Acta Cryst A FOUNDATIONS ADVANCES

ISSN 2053-2733

A space for lattice representation and clustering

Lawrence C. Andrews,[a]* Herbert J. Bernstein[b,c] and Nicholas K. Sauter[d]

[a]Ronin Institute, 9515 NE 137th Street, Kirkland, WA 98034-1820, USA, [b]Rochester Institute of Technology, c/o NSLS-II, Brookhaven National Laboratory, Upton, NY 11973-5000, USA, [c]Ronin Institute, c/o NSLS-II, Brookhaven National Laboratory, Upton, NY 11973-5000, USA, and [d]Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA. *Correspondence e-mail: lawrence.andrews@ronininstitute.org

Received 9 January 2019
Accepted 22 February 2019

Edited by A. Altomare, Institute of Crystallography - CNR, Bari, Italy

Keywords: unit-cell reduction; Delaunay; Delone; Niggli; Selling; clustering.

Algorithms for quantifying the differences between two lattices are used for Bravais lattice determination, database lookup for unit cells to select candidates for molecular replacement, and recently for clustering to group together images from serial crystallography. It is particularly desirable for the differences between lattices to be computed as a perturbation-stable metric, *i.e.* as distances that satisfy the triangle inequality, so that standard tree-based nearest-neighbor algorithms can be used, and for which small changes in the lattices involved produce small changes in the distances computed. A perturbation-stable metric space related to the reduction algorithm of Selling and to the Bravais lattice determination methods of Delone is described. Two ways of representing the space, as six-dimensional real vectors or equivalently as three-dimensional complex vectors, are presented and applications of these metrics are discussed. (Note: in his later publications, Boris Delaunay used the Russian version of his surname, Delone.)

1. Introduction

Andrews *et al.* (2019) discuss the simplification resulting from using Selling reduction as opposed to using Niggli reduction. Here we continue that discussion with information on the space of unit cells and the subspace of reduced cells as the six-dimensional space $\mathbf{S}^6$ of Selling inner products.

Algorithms for quantifying the differences among lattices are used for Bravais lattice determination, database lookup for unit cells to select candidates for molecular replacement, and recently for clustering to group together images from serial crystallography. For crystallography, there are many alternative representations to choose from as a basis for distance calculations. Andrews *et al.* (1980) discussed $\mathbf{V}^7$, a perturbation-stable space in which, using real- and reciprocal-space Niggli reduction, a lattice is represented by three cell edge lengths, three reciprocal cell edge lengths and the cell volume, which was proposed for cell database searches, but which has difficulties when used for lattice determination. Andrews & Bernstein (1988) discussed $\mathbf{G}^6$ that uses a modified metric tensor and a search through 25 alternative reduction boundary transforms (Gruber, 1973) to work in a satisfactory manner both for database searches and for lattice identification in the presence of experimental error. Andrews & Bernstein (2014) discussed sewing together regions of the fundamental region of $\mathbf{G}^6$ under Niggli reduction at 15 boundaries. Andrews *et al.* (2019) presented the simplest and fastest currently known representation of lattices as the six Selling scalars obtained from the dot products of the unit-cell

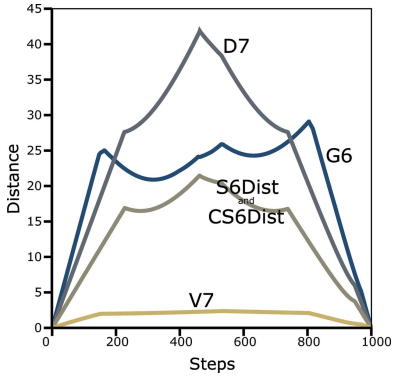

Table 1
The reflections in $\mathbf{S^6}$.

The 24 equivalent positions (Andrews *et al.*, 2019) in $\mathbf{S^6}$ as matrices, given in the same order as the reflections in Table 2.

[100000	010000	001000	000100	000010	000001]
[100000	001000	010000	000100	000001	000010]
[100000	000010	000001	000100	010000	001000]
[100000	000001	000010	000100	001000	010000]
[010000	100000	001000	000010	000100	000001]
[010000	001000	100000	000010	000001	000100]
[010000	000100	000001	000010	100000	001000]
[010000	000001	000100	000010	001000	100000]
[001000	100000	010000	000001	000100	000010]
[001000	010000	100000	000001	000010	000100]
[001000	000100	000010	000001	100000	010000]
[001000	000010	000100	000001	010000	100000]
[000100	010000	000001	100000	000010	001000]
[000100	001000	000010	100000	000001	010000]
[000100	000010	001000	100000	010000	000001]
[000100	000001	010000	100000	001000	000010]
[000010	100000	000001	010000	000100	001000]
[000010	001000	000100	010000	000001	100000]
[000010	000100	001000	010000	100000	000001]
[000010	000001	100000	010000	001000	000100]
[000001	100000	000010	001000	000100	010000]
[000001	010000	000100	001000	000010	100000]
[000001	000100	010000	001000	100000	000010]
[000001	000010	100000	001000	010000	000100]

axes in addition to the negative of their sum (a body diagonal). Labeling these **a**, **b**, **c** and **d** (**d** = −**a** −**b** −**c**), the scalars are

$$\mathbf{b}\cdot\mathbf{c}, \mathbf{a}\cdot\mathbf{c}, \mathbf{a}\cdot\mathbf{b}, \mathbf{a}\cdot\mathbf{d}, \mathbf{b}\cdot\mathbf{d}, \mathbf{c}\cdot\mathbf{d}$$

