



Received 1 April 2021

Accepted 8 June 2021

Edited by T. J. Sato, Tohoku University, Japan

Keywords: quasicrystals; higher-dimensional crystallography.**Supporting information:** this article has supporting information at journals.iucr.org/j

PyQCstrc.ico: a computing package for structural modelling of icosahedral quasicrystals

Tsunetomo Yamada*

Faculty of Science, Department of Applied Physics, Tokyo University of Science, 6-3-1 Niijuku, Katsushika-ku, Tokyo 125-8585, Japan. *Correspondence e-mail: tsunetomo.yamada@rs.tus.ac.jp

The atomic structure of quasicrystals (QCs) is described as a section of a higher-dimensional structure that consists of a periodic arrangement of occupation domains (ODs). Determination of the shape of ODs and their partitioning is crucial in the structural analysis of QCs. However, owing to the complicated shape of ODs, building the initial structure model requires a great deal of time and effort. Thus, a computer program for building structure models of QCs is needed. Presented here is a Python3 package for structure modelling of icosahedral QCs.

1. Introduction

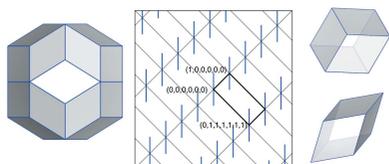
Quasicrystals (QCs) are long-range-ordered materials that generate self-similar diffraction patterns related to their scale incompatible with translational symmetry (Shechtman *et al.*, 1984; Levine & Steinhardt, 1984). The atomic structure of QCs can be described as a 3D section of n -dimensional (nD) periodic structures ($n > 3$). The nD structure consists of 'occupation domains' (ODs) or atomic surfaces whose geometrical shapes are defined in the $(n - 3)D$ complementary space called perpendicular space (\mathbf{E}_{\perp}), perpendicular to the 3D real space, called parallel space (\mathbf{E}_{\parallel}) (see *e.g.* Yamamoto, 1996; Janssen *et al.*, 2007). The ODs form 2D surfaces or 3D solids for dihedral ($n = 5$) or icosahedral ($n = 6$) QCs, respectively.

Structural analysis of QCs can be executed by utilizing the software package *QUASI* developed by Yamamoto (2008). This contains several programs which can be utilized for the direct method (*lodemac*) (Takakura *et al.*, 2001), structure refinement (*qcdiff*), generation of atomic positions in \mathbf{E}_{\parallel} (*qcstrc*), and generation of Fourier and maximum entropy method (MEM) maps (*qcmem*) *etc.* However, a tool for building the initial structure model in the nD space has been lacking, which means scientists have to put a great deal of work into building the initial models. To overcome this difficulty, the *PyQCstrc* library for the Python3 programming language was developed, with the aim of providing a tool for building nD models. In this article, I present the *PyQCstrc.ico* package in the library, which is dedicated to QCs with an icosahedral symmetry.

This article is organized as follows. In Section 2 I briefly describe the ODs in the 6D models for two 3D quasi-periodic tilings. In Section 3 I describe the Python package *PyQCstrc.ico*, which can be used to build and arrange the ODs in a 6D icosahedral lattice. In Section 4 an example is given, and a summary is provided in Section 5.

2. Occupation domains

In the higher-dimensional description of icosahedral quasicrystals (iQCs), the 6D model is described by a 6D icosahedral



OPEN ACCESS

lattice which is decorated by 3D ODs embedded in \mathbf{E}_\perp . The 6D positional vector $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$ can be represented by using the unit vectors of the 6D icosahedral lattice, \mathbf{d}_i ($i = 1, 2, \dots, 6$) (see Appendix A). Note that the choice of the unit vectors is arbitrary, and there are several conventions. The most frequently used are the conventions proposed by Cahn *et al.* (1986), Elser (1985) and Yamamoto (1996), and Yamamoto's is used in the package. The atomic positions are obtained from periodically arranged ODs by taking the intersections with \mathbf{E}_\parallel . Thus, the shape of ODs and their positions in the 6D space must be derived to build the initial model of the iQCs.

