# CyXtal: C(P)ython Package for Crystal Plasticity Data Analysis

**development note**

Chen Zhang

January 21, 2016

## 1 Module: ctools

## 2 Module: cxtallite

## 3 External Module: ext_aps

### 3.1 parsers.py

This particular module include several function that help facilitate the analysis of data collected at beamline-34-ID-E at Advanced Photon Source in Argonne National Lab.

1. **parser_xml(. . . )**

2. **strain_refine(. . . )**
   The algorithm of strain refinement is based on the software package "LaueGo" developed and maintained by Dr. Tischler at APS, ANL. [1]

   The strain refinement is not a direct (deterministic) measurement of strain using DAXM characterization. Instead, the strain tensor extracted from the Laue diffraction pattern is more of a "guessed value". The general algorithm of extracting strain tensor from DAXM scan data is as follows:

   - Acquire the crystal orientation by forward prediction of the Laue diffraction pattern. Due to small strain and experimental measurement error, the simulated Laue pattern will only partially match the measured pattern. Here the word "partially" has two levels of meanings:

[1] http://www.aps.anl.gov/Sectors/33_34/microdiff/

  - Only some of the diffraction spots are used to identify the crystal orientation of the voxel as it is not possible to match all diffraction spots in the pattern due to non-ideal deconvolution process. [2]
  - The matching between the calculated diffraction spots and measured diffraction spots are not perfect. The difference between calculation and measurements can be attributed to experimental error (uncontrollable and hopefully small) and the residual strain present in the crystal lattice, which distort the diffraction spots from its ideal position.

- Using the crystal orientation obtained by ignoring the lattice distortion (allowing relatively big mismatch of Laue patterns), it is possible to calculate the ideal [reciprocal] lattice vectors for given voxel. Perturbing the ideal [reciprocal] lattice vectors (or lattice constants) and recalculating the diffraction spots, one can theoretically stumble upon the real lattice vectors with lattice strain present. Comparing the transformation matrix between the ideal and real, the deformation gradient can be found through

  $$\mathbf{F} = \mathbf{V}_{\text{ideal}}\mathbf{V}_{\text{real}}^T$$

  where $\mathbf{V}$ is the bases formed by [reciprocal] lattice vectors.

  With the deformation gradient known, it is fairly easy to calculate the strain tensor through the Green strain tensor,

  $$\begin{aligned}\mathbf{C} &= \mathbf{F}^T\mathbf{F} \\ &= (1+\epsilon)^2 \\ &\approx 1 + 2\epsilon + O(\epsilon^2)\end{aligned}$$

  So the strain tensor for given voxel can be approximated by

  $$\epsilon = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I})$$

- Since it is not possible to directly compute $\mathbf{V}_{\text{real}}$, an automated optimization process is used to find a well "guessed" $\mathbf{V}_{\text{real}}$. However, it is possible that different optimization algorithm and initial guessing conditions will lead to different $\mathbf{V}_{\text{real}}$, which is why the strain tensor extracted using this method is not deterministic.

Another point worth mentioning here is that the full strain tensor can only be "guessed" when the beam energy is known. In other words, without the knowledge of the length of $\mathbf{q}$ (diffraction vector), it is not possible to inferred the whole strain tensor. However, if the volume of the unit cell is assumed to remain constant through deformation, the

[2] The shadows from neighboring grains and echoes from sample internal cannot be fully eliminated during the deconvolution of the diffraction pattern, which is used to acquire Laue pattern for each voxel during post processing of DAXM data.

deviatoric component of the strain tensor is not tied to $||\mathbf{q}||$. [3] Thus, it is necessary to add this assumption to the optimization if white beam calibration is not done, which add one more step to get the deviatoric strain component,

$$\epsilon_{dev} = \epsilon - \frac{1}{3}\text{tr}(\epsilon)$$

# 4 External Module: ext_damask

# 5 External Module: ext_vtk

# 6 Appendix

## 6.1 Find Base Vectors from Lattice Constants

In the strain refinement, one important step is to find the reciprocal lattice vectors from given lattice constants. The easiest way to find the reciprocal lattice vectors is to find its dual, real space lattice vectors.