(where, *e.g.*, **b** · **c** represents the dot product of the **b** and **c** axes). For the purpose of organizing these six quantities as a vector space in which one can compute simple Euclidean distances, we describe the set of scalars as a vector, **s**, with components, $s_1, s_2, s_3, \ldots, s_6$. The cell is Selling reduced if all six components are negative or zero (Delone, 1933). Minimizing among distances computed from alternate paths between Selling-reduced cells with appropriate sewing at the six boundaries of the Selling-reduced fundamental region of $\mathbf{S^6}$ yields a computationally sound metric space within which to do lattice identification, cell database searching and serial crystallography clustering.

We define two equivalent spaces related to the Selling reduction: the space of six-dimensional real vectors, $\mathbf{S^6}$, or equivalently the space of three-dimensional complex vectors $\mathbf{C^3}$:

$$[\mathbf{b}\cdot\mathbf{c} + i\mathbf{a}\cdot\mathbf{d}, \mathbf{a}\cdot\mathbf{c} + i\mathbf{b}\cdot\mathbf{d}, \mathbf{a}\cdot\mathbf{b} + i\mathbf{c}\cdot\mathbf{d}]$$

or

$$[s1 + is4, s2 + is5, s3 + is6].$$

Although $\mathbf{S^6}$ and $\mathbf{C^3}$ simply reorganize the same data, some operations are simpler to visualize in one space than the other. In some cases, we will choose to show only the simpler one.

The objective of this paper is to explain how to compute the distances between lattices using $\mathbf{S^6}$ and $\mathbf{C^3}$.

2. The space S^6

For a Bravais tetrahedron (Bravais, 1850) with defining vectors **a**, **b**, **c**, **d** (the edge vectors of the unit cell plus the negative sum of them), a point in $\mathbf{S^6}$ is

$$[\mathbf{b}\cdot\mathbf{c}, \mathbf{a}\cdot\mathbf{c}, \mathbf{a}\cdot\mathbf{b}, \mathbf{a}\cdot\mathbf{d}, \mathbf{b}\cdot\mathbf{d}, \mathbf{c}\cdot\mathbf{d}].$$

A simple example is the orthorhombic unit cell (10, 12, 20, 90, 90, 90) (a, b, c, α, β, γ). The corresponding $\mathbf{S^6}$ vector is

$$[0, 0, 0, -100, -144, -400]. \quad (1)$$

The scalars in $\mathbf{S^6}$ are of a single type, unlike cell parameters (lengths and angles) and unlike $\mathbf{G^6}$ (squared lengths and dot products). Delone *et al.* (1975) state '*The Selling parameters are geometrically fully homogeneous*'.

Because there is no crystallographic reason to favor one ordering of **a**, **b**, **c**, **d** over another, for any given Selling-reduced cell there are 24 fully equivalent presentations as $\mathbf{S^6}$ vectors generated by the $4 \times 3 \times 2 \times 1 = 24$ possible permutations of **a**, **b**, **c**, **d** (Andrews *et al.*, 2019). To compute a distance between two different Selling-reduced cells, the least we will need to do is to compute the minimum of the distances between one of the cells and the 24 possible permutations of the other (Andrews *et al.*, 2019).

In addition, because Selling-reduced cells are defined as having only zero or negative scalars, the space has boundaries at the transitions to positive scalars. Therefore, if either of the two different Selling-reduced cells is in the vicinity of a boundary, we also need to consider the path changes that may arise from the reduction steps at that boundary. Additional, lower-dimension boundaries may be implied when scalars have equal values, but explicit consideration of those in addition to the permutations and sign-transition boundary transformations does not appear to be needed.

Some of the properties of $\mathbf{S^6}$ are simple. The six base axes are orthogonal, unlike those of $\mathbf{G^6}$ (Andrews & Bernstein, 2014). For example, the matrix projecting onto the s_2 axis is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

and the matrix projecting onto the five-dimensional polytope (the 'perp') spanned by $s_1, 0, s_3, s_4, s_5, s_6$ orthogonal to the s_2 axis at $s_2 = 0$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Table 2
The reflections in $\mathbf{C^3}$.

The 24 equivalent positions (Andrews *et al.*, 2019) in $\mathbf{C^3}$ as permuations and real–imaginary exchanges ($\mathfrak{X}$), given in the same order as the equivalent matrices in Table 1.