Fig. 1(a) shows the OD that generates the vertices of the 3D Amman–Kramer–Neri (AKN) tiling (Kramer & Neri, 1984; Duneau & Katz, 1985; Elser, 1985, 1986; Kalugin *et al.*, 1985). The OD forms a rhombic triacontahedron (RT), which corresponds to the projection of the unit cell of the 6D icosahedral lattice onto \mathbf{E}_\perp , and it is centred at either the origin $(0, 0, 0, 0, 0, 0)$ or the body centre $(1, 1, 1, 1, 1, 1)/2$, as shown in Fig. 1(b). Since the site symmetry of these positions is $m\bar{3}5$, the RT OD can be generated from its asymmetric unit by applying the 120 symmetry elements of $m\bar{3}5$. Here, the asymmetric unit forms a tetrahedron, and its Cartesian coordinates in \mathbf{E}_\parallel and \mathbf{E}_\perp are obtained from 6D vectors $(0, 0, 0, 0, 0, 0)$, $(1, \bar{1}, \bar{1}, \bar{1}, \bar{1}, \bar{1})/2$, $(1, \bar{1}, \bar{1}, 1, \bar{1}, \bar{1})/2$ and $(1, \bar{1}, \bar{1}, 0, \bar{1}, 0)/2$ by the projection operator (see Appendix A). The AKN tiling is composed of two building units, obtuse

and acute rhombohedra, shown in Figs. 1(c) and 1(d), respectively.

To decorate the 3D quasiperiodic tiling with atoms, a new OD must be introduced and arranged in the 6D icosahedral lattice. The shape and position of the OD can be derived by utilizing the basic OD for the 3D quasiperiodic tiling. More details about the atomic decorations can be found in the literature (Yamamoto, 1992, 1996; Quiquandon *et al.*, 2014). For example, the edge centre position of the AKN tiling is generated by the rhombic icosahedron (RI) shown in Fig. 2(a), centred at $(1, 0, 0, 0, 0, 0)/2$. The RI OD is obtained as a common part of two RT ODs at $(0, 0, 0, 0, 0, 0)$ and $(1, 0, 0, 0, 0, 0)$, when they are projected onto \mathbf{E}_\perp (Yamamoto, 1992).

The basic OD that is used for a realistic model becomes more complicated. For example, in the cluster-based model of primitive iQCs, the atomic clusters are arranged at the 12-fold packing sites of the AKN tiling (Yamamoto & Hiraga, 1988; Yamamoto, 1992; Takakura *et al.*, 2007; Quiquandon *et al.*, 2014). The cluster centre positions are generated by the truncated RT OD proposed by Henley (1986), shown in Fig. 2(b), centred at either $(0, 0, 0, 0, 0, 0)$ or $(1, 1, 1, 1, 1, 1)/2$, and the relationship between local configurations of the clusters and the partitioning of the OD can be found in the literature (Takakura, 2008; Takakura & Strzałka, 2017). The truncated RT can be generated from its asymmetric unit. The atomic decoration of the clusters can be obtained by arranging the basic truncated RT OD in the 6D icosahedral lattice. When the ODs projected onto \mathbf{E}_\perp intersect, in many cases the common part forms a concave polyhedron. One of the most difficult parts in building the initial model is related to the calculation of the common part.

3. Implementation and availability

In order to use *QUASI* for structural analysis, each OD must be defined as a set of tetrahedra, and the 6D positional vector \mathbf{x} whose perpendicular components assign each vertex of the constituent tetrahedra must be known. In the *PyQCstrc.ico* package, all vertices of a tetrahedron forming an OD are defined by 6D coordinates $(x_1, x_2, x_3, x_4, x_5, x_6)$, and the value of x_j ($j = 1, 2, \dots, 6$) is expressed as $(k_1 + \tau k_2)/k_3$, where k_1, k_2, k_3 are integers and τ is the golden ratio equal to $(1 + 5^{1/2})/2$.

The Cartesian coordinates in \mathbf{E}_\perp are given by $\tilde{Q}_\perp X$, where \tilde{Q}_\perp is the lower 3×6 part of the transposed matrix of Q [equation (2)] and X is a transposed matrix of $(x_1, x_2, x_3, x_4, x_5, x_6)$. The resulting Cartesian coordinates are represented by a list with integer components when the unit is taken to be $a(\tau + 2)^{-1/2}$. This representation allows one to perform numerical computation free from the influence of rounding error.