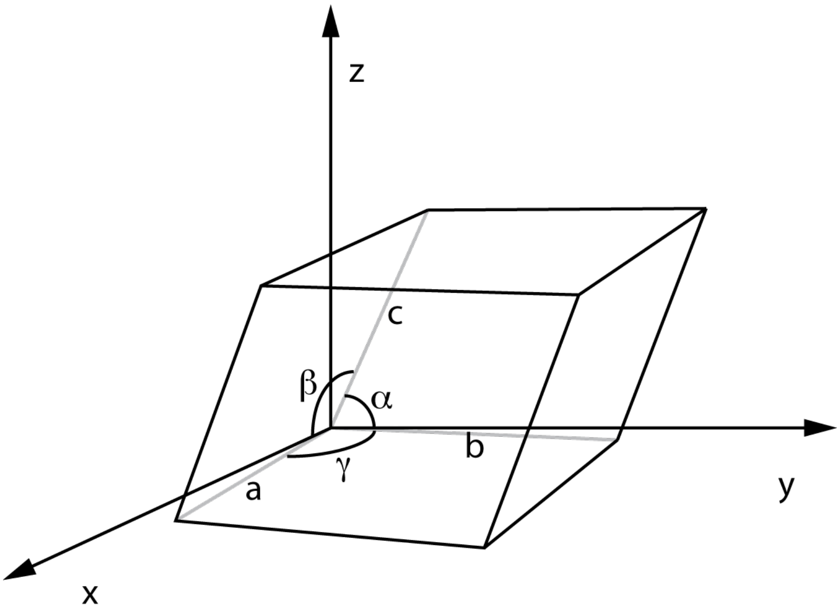

Figure 1: A general unit cell with its six lattice constants

Fig. 1 shows an example of unit cell randomly oriented in space. Although there are six parameters $(a, b, c, \alpha, \beta, \gamma)$ available to describe the shape of the unit cell, the exact numerical representation of the unit cell, namely the base lattice vectors, is not easy to determine.

[3] Diffraction vector: $\mathbf{q}_{hkl} = \mathbf{V} \cdot (khl)$

To make the math a little bit easier, let assume that **a** is align with x-axis, which gives us

$$\mathbf{a} = (a_1, a_2, a_3) = (a, 0, 0)$$

Then let's determine the x-y plane through **a** and **b**. In other words, **b** can be easily written out as

$$\mathbf{b} = (b_1, b_2, b_3) = (b\cos\gamma, b\sin\gamma, 0)$$

For a general unit cell, the volume of the cell is determined through its six lattice parameters/constants,

$$V = abc\sqrt{1 + 2\cos\alpha\cos\beta\cos\gamma - \cos^2\alpha - \cos^2\beta - \cos^2\gamma}$$

The volume of the unit cell can also be calculated through

$$V = A_{xy} \cdot c_z = abc_3 \sin\gamma$$

which gives us

$$c_3 = \frac{V}{ab\sin\gamma}$$

Arbitrary choice of the reference system should not affect the angle between two vectors, which gives us

$$\begin{aligned} \mathbf{a}\cdot\mathbf{c} &= ac\cos\beta = a_1c_1 + a_2c_2 + a_3c_3 \\ \mathbf{b}\cdot\mathbf{c} &= bc\cos\alpha = b_1c_1 + b_2c_2 + b_3c_3 \end{aligned}$$

Since **a** and **b** are known, we have

$$\begin{aligned} a_1c_1 + 0 + 0 &= ac_1 = ac\cos\beta \\ b_1c_1 + b_2c_2 + 0 &= bc_1\cos\gamma + bc_2\sin\gamma = bc\cos\alpha \end{aligned}$$

Solve the above linear system, we have

$$\begin{aligned} c_1 &= c\cos\beta \\ c_2 &= \frac{\cos\alpha - \cos\beta\cos\gamma}{\sin\gamma}c \end{aligned}$$

So the last base vector, **c** can be expressed as

$$\mathbf{c} = (c\ \cos\beta, c\,\frac{\cos\alpha - \cos\beta\cos\gamma}{\sin\gamma}, \frac{V}{ab\sin\gamma})$$

where $V$ is the volume of the unit cell.

To sum it up, for a general unit cell with six lattice parameters known, if we select **a** to be x-axis and **b** to lie in the x-y plane, the expression of base lattice vectors can be expressed as

$$\begin{aligned}
\mathbf{a} &= (a, 0, 0) \\
\mathbf{b} &= (b\cos\gamma, b\sin\gamma, 0) \\
\mathbf{c} &= (c\ \cos\beta, c\,\frac{\cos\alpha - \cos\beta\cos\gamma}{\sin\gamma}, \frac{V}{ab\sin\gamma})
\end{aligned}$$

where $V$ is the volume of the unit cell and can be calculate through

$$V = abc\sqrt{1 + 2\cos\alpha\cos\beta\cos\gamma - \cos^2\alpha - \cos^2\beta - \cos^2\gamma}.$$

Another convection would be forcing **b** to be parallel to y-axis, however this convention has been dropped at APS. Thus, there is no reason to consider other situation anymore.

## 6.2 Coordinates Transformation for DