# CrystFEL matrix notation conventions

CrystFEL's UnitCell structure doesn't expose its contents as a matrix, only as named components ($a_x^*$, $c_y$ an so on). However, the arrangement of the transformation matrices given to functions such as cell_transform_rational() are important. This document aims to make the conventions explicit. The notation used in CrystFEL follows that used in the International Tables for Crystallography, volume A, chapter 5.1 ("Transformations of the coordinate system (unit-cell transformations)") and 11.1 ("Point coordinates, symmetry operations and their symbols").

- *cell_transform_gsl_direct*, *cell_transform_rational* and *cell_transform_intmat* all do what is written in equation 1. These functions all take a $\hat{\boldsymbol{P}}$ matrix. The "intmat" and "rational" versions will additionally determine the centering of the resulting unit cell (and hopefully eventually the lattice type and unique axis).
- *transform_indices* does what is written in equation 2. It takes a vector of (reciprocal space) Miller indices and a $\hat{\boldsymbol{P}}$ matrix.
- A SymOpList is essentially a list of $\hat{\boldsymbol{W}}^{-1}$ matrices, which, as shown below, behave like $\hat{\boldsymbol{P}}$ matrices.
- transform_fractional_coords_rtnl takes a $\hat{\boldsymbol{P}}$ matrix and a column vector of fractional coordinates, calculates the $\hat{\boldsymbol{Q}}$ matrix and evaluates equation 3 (NB **not** equation 5). *Implementation detail:* it actually solves a matrix-vector equation rather than going all the way to the inverted matrix.
- transform_fractional_coords_rtnl_inverse takes a $\hat{\boldsymbol{P}}$ matrix and a column vector of fractional coordinates $\boldsymbol{v}$, and does the reverse of transform_fractional_coords_rtnl. That means it just has to evaluate $\hat{\boldsymbol{P}}\boldsymbol{v}$. This looks like equation 5.

## Matrices of unit cell components

When matrix calculations are performed in CrystFEL, which is usually only inside certain libcrystfel functions, basis vectors $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ are written as the columns of the matrix.

$$\hat{\boldsymbol{M}} = \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{pmatrix}$$

Reciprocal basis vectors are written as the rows of the matrix.

$$\hat{\boldsymbol{R}} = \begin{pmatrix} a_x^* & a_y^* & a_z^* \\ b_x^* & b_y^* & b_z^* \\ c_x^* & c_y^* & c_z^* \end{pmatrix}$$

Real and reciprocal spaces are sometimes said to be "inverse transposes of each other". The components of the two matrices are already transposed with respect to one another (one has vectors in columns, the other in rows). Therefore $\hat{\boldsymbol{M}} = \hat{\boldsymbol{R}}^{-1}$ and $\hat{\boldsymbol{R}} = \hat{\boldsymbol{M}}^{-1}$.

## Transformation of unit cell basis

Given a transformation matrix $\hat{\boldsymbol{P}}$ which means $\boldsymbol{a}' = \boldsymbol{a} - \boldsymbol{b}$, $\boldsymbol{b}' = \boldsymbol{b}$, $\boldsymbol{c}' = \boldsymbol{c}$:

$$\hat{\boldsymbol{M}}' = \hat{\boldsymbol{M}}\hat{\boldsymbol{P}} = \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \bar{1} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

Note that the "amount of old a,b,c to make new a" is found in the first column of the $\hat{\boldsymbol{P}}$ matrix, and so on.

## Transformation of reciprocal basis

With the same transformation as before ($\boldsymbol{a}' = \boldsymbol{a} - \boldsymbol{b}$, $\boldsymbol{b}' = \boldsymbol{b}$, $\boldsymbol{c}' = \boldsymbol{c}$), the reciprocal basis transforms as follows:

$$\hat{\boldsymbol{R}}' = \begin{pmatrix} a_x^* & a_y^* & a_z^* \\ b_x^* & b_y^* & b_z^* \\ c_x^* & c_y^* & c_z^* \end{pmatrix}' = \hat{\boldsymbol{P}}^{-1}\hat{\boldsymbol{R}} = \hat{\boldsymbol{Q}}\hat{\boldsymbol{R}} = \begin{pmatrix} 1 & 0 & 0 \\ \bar{1} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} a_x^* & a_y^* & a_z^* \\ b_x^* & b_y^* & b_z^* \\ c_x^* & c_y^* & c_z^* \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_x^* & a_y^* & a_z^* \\ b_x^* & b_y^* & b_z^* \\ c_x^* & c_y^* & c_z^* \end{pmatrix}$$