The package provides tools commonly used to build the initial structure of iQCs, which include translation and symmetry operations in the 6D space, and intersection operations on ODs in \mathbf{E}_\perp . In addition, export and import of the ODs are supported: the ODs can be exported in *VESTA*

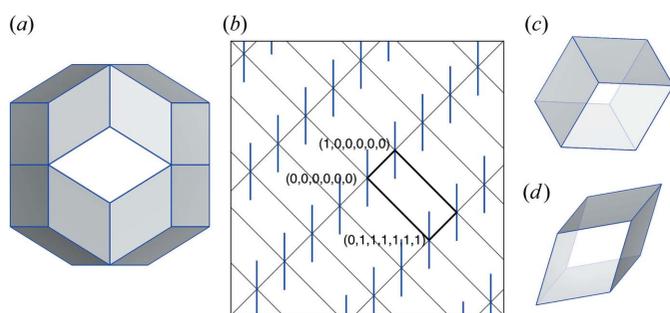


Figure 1
6D structure model of the AKN tiling. (a) The rhombic triacontahedron (RT) OD that generates the vertices of the tiling. (b) A fivefold section of the 6D structure. Blue bars and black thick lines indicate projection of the RT ODs and 6D unit cell onto the section, and the horizontal and the vertical of the figure are parallel to \mathbf{E}_\parallel and \mathbf{E}_\perp , respectively. Two building units of the AKN tiling, (c) obtuse and (d) acute rhombohedra.

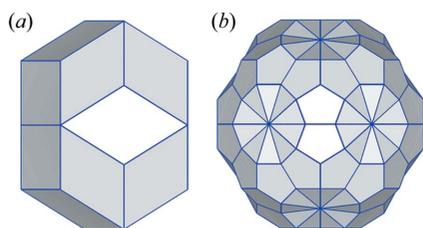


Figure 2
(a) Rhombic icosahedron OD that generates the edge centre position of the AKN tiling. (b) A truncated RT OD that generates the 12-fold packing sites of the AKN tiling. The truncated RT OD in (b) is enlarged by $1 + \tau$, where τ is the golden ratio equal to $(1 + 5^{1/2})/2$.

format (.vesta) and XYZ format (.xyz), and these can be visualized by utilizing VESTA (Momma & Izumi, 2011). The latter can be imported so that users can recall their ODs in Python scripts. Moreover, the package supports the QUASI formats (.pod and .atm) (Yamamoto, 2008), as described in the supporting information.

In realistic models of iQCs, the ODs may be concave polyhedra, and their tetrahedralization is not straightforward. In the package, the intersection operation of such ODs is performed by considering every pair of two tetrahedra included in each OD. Because the intersection of two tetrahedra forms a convex polyhedron, its tetrahedralization can be achieved by 3D Delaunay triangulation. Then, the common part is obtained as the union of the resulting tetrahedra. In the case where the common part forms a convex polyhedron, such as the intersection of two convex polyhedra, the above process can be shortened, and the solution is obtained by 3D Delaunay triangulation.

The package is distributed in the Python Package Index (PyPI) software repository (<https://pypi.org/project/pyqcstrc/>) and is freely available via `pip install pyqcstrc`. The package requires two libraries, *numpy* (<https://numpy.org/>) and *scipy* (<https://www.scipy.org/>), and the Cython programming language (<https://cython.org/>). Moreover, it is recommended that one uses Python3.7 or greater built with the SSL module, *OpenSSL* (<https://www.openssl.org/>). The package was developed and tested under macOS (10.14), and it was also tested on other operating systems, including Linux (Ubuntu20.04, Fedora32 and CentOS8) and Windows 10.

4. Example

In this section, the basic usage of the *PyQCstrc.ico* package is described by providing a procedure in a Python script to obtain the RI OD [Fig. 1(b)]. We start with the asymmetric unit of the basic RT OD [Fig. 1(a)]. This forms a tetrahedron whose vertices are assigned 6D coordinates as mentioned before. As seen below, the *j*th component of the 6D coordinates ($j = 1, 2, \dots, 6$), $(k_1 + \tau k_2)/k_3$, is expressed by utilizing a list $[k_1, k_2, k_3]$. The RI OD can be calculated in the following way:

(i) Creating the asymmetric unit of the basic RT OD:

```
# import libraries
import numpy as np
import pyqcstrc.ico.occupation_domain as od
# Vertices of tetrahedron, v0,v1,v2,v3, which
# defines the asymmetric unit.
v0 = np.array([[ 0, 0, 1],[ 0, 0, 1],[ 0, 0, 1],
               [ 0, 0, 1],[ 0, 0, 1],[ 0, 0, 1]])
v1 = np.array([[ 1, 0, 2],[-1, 0, 2],[-1, 0, 2],
               [-1, 0, 2],[-1, 0, 2],[-1, 0, 2]])
v2 = np.array([[ 1, 0, 2],[-1, 0, 2],[-1, 0, 2],
               [ 1, 0, 2],[-1, 0, 2],[-1, 0, 2]])
v3 = np.array([[ 1, 0, 2],[-1, 0, 2],[-1, 0, 2],
               [ 0, 0, 2],[-1, 0, 2],[ 0, 0, 2]])
od0 = np.vstack([v0,v1,v2,v3]).reshape(1,4,6,3)
```

Alternatively, the asymmetric part predefined in a file (rt_asymmetric.xyz) provided in the supporting information can be loaded as od0:

```
od0 = od.read_xyz(path='/dir/of/xyz', basename='rt_asymmetric')
```

Here, '/dir/of/xyz' indicates a directory where the rt_asymmetric.xyz file is located.