[c_1,	c_2,	c_3]	$= [s_1, s_2, s_3]$	$+i[s_4, s_5, s_6]$
[c_1,	c_3,	c_2]	$= [s_1, s_3, s_2]$	$+i[s_4, s_6, s_5]$
[c_1,	$\mathfrak{X}(c_2)$,	$\mathfrak{X}(c_3)$]	$= [s_1, s_5, s_6]$	$+i[s_4, s_2, s_3]$
[c_1,	$\mathfrak{X}(c_3)$,	$\mathfrak{X}(c_2)$]	$= [s_1, s_6, s_5]$	$+i[s_4, s_3, s_2]$
[c_2,	c_1,	c_3]	$= [s_2, s_1, s_3]$	$+i[s_5, s_4, s_6]$
[c_2,	c_3,	c_1]	$= [s_2, s_3, s_1]$	$+i[s_5, s_6, s_4]$
[c_2,	$\mathfrak{X}(c_1)$,	$\mathfrak{X}(c_3)$]	$= [s_2, s_4, s_6]$	$+i[s_5, s_1, s_3]$
[c_2,	$\mathfrak{X}(c_3)$,	$\mathfrak{X}(c_1)$]	$= [s_2, s_6, s_4]$	$+i[s_5, s_3, s_1]$
[c_3,	c_1,	c_2]	$= [s_3, s_1, s_2]$	$+i[s_6, s_4, s_5]$
[c_3,	c_2,	c_1]	$= [s_3, s_2, s_1]$	$+i[s_6, s_5, s_4]$
[c_3,	$\mathfrak{X}(c_1)$,	$\mathfrak{X}(c_2)$]	$= [s_3, s_4, s_5]$	$+i[s_6, s_1, s_2]$
[c_3,	$\mathfrak{X}(c_2)$,	$\mathfrak{X}(c_1)$]	$= [s_3, s_5, s_4]$	$+i[s_6, s_2, s_1]$
[$\mathfrak{X}(c_1)$,	c_2,	$\mathfrak{X}(c_3)$]	$= [s_4, s_2, s_6]$	$+i[s_1, s_5, s_3]$
[$\mathfrak{X}(c_1)$,	c_3,	$\mathfrak{X}(c_2)$]	$= [s_4, s_3, s_5]$	$+i[s_1, s_6, s_2]$
[$\mathfrak{X}(c_1)$,	$\mathfrak{X}(c_2)$,	c_3]	$= [s_4, s_5, s_3]$	$+i[s_1, s_2, s_6]$
[$\mathfrak{X}(c1)$,	$\mathfrak{X}(c3)$,	$c2$]	$= [s_4, s_6, s_2]$	$+i[s_1, s_3, s_5]$
[$\mathfrak{X}(c_2)$,	c_1,	$\mathfrak{X}(c_3)$]	$= [s_5, s_1, s_6]$	$+i[s_2, s_4, s_3]$
[$\mathfrak{X}(c_2)$,	c_3,	$\mathfrak{X}(c_1)$]	$= [s_5, s_3, s_4]$	$+i[s_2, s_6, s_1]$
[$\mathfrak{X}(c_2)$,	$\mathfrak{X}(c_1)$,	c_3]	$= [s_5, s_4, s_3]$	$+i[s_2, s_1, s_6]$
[$\mathfrak{X}(c_2)$,	$\mathfrak{X}(c_3)$,	c_1]	$= [s_5, s_6, s_1]$	$+i[s_2, s_3, s_4]$
[$\mathfrak{X}(c_3)$,	c_1,	$\mathfrak{X}(c_2)$]	$= [s_6, s_1, s_5]$	$+i[s_3, s_4, s_2]$
[$\mathfrak{X}(c_3)$,	c_2,	$\mathfrak{X}(c_1)$]	$= [s_6, s_2, s_4]$	$+i[s_3, s_5, s_1]$
[$\mathfrak{X}(c_3)$,	$\mathfrak{X}(c_1)$,	c_2]	$= [s_6, s_4, s_2]$	$+i[s_3, s_1, s_5]$
[$\mathfrak{X}(c_3)$,	$\mathfrak{X}(c_2)$,	c_1]	$= [s_6, s_5, s_1]$	$+i[s_3, s_2, s_4]$

2.1. The reflections in S^6

The 24 equivalent positions (Andrews *et al.*, 2019) in $\mathbf{S^6}$ have corresponding matrices designed to act on $\mathbf{S^6}$ vectors to map them into crystallographically equivalent vectors. For convenience, they are all listed in Table 1. The structure of the set is clearer in $\mathbf{C^3}$. See Table 2, which presents the reflections in the same order.

The unsorted nature of Selling reduction implies that distance calculations will need to consider the reflections. Even if a usable sorting of points in the fundamental unit were created, at least some of the reflections would still be required for near-boundary cases.

2.2. Reduction in S^6

Lattice reduction is quite simple in $\mathbf{S^6}$ (Andrews *et al.*, 2019), but it has a clearer structure in $\mathbf{C^3}$, so it will be treated there (Section 3.2). Because of the simple nature of $\mathbf{S^6}$, the inverse of each reduction operation is the same as the unreduction operation, so we term them edge transforms. The matrices in $\mathbf{S^6}$ are unitary, so the metric is the same in each region. However, the transformation matrices are not diagonal, with the result that the boundaries are not simple mirrors.

We present the edge transforms as matrices, two for each scalar; the second line for each is the alternate choice of which pair to exchange [copied from Andrews *et al.* (2019)].

