Infrastructure for Integrated Structural Biology FRISBI ANR-10-INBS-05, and of Instruct-ERIC.

References

- Adams, P. D., Afonine, P. V., Bunkóczi, G., Chen, V. B., Davis, I. W., Echols, N., Headd, J. J., Hung, L.-W., Kapral, G. J., Grosse-Kunstleve, R. W., McCoy, A. J., Moriarty, N. W., Oeffner, R., Read, R. J., Richardson, D. C., Richardson, J. S., Terwilliger, T. C. & Zwart, P. H. (2010). *Acta Cryst.* **D66**, 213–221.
- Afonine, P. V., Grosse-Kunstleve, R. W., Urzhumtsev, A. & Adams, P. D. (2009). *J. Appl. Cryst.* **42**, 607–615.
- Bigras, P. & Bonev, I. (2012). *Simulation des Angles d'Euler*, <https://cours.etsmtl.ca/gpa546/Applets/Orient3D.html>.
- Brünger, A. T., Adams, P. D., Clore, G. M., DeLano, W. L., Gros, P., Grosse-Kunstleve, R. W., Jiang, J.-S., Kuszewski, J., Nilges, M., Pannu, N. S., Read, R. J., Rice, L. M., Simonson, T. & Warren, G. L. (1998). *Acta Cryst.* **D54**, 905–921.
- Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* **D50**, 760–763.
- DeLano, W. L. (2002). *The pyMOL Molecular Graphics System*. DeLano Scientific, San Carlos, CA, USA.
- Diamond, R. & Cranswick, L. M. D. (2010). *International Tables for Crystallography*, Vol. B, *Reciprocal Space*, 2nd online ed., pp. 418–448. Chester: International Union of Crystallography.
- Emsley, P., Lohkamp, B., Scott, W. G. & Cowtan, K. (2010). *Acta Cryst.* **D66**, 486–501.
- Euler, L. (1776). *Novi Comment. Acad. Sci. Imp. Petropol.* **20**, 189–207.
- Gaschler, A. (2016). *3D Rotation Converter*, <http://www.andre-gaschler.com/rotationconverter/>.
- Giacovazzo, C., Monaco, H. L., Viterbo, D., Scordari, F., Gilli, G., Zanotti, G. & Catti, M. (1992). *Fundamentals of Crystallography*, edited by G. Giacovazzo. Oxford University Press.
- Goldstein, H. (1950). *Classical Mechanics*. Cambridge: Addison-Wesley.
- Heymann, J. B., Chagoyen, M. & Belnap, D. M. (2005). *J. Struct. Biol.* **151**, 196–207.
- Heymann, J. B., Chagoyen, M. & Belnap, D. M. (2006). *J. Struct. Biol.* **153**, 312.
- Kronenburg, M. J. (2004). *Acta Cryst.* **A60**, 250–256.
- Kuipers, J. B. (1999). *Quaternions and Rotation Sequences – A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton University Press.
- Lattman, E. E. (1972). *Acta Cryst.* **B28**, 1065–1068.
- Morawiec, A. (2015). *A Description of Pan v. 1.979*, http://imim.pl/personal/adam.morawiec/A_Morawiec_Web_Page/Pan.pdf.
- Rossmann, M. G. & Blow, D. M. (1962). *Acta Cryst.* **15**, 24–31.
- Rupp, B. (2010). *Biomolecular Crystallography. Principles, Practice, and Application to Structural Biology*. New York: Garland Science, Taylor & Francis Group.
- Schomaker, V. & Trueblood, K. N. (1968). *Acta Cryst.* **B24**, 63–76.
- Shmueli, U. (2010). *International Tables for Crystallography*, Vol. B, *Reciprocal Space*, 2nd online ed., pp. 2–9. Chester: International Union of Crystallography.
- Tulloue, G. (2001). *Coordonnées Sphériques*, http://www.sciences.univ-nantes.fr/sites/genevieve_tulloue/Meca/Cinematique/coord_spheriques.php.
- Urzhumtsev, A. G., Lunin, V. Yu. & Vernoslova, E. A. (1989). *J. Appl. Cryst.* **22**, 500–506.
- Urzhumtsev, A. & Urzhumtseva, L. (2002). *Acta Cryst.* **D58**, 2066–2075.
- Urzhumtseva, L. M. & Urzhumtsev, A. G. (1997). *J. Appl. Cryst.* **30**, 402–410.
- Urzhumtseva, L. & Urzhumtsev, A. (2002). *J. Appl. Cryst.* **35**, 644–647.