(ii) Creating the RT OD from its asymmetric unit by applying the symmetry elements of $m\bar{3}5$ around the origin:

```
# generate symmetric OD at the symmetric centre v0.
od1 = od.symmetric(od0, v0)
```

(iii) Creating the RT OD centred at (1, 0, 0, 0, 0, 0):

```
# 6D coordinates of position_1
pos1 = np.array([[ 1, 0, 1],[ 0, 0, 1],[ 0, 0, 1],
                 [ 0, 0, 1],[ 0, 0, 1],[ 0, 0, 1]])
od2 = od.shift(od1,pos1) # move to position_1
```

(iv) Intersection operation on od1 and od2:

```
# import libraries
import pyqcstrc.ico.two_occupation_domains as ods
od3 = ods.intersection(od1, od2, verbose=0)
```

(v) Export the resulting RI OD (od3) in VESTA and XYZ formats:

```
od.write_vesta(od3, path='/dir/of/work', basename='od3')
od.write_xyz(od3, path='/dir/of/work', basename='od3')
```

Here, '/dir/of/work' indicates a working directory.

(vi) od3 can be imported from the od3.xyz file:

```
od3 = od.read_xyz(path='/dir/of/work', basename='od3')
```

The calculation takes several minutes. To output the progress on the terminal, users can turn on verbose output by specifying an argument 'verbose' set to be 1 (verbose = 1) or larger.

Since the RI OD (od3) forms a convex polyhedron, it is tetrahedralizable by 3D Delaunay triangulation. In this case, `ods.intersection_convex()` can be used, instead of (iv), as follows:

```
od3 = ods.intersection_convex(od1, od2, verbose=0)
```

Moreover, the calculation cost can be greatly reduced by considering the asymmetric unit of the position where the solution is located. Here, a portion of od1 inside the asymmetric unit of $\bar{5}m$ at $(1, 0, 0, 0, 0, 0)/2$ is considered, and obtained as below:

```
# Three vectors w1,w2,w3, which defines the asym-
# etric unit.
w1 = np.array([[ 1, 0, 2],[-1, 0, 2],[-1, 0, 2],
               [-1, 0, 2],[-1, 0, 2],[-1, 0, 2]])
w2 = np.array([[ 0, 0, 1],[-1, 0, 4],[-1, 0, 4],
               [ 1, 0, 2],[-1, 0, 2],[ 1, 0, 2]])
w3 = np.array([[ 0, 0, 1],[ 1, 0, 4],[-1, 0, 2],
               [ 1, 0, 2],[-1, 0, 2],[ 1, 0, 4]])
vec_aum5 = np.vstack([w1,w2,w3]).reshape(3,6,3)
# 6D coordinates of edge-centre site.
posEC = np.array([[ 1, 0, 2],[ 0, 0, 1],[ 0, 0, 1],
                  [ 0, 0, 1],[ 0, 0, 1],[ 0, 0, 1]])
# The portion of rtod1.
od1_asym = od.asymmetric(od1, posEC, vec_aum5)
```

Then, the asymmetric unit of od3 can be obtained by the following:

```
od3_asym = ods.intersection(od1_asym, od2, verbose=0)
```

In the above expression, the asymmetric unit of $\bar{5}m$ is defined by three vectors \mathbf{w}_1 , \mathbf{w}_2 and \mathbf{w}_3 . The projection of \mathbf{w}_1 onto \mathbf{E}_\perp is parallel to a fivefold axis given by the projection of $(1, 0, 0, 0, 0, 0)/2$, while the projections of \mathbf{w}_2 and \mathbf{w}_3 are

orthogonal to the fivefold axis. They are on different mirror planes which make an angle of 36° .

5. Summary and perspectives

The *PyQCstrc.ico* package provides tools commonly used to build the initial structure models of iQCs in the Python3 programming language. This is particularly crucial for realistic structure models where the overall shape of ODs is quite complicated. In addition, the package is useful to check the closeness conditions of the ODs and to specify atomic pairs related to phason flips. The latter can be used to compute the short-range-order diffuse scattering, based on the theory proposed by Yamamoto (2010a,b).