For the $\mathbf{b} \cdot \mathbf{c} = 0$ boundary

$$[\bar{1}00000/110000/100010/\bar{1}00100/101000/100001]$$

or

$$[\bar{1}00000/100001/101000/\bar{1}00100/100010/110000].$$

For the $\mathbf{a} \cdot \mathbf{c} = 0$ boundary

$$[110000/0\bar{1}0000/010100/011000/0\bar{1}0010/010001]$$

or

$$[010001/0\bar{1}0000/011000/010100/0\bar{1}0010/110000].$$

For the $\mathbf{a} \cdot \mathbf{b} = 0$ boundary

$$[101000/001100/00\bar{1}000/011000/001010/00\bar{1}001]$$

or

$$[001010/011000/00\bar{1}000/001100/101000/00\bar{1}001].$$

For the $\mathbf{a} \cdot \mathbf{d} = 0$ boundary

$$[100\bar{1}00/001100/010100/000\bar{1}00/000110/000101]$$

or

$$[100\bar{1}00/010100/001100/000\bar{1}00/000101/000110].$$

For the $\mathbf{b} \cdot \mathbf{d} = 0$ boundary

$$[001010/0100\bar{1}0/100010/000110/0000\bar{1}0/000011]$$

or

$$[100010/0100\bar{1}0/001010/000011/0000\bar{1}0/000110].$$

For the $\mathbf{c} \cdot \mathbf{d} = 0$ boundary

$$[010001/100001/00100\bar{1}/000101/000011/00000\bar{1}]$$

or

$$[100001/010001/00100\bar{1}/000011/000101/00000\bar{1}].$$

2.3. The boundaries in S^6

The first type of boundary in $\mathbf{S^6}$ is the polytope where one of the six axes is zero. [Contrast this with $\mathbf{G^6}$ (Andrews & Bernstein, 2014), which has 15 boundaries of several types.] Obviously, the zeros correspond to unit-cell angles of 90°. In $\mathbf{S^6}$, the zeros mark the regions where components change from negative to positive, *i.e.* the place where cells become non-Selling reduced. A second kind of boundary is where certain 'opposite' pairs of scalars are equal; this is more easily visualized in $\mathbf{C^3}$ where those pairs are just the real and imaginary parts of one complex scalar. These are handled as 'reflections' (see Sections 2.1 and 3.1).

The consequence for distance calculations will be that the reduction operations will be involved in the distance computations.

3. The space C^3

Alternatively, the space $\mathbf{S^6}$ can be as represented as $\mathbf{C^3}$, a space of three complex axes. $\mathbf{C^3}$ has advantages for understanding some of the properties of the space. When we compose $\mathbf{S^6}$ of the scalars $s_1, \ldots, s_6$, the components of $\mathbf{C^3}$ are the pairs of 'opposite' (Delone *et al.*, 1975) scalars. In terms of the elements of $\mathbf{S^6}$, a unit cell in $\mathbf{C^3}$ is $[s_1 + is_4, s_2 + is_5, s_3 + is_6]$.

The $\mathbf{C}^3$ presentation of the vector (1) from Section 2 is $[-100i, -144i, -400i]$.

3.1. The reflections in C^3

The 24 reflections of the scalars correspond to 24 reflection operations in $\mathbf{C}^3$. First, any pair of $\mathbf{C}^3$ coordinates may be exchanged. The other reflection operation is the exchange of the real and imaginary parts of each member of any pair of $\mathbf{C}^3$ coordinates. We use $\mathfrak{X}$ (for 'eXchange') to denote this operation. For example, $c_1 = c_{1,r} + ic_{1,i}$ and $c_3 = c_{3,r} + ic_{3,i}$ can transform to $\mathfrak{X}(c_1) = c_{1,i} + ic_{1,r}$ and $\mathfrak{X}(c_3) = c_{3,i} + ic_{3,r}$. For complex numbers such an exchange can be effected by taking the complex conjugate and multiplying by i, so $\mathfrak{X}(c) = i\overline{c}$.

Combining the exchange operation with the coordinate interchanges in all possible combinations gives the 24 reflections (including the identity).

Representing the operation of interchanging the real and imaginary parts of a complex number by $\mathfrak{X}$, the 24 reflections in $\mathbf{C}^3$ as permutations of $[c_1, c_2, c_3]$ are given in Table 2.

3.2. Reduction in C^3

In $\mathbf{C}^3$, reduction has a more ordered form than in $\mathbf{S}^6$. Consider a general point in $\mathbf{C}^3$ with components c_a, c_n, c_x. For descriptive purposes, let us assume that the imaginary part of c_n is the sole positive scalar, the one we must reduce.

Step 1: subtract the imaginary part of c_n from the real part and change the imaginary part of c_n to its negative value.

Step 2: add the original value of the imaginary part of c_n to the real and the imaginary parts of c_a and c_x.

Step 3: exchange the real part of c_a with the imaginary part of c_x. (The alternative choice of exchanging the real part of c_a with the imaginary part of c_x is also valid.)

The reduction operations do not commute, which will add complexity to distance calculations (see Section 4 below). The two choices are related by one of the reflection operations. For distance calculations, all of the reflections must be considered, so the choice will not matter in the end.

3.3. An asymmetric unit in C^3

The fundamental unit in $\mathbf{S}^6$ and $\mathbf{C}^3$ is chosen to be the region where all six scalars are zero or negative. However, there are 24 representations of a general point in that orthant. $\mathbf{C}^3$ provides the possibility of choosing a particular region of the fundamental unit as the asymmetric unit where there is only a single representation of the general point (similar to an asymmetric unit in a space group).