## Miller indices (reciprocal space) under transformation of unit cell basis (real-space)

From ITA chapter 5.1, this is how the reflections would be re-labelled if the axes were changed. See later for how to apply symmetry operations.

$$(h', k', l') = (h', k', l')\,\hat{\boldsymbol{P}} = (h, k, l) \begin{pmatrix} 1 & 0 & 0 \\ \bar{1} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (h - k, k, l) \tag{2}$$

## Fractional coordinates under transformation of unit cell basis

Where u, v and w are fractional coordinates, i.e. the cartesian coordinates are:

$$\hat{\boldsymbol{M}}\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ a_z & b_z & c_z \end{pmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

From ITA chapter 5.1, the new coordinates are:

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \hat{\boldsymbol{P}}^{-1}\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \hat{\boldsymbol{Q}}\begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{3}$$

## Symmetry operations acting on fractional coordinates

Following ITA chapter 11.1, the following matrix represents the symmetry operation written as $-y, x, z$, where for this time only, x,y and z are fractional coordinates:

$$\hat{\boldsymbol{W}} = \begin{pmatrix} 0 & \bar{1} & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

It transforms the fractional coordinates as follows:

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \hat{\boldsymbol{W}}\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 0 & \bar{1} & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix} \tag{5}$$

Note also that we're ignoring the translational part of the symmetry operation (there isn't one in this case).

Note that $\hat{\boldsymbol{W}}$ behaves like $\hat{\boldsymbol{P}}^{-1}$ above. CrystFEL's "SymOpList" structure contains IntegerMatrix objects representing $\hat{\boldsymbol{W}}^{-1}$ matrices, also known as $\hat{\boldsymbol{P}}$ matrices.

## Symmetry operations acting on Miller indices (reciprocal space)

Since $\hat{\boldsymbol{W}}^{-1}$ matrices are just $\hat{\boldsymbol{P}}$ matrices, simply change the $\hat{\boldsymbol{P}}$ into $\hat{\boldsymbol{W}}^{-1}$ in equation 2:

$$(h', k', l') = (h, k, l)\,\hat{\boldsymbol{W}}^{-1} = (h', k', l')\begin{pmatrix} 0 & 1 & 0 \\ \bar{1} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (\bar{k}, h, l) \tag{6}$$

In this case, the symmetry operation is a rotation, so the matrix inverse is just the same as a matrix transpose. However, don't get confused: the matrix appearing here is the inverse of the one appearing in equation 4.

Note that the "amount of old h, k and l to make new h" is found in the first column of the $\hat{\boldsymbol{W}}^{-1}$ matrix, and so on.

## Transforming symmetry operations

Given a $\hat{\boldsymbol{P}}$ matrix which transforms the unit cell as written in equation 1, and a $\hat{\boldsymbol{W}}^{-1}$ matrix representing a symmetry operation, we seek to apply the symmetry operation to the cell as it would appear after applying the transformation. For example, let $\hat{\boldsymbol{P}}$ permute the cell axes, turning a "unique axis b" cell into a "unique axis c" one, and let $\hat{\boldsymbol{W}}^{-1}$ represent a twofold rotation around the c axis:

To transform a set of Miller indices $(h, k, l)$, first use $\hat{\boldsymbol{P}}$ to get the indices referred to the "unique axis c" setting using equation 2, then apply the symmetry operation using equation 6, and finally transform back to the original cell using equation 2 and the inverse matrix:

$$(h', k', l') = (h, k, l)\hat{\boldsymbol{P}}\hat{\boldsymbol{W}}^{-1}\hat{\boldsymbol{P}}^{-1}$$

The combined matrix $\hat{\boldsymbol{P}}\hat{\boldsymbol{W}}^{-1}\hat{\boldsymbol{P}}^{-1}$ represents a twofold rotation symmetry operation along the b axis instead of c.