The *PyQCstrc.ico* package is part of the *PyQCstrc* library. Two other packages dedicated to decagonal and dodecagonal QCs are scheduled to be included in the future.

APPENDIX A Coordinate system

The coordinate system used in this study is based on the one described in the literature (Yamamoto, 1996). The unit vectors of the 6D icosahedral lattice, \mathbf{d}_i ($i = 1, 2, \dots, 6$), are written using unit vectors in 3D \mathbf{E}_{\parallel} , \mathbf{a}_1 , \mathbf{a}_2 , \mathbf{a}_3 , and 3D \mathbf{E}_{\perp} , \mathbf{a}_4 , \mathbf{a}_5 , \mathbf{a}_6 , as

$$\mathbf{d}_i = \sum_{j=1}^6 Q_{ij} \mathbf{a}_j \quad (1)$$

with

$$Q_{ij} = \frac{a}{(\tau + 2)^{1/2}} \begin{bmatrix} 1 & \tau & 0 & \tau & -1 & 0 \\ \tau & 0 & 1 & -1 & 0 & \tau \\ \tau & 0 & -1 & -1 & 0 & -\tau \\ 0 & 1 & -\tau & 0 & \tau & 1 \\ -1 & \tau & 0 & -\tau & -1 & 0 \\ 0 & 1 & \tau & 0 & \tau & -1 \end{bmatrix}, \quad (2)$$

where a is the lattice parameter of the icosahedral lattice.

Acknowledgements

The author is grateful to A. Yamamoto for comments on the manuscript and for testing the *PyQCstrc.ico* package, and to H. Takakura for testing the package.

Funding information

Funding for this research was provided by Japan Society for the Promotion of Science (grant No. JP18K13987; grant No. JP19H05818).

References

- Cahn, J. W., Shechtman, D. & Gratias, D. (1986). *J. Mater. Res.* **1**, 13–26.
- Duneau, M. & Katz, A. (1985). *Phys. Rev. Lett.* **54**, 2688–2691.
- Elser, V. (1985). *Phys. Rev. B*, **32**, 4892–4898.
- Elser, V. (1986). *Acta Cryst.* **A42**, 36–43.
- Henley, C. L. (1986). *Phys. Rev. B*, **34**, 797–816.
- Janssen, T., Chapuis, G. & De Boissieu, M. (2007). *Aperiodic Crystals: From Modulated Phases to Quasicrystals*. Oxford University Press.
- Kalugin, P., Kitaev, A. & Levitov, L. (1985). *JETP Lett.* **41**, 145–149.
- Kramer, P. & Neri, R. (1984). *Acta Cryst.* **A40**, 580–587.
- Levine, D. & Steinhardt, P. J. (1984). *Phys. Rev. Lett.* **53**, 2477–2480.
- Momma, K. & Izumi, F. (2011). *J. Appl. Cryst.* **44**, 1272–1276.
- Quiquandon, M., Portier, R. & Gratias, D. (2014). *Acta Cryst.* **A70**, 229–238.
- Shechtman, D., Blech, I., Gratias, D. & Cahn, J. W. (1984). *Phys. Rev. Lett.* **53**, 1951–1953.
- Takakura, H. (2008). *Philos. Mag.* **88**, 1905–1912.
- Takakura, H., Gómez, C. P., Yamamoto, A., De Boissieu, M. & Tsai, A. P. (2007). *Nat. Mater.* **6**, 58–63.
- Takakura, H., Shiono, M., Sato, T. J., Yamamoto, A. & Tsai, A. P. (2001). *Phys. Rev. Lett.* **86**, 236–239.
- Takakura, H. & Strzalka, R. (2017). *J. Phys. Conf. Ser.* **809**, 012002.
- Yamamoto, A. (1992). *Phys. Rev. B*, **45**, 5217–5227.
- Yamamoto, A. (1996). *Acta Cryst.* **A52**, 509–560.
- Yamamoto, A. (2008). *Sci. Technol. Adv. Mater.* **9**, 013001.
- Yamamoto, A. (2010a). *Acta Cryst.* **A66**, 384–393.
- Yamamoto, A. (2010b). *Acta Cryst.* **A66**, 372–383.
- Yamamoto, A. & Hiraga, K. (1988). *Phys. Rev. B*, **37**, 6207–6214.