The three components can be sorted by their magnitude. The second step is to exchange the real and imaginary parts of c_1 so that the real part is less than or equal to the imaginary part (if necessary); that requires also exchanging c_2 or c_3. Finally, c_2 has its real and imaginary parts exchanged if necessary and of course those of c_3 also. Note that the ordering of the real and imaginary parts of c_3 is not defined.

$\mathbf{S}^6$ does not provide a comparable simple suggestion for an asymmetric unit with a single unique representation of each lattice, except by converting to $\mathbf{C}^3$ and back.

4. Measuring distance

We require a distance metric that defines the shortest path among all the representations of two points (lattices). Common uses of a metric for lattices are searching in databases of unit-cell parameters, finding possible Bravais lattice types, locating possible suitable molecular replacement candidates and, recently, clustering of the images from serial crystallography.

A simple example of the complexity of the task is that we must decide which of the 24 reflections of one of the points is the closest to the other point. Using the reduction operations so that other paths are examined is also required. That the reduction operations do not commute means that the order of operations may in some cases be important.

It is also important to note that the necessary examination of reflections in calculating a distance may undo any time savings achieved by identification of unique cells in an asymmetric unit, so it is usually better to work in the full fundamental unit, rather than restricting our attention to the asymmetric unit. In the current work, the full fundamental unit is always considered.

The non-diagonal nature of reduction operations in this space means that measuring the distance between points in different regions of space is not as simple as finding the Cartesian distance. The edge-transform matrices transform a point in the fundamental unit to another, non-reduced unit, one where one scalar is positive. (Continued applications of the matrices will generate one or two more positive scalars.) Because of the non-diagonal nature of the matrices, the metric direction will change between each unit. The simple Euclidean distance from a point in the fundamental unit to one in another unit is not necessarily the minimal distance. A path broken by reflections and reduction transformations may be shorter. We present two alternative algorithms that do find a valid minimal distance. See Sections 4.1 and 4.2.

4.1. Measuring distance: virtual Cartesian points (VCPs)

4.1.1. Creating virtual Cartesian points. For a point and a chosen operator for the reduction, we separate the point into two vectors: the projection onto the polytope for which the reduction axis is zero and the perp, the projection onto the reduction axis. The reduction operation is applied to the boundary-projected vector, and then the negative of the perp is added to that result. We call that resulting point the VCP (see Fig. 1). The goal of creating a VCP is that in measuring distances to points in the fundamental unit one can use the Euclidean metric of the fundamental unit.

4.1.2. Using VCPs to determine distance. To begin, the six VCPs (one for each boundary) are computed for the first of the two input points. Then the 24 reflections are computed for those six results plus the initial point itself. The desired

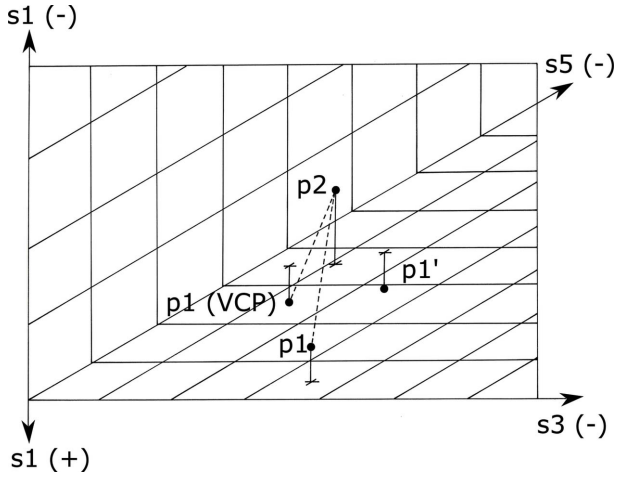

Figure 1
Example of a one-boundary virtual Cartesian point distance calculation. Only the VCP operation is shown, no reflections. Both points p_1 and p_2 are Selling reduced. The image is the three-dimensional, all-negative octant of the three $\mathbf{S^6}$ axes, s_1, s_3 and s_5; the reduction is done along the s_1 axis, and s_3 and s_5 are the two scalars that will be interchanged. The points are shown above or below the s_3/s_5 plane, with their projections onto that plane marked with a +. To compute the minimal distance between p_1 and p_2, begin by computing the Euclidean distance between the two. The s_1 reduction transforms p_1 into p_1', but the metric changes when going from negative s_1 to positive s_1, so the simple Euclidean distance may not be minimal. To generate p_1(VCP) to which the distance may be shorter, project p_1 onto the s_3/s_5 plane, transform that projected point, and subtract s_1 from that point. The distance from p_1 to p_1(VCP) can now be used to decide whether it is shorter than the $p_1 - p_2$ Euclidean distance. The best distance for this case is the shorter of the distances between p_1 and p_2 as opposed to the distance between p_1(VCP) and p_2.

distance is the minimum of the distances between the second point and all of the 168 points created in the first step. This is a one-boundary case. Monte Carlo experiments show that fewer than 1% of the minimal distances can be improved by two-boundary solutions and in most cases the difference is less than 10% (see Fig. 2).

Two-boundary solutions are created by first generating the same 168 points as in the-one boundary solution. Then we generate the six VCPs of the second input point and find the minimal distance between the 168 versus the first point and the seven points consisting of the six VCPs and the first point.

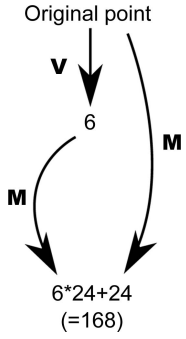

Figure 2
In this figure, **M** represents the application of all 24 reflection matrices. **V** represents the generation of six virtual Cartesian points from an input point.

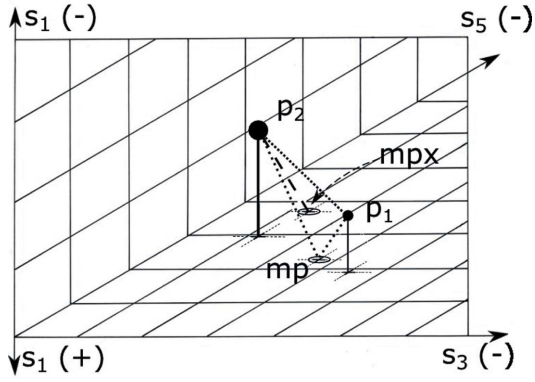

Figure 3
This an example of a one-boundary tunneled mirrored boundary distance calculation. As with Fig. 1 the 24 reflections are not shown. Both points p_1 and p_2 are Selling reduced. The image is the three-dimensional, all-negative octant of the three $\mathbf{S^6}$ axes, s_1, s_3, and s_5; the reduction is done along the s_1 axis, and s_3 and s_5 are the two scalars that will be interchanged. The points are shown above the s_3/s_5 plane, with their projections onto that plane marked with a circled 'X'. The Euclidean distance from p_1 to p_2 is shown as a dotted line. Let *mp* be the mirror point on the boundary going from p_1 to p_2 via the boundary. Then the shortest distance from p_1 to *mp* to p_2 is also shown as a dotted line. The transformed image of *mp* is *mpx*. The distance between p_1 and *mp* is the same as the distance between a transformed p_1 and *mpx*. There is a no-cost tunnel from *mp* to *mpx*. So the total alternative distance for this case is the distance between p_1 and *mp* plus the distance from *mpx* to p_2 (shown as a dashed line).

4.2. Measuring distance: tunneled mirrored boundaries

An alternative to computing VCPs outside the fundamental unit is to compute mirror points in the boundaries and to tunnel between them with the boundary transformations. Start with points p_1 and p_2 and one boundary *bd*, with projector P_{bd}; *e.g.* in Fig. 3 the *bd* is $s_1 = 0$. A simple mirror for a path from p_1 to *bd* and then to p_d can be constructed from the hypotenuses of the two right triangles with heights equal to the distances from p_1 to *bd* and p_2 to *bd*, respectively, and legs made by dividing the line from $P_{bd}(p_1)$ to $P_{bd}(p_2)$ in the same proportions. Shorter paths may result by replacing the simple mirror point *mbd* with its image *mbd*′ under a boundary transformation and applying the 24 reflections both to the mirror point and to its transformation.

More general tunneling of this type is possible using two boundaries bd_{dwn} with projector $P_{bd_{dwn}}$, and bd_{up} with projector $P_{bd_{up}}$

4.3. Measuring distance: example

In order to verify the correctness and completeness of the implementations of distance algorithms, the '*Follower*' algorithm was developed. It is implemented in the program *PointDistanceFollower*. Two points are chosen, a line constructed between them and then distances are calculated from each point along the line to the final point. One of the choices in the program is to make the final point be the reduced point of the starting point. The program also provides timing for the various options (see Fig. 4). Several criteria for quality control can be applied, such as: zero distance at both

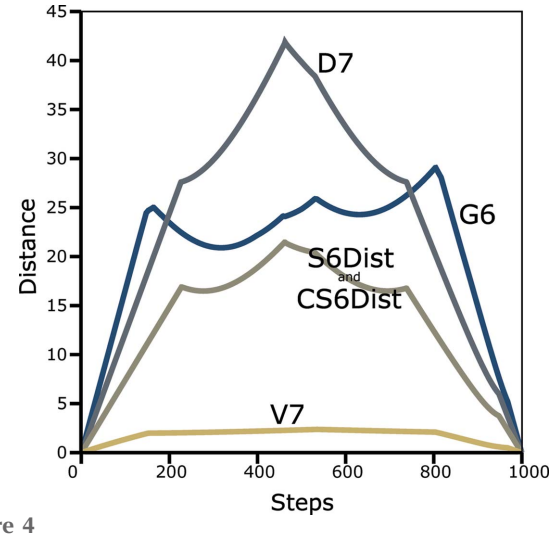

Figure 4
Distance between points using the *Follower* algorithm. To verify the distance algorithms, the '*Follower*' algorithm has been developed. *Follower* chooses two points and determines the distance between one of them and all of the points on a line between the two original points. Here, one unreduced point is chosen and the second point is the reduced point of that point. So the distance between the original point and the final point is zero. Distances are shown for the $\mathbf{G}^6$ metric (Andrews & Bernstein, 2014), the $\mathbf{V}^7$ metric (Andrews *et al.*, 1980), the $\mathbf{D}^7$ metric (Andrews *et al.*, 2019), and the two implementations in $\mathbf{S}^6$. Timing in ms: $\mathbf{G}^6$ (NCDist) 4542, $\mathbf{D}^7$ 676, $\mathbf{V}^7$ 7, S6Dist 394, CS6Dist 14.

ends of the scan, continuity and only occasional discontinuities in slope (due to boundary crossings). This figure compares results from four metrics: $\mathbf{S}^6$, $\mathbf{G}^6$, $\mathbf{D}^7$ and $\mathbf{V}^7$.

It can be observed that the $\mathbf{V}^7$ metric as seen in Fig. 4 is both fast to compute and smooth, and that leads one to ask whether $\mathbf{V}^7$ should not be the favored metric. The issue seems to have not been described well in the literature. For crystallographic purposes, a smooth metric is not sufficient. We also need sensitivity to the differences among lattices, especially for clustering.

The $\mathbf{V}^7$ metric (Andrews *et al.*, 1980) was developed for the purpose of searching databases of unit-cell parameters. It was developed again by Rodgers & LePage (1992). The designation as $\mathbf{V}^7$ began in the work of Andrews & Bernstein (2014). The elements of the $\mathbf{V}^7$ metric are: the reduced cell lengths, the reciprocals of the edge lengths of the reduced reciprocal cell, and the cube root of the volume of the primitive cell. Note that this definition means that each element has the same units. Because the edge lengths of reduced cells are stable to pertubation (Andrews *et al.*, 1980) and the primitive unit-cell volume is an invariant of the lattice (Andrews & Bernstein, 1995), we can be assured that the $\mathbf{V}^7$ metric is stable to perturbation. In fact, this stability has led to systems where searches are only done using reduced cell edge lengths (and perhaps volume) (Mighell & Karen, 1996); obviously such searches have little to no sensitivity to angle differences.

The core problem with the $\mathbf{V}^7$ metric is that the sensitivity to angles decreases as angles approach 90°. This issue appears because of the definition of reciprocal cell parameters. For example, the reciprocal cell parameter a^* is defined as: $a^* = |b||c| \sin(\alpha)/V$, where V is the volume of the unit cell.

The issue that arises is that the sine function varies slowly in the neighborhood of 90°. Sensitivity to angle (cosine, the derivative of sine) approaches zero as the angle approaches 90°. Unfortunately for us, 90° angles are common in crystals, rendering $\mathbf{V}^7$ an insensitive metric for important regions.

Three issues can be seen immediately. First, if the derivatives are approaching zero, least-squares in $\mathbf{V}^7$ is likely to not perform well in some cases. Second, in the case of database searches, false-positive reports will be common. For example, Byram *et al.* (1996) explicitly describe the problem:

'*Algorithms are designed to ensure that no known unit cells are missed in the search. The output may sometimes present numerous candidates for a match, but this can be screened readily by the researcher and is not considered problematic since the search is done only once per new crystal studied*'.

Third, in clustering, the failure of $\mathbf{V}^7$ to distinguish lattices near 90° can prevent us from creating reasonably homogeneous clusters that can be distinguished with $\mathbf{S}^6$, $\mathbf{G}^6$ or $\mathbf{D}^7$. Of those three, $\mathbf{S}^6$ is the fastest.

5. Clustering

This is a time of disruptive change in the image-clustering methods used in structural biology to understand polymorphs and dynamics at X-ray free-electron lasers and at synchrotrons. Serial crystallography is an essential technique at X-ray free-electron laser (XFEL) light sources and has become an important technique at synchrotrons as well (Rossmann, 2014), especially at newer high-brilliance beamlines. Methods that distribute the many diffraction images into clusters that likely represent crystals composed of proteins in similar states allow one to separate polymorphs and to categorize their dynamics. The inexorable increases in brilliance of these sources drives us to seek continual improvement in our algorithms and pipelines.

Clustering based on cell parameters is effective at the early stages of clustering when dealing with partial data sets. Here the Andrews–Bernstein NCDist cell-distance method (Andrews & Bernstein, 2014) used by Zeldin *et al.* (2015) is effective. One might investigate other criteria such as differences of Wilson plots to measure similarities of data (Foadi *et al.*, 2013). When the original data are complete (>75% today for similar applications), or one wants to achieve higher levels of completeness, one can cluster on correlation of intensities (CC, which stands for 'correlation coefficient') (Bernstein *et al.*, 2017). Changing the space being used from $\mathbf{G}^6$ with NCDist to $\mathbf{S}^6$ provides a significant performance improvement.

While NCDist has been effective for clustering, the original implementation is very demanding of computational resources. The development of CS6Dist, a macro-based $\mathbf{S}^6$ cell distance method, has improved cluster timing, both indirectly for NCDist by first reducing with $\mathbf{S}^6$ before finishing with Niggli reduction, and directly by computing $\mathbf{S}^6$ distances in which only six boundaries need to be considered instead of $\mathbf{G}^6$ distances in which 15 boundaries need to be considered. Use of $\mathbf{S}^6$ distances results in identical or qualitatively very similar

dendrograms of cluster candidates obtained using $\mathbf{G}^6$. For example, the commonly used CCP4 clustering program *Blend* (Foadi *et al.*, 2013) has been modified to use $\mathbf{S}^6$ reduction and CS6Dist distances and tested on a set of 71 lysozyme 5° wedges from a slightly doped crystal, comparing NCDist and CS6Dist timing, on a 12-core, 24-thread AMD Ryzen threadripper system. The NCDist run took 28 s real time and 72 s user time. The CS6Dist run took 25 s real time and 40 s user time. The results were identical. This example and more challenging examples of the application of $\mathbf{S}^6$ in clustering will be discussed in more detail in a subsequent paper.

6. Summary

We have presented representations of a space (parameterized as $\mathbf{S}^6$ and $\mathbf{C}^3$) based on the Selling parameters and using the Selling reduction. Geometrically, this represents a significant simplification compared with the complex, non-convex asymmetric unit of Niggli reduction and $\mathbf{G}^6$.

Conceptually, there is simplification due to the orthogonal rather than inclined axes and single type of boundary of the reduced cell fundamental unit. Reasoning is simpler in such a Cartesian system. For one thing, there are fewer and simpler boundaries to the fundamental unit.

Distance calculations are faster in $\mathbf{S}^6$ than in $\mathbf{G}^6$. This is due to the simpler structure of the space which leads to simpler algorithms. Niggli reduction sorts the cell parameters, eliminating the 24-fold ambiguity that remains in Selling reduction. However, that advantage disappears when computing distances because it is still necessary to examine the same edge cases. Selling reduction saves time both for the reduction, and, more importantly, for the calculation of distances among lattices in lattice identification, in cell databases, and in cell clustering.

7. Availability of code

The C++ code for distance calculations in $\mathbf{S}^6$ is available using https://github.com/; for CS6Dist.h, use https://github.com/yayahjb/ncdist; for *PointDistanceFollower* (*Follower* implementation), S6Dist.h and .cpp, use https://github.com/duck10/LatticeRepLib.

Acknowledgements

Careful copy-editing and corrections by Frances C. Bernstein are gratefully acknowledged. Elizabeth Kincaid gave expert help with graphics. Our thanks to Jean Jakoncic and Alexei Soares for helpful conversations and access to data and facilities at Brookhaven National Laboratory.

Funding information

Funding for this research was provided in part by: US Department of Energy Offices of Biological and Environmental Research and of Basic Energy Sciences (grant No. DE-AC02-98CH10886; grant No. E-SC0012704); US National Institutes of Health (grant No. P41RR012408; grant No. P41GM103473; grant No. P41GM111244; grant No. R01GM117126); Dectris, Ltd.

References

Andrews, L. C. & Bernstein, H. J. (1988). *Acta Cryst.* A**44**, 1009–1018.
Andrews, L. C. & Bernstein, H. J. (1995). *Acta Cryst.* A**51**, 413–416.
Andrews, L. C. & Bernstein, H. J. (2014). *J. Appl. Cryst.* **47**, 346–359.
Andrews, L. C., Bernstein, H. J. & Pelletier, G. A. (1980). *Acta Cryst.* A**36**, 248–252.
Andrews, L. C., Bernstein, H. J. & Sauter, N. K. (2019). *Acta Cryst.* A**75**, 115–120.
Bernstein, H. J., Andrews, L. C., Foadi, J., Fuchs, M. R., Jakoncic, J., McSweeney, S., Schneider, D. K., Shi, W., Skinner, J., Soares, A. & Yamada, Y. (2017). *bioRxiv*, p. 141770.
Bravais, A. (1850). *J. Ec. Polytech.* pp. 1–128.
Byram, S. K., Campana, C. F., Fait, J. & Sparks, R. A. (1996). *J. Res. Natl Inst. Stand. Technol.* **101**, 295–300.
Delone, B. N. (1933). *Z. Kristallogr.* **84**, 109–149.
Delone, B. N., Galiulin, R. V. & Shtogrin, M. I. (1975). *J. Math. Sci.* **4**, 79–156.
Foadi, J., Aller, P., Alguel, Y., Cameron, A., Axford, D., Owen, R. L., Armour, W., Waterman, D. G., Iwata, S. & Evans, G. (2013). *Acta Cryst.* D**69**, 1617–1632.
Gruber, B. (1973). *Acta Cryst.* A**29**, 433–440.
Mighell, A. D. & Karen, V. L. (1996). *J. Res. Natl Inst. Stand. Technol.* **101**, 273.
Rodgers, J. & LePage, Y. (1992). *American Crystallographic Association Annual Meeting and Pittsburgh Diffraction Conference 50th Annual Meeting*, 9–14 August 1992, Pittsburgh, Pennsylvania, USA, Poster Abstract No. PA106.
Rossmann, M. G. (2014). *IUCrJ*, **1**, 84–86.
Zeldin, O. B., Brewster, A. S., Hattne, J., Uervirojnangkoorn, M., Lyubimov, A. Y., Zhou, Q., Zhao, M., Weis, W. I., Sauter, N. K. & Brunger, A. T. (2015). *Acta Cryst.* D**71**, 352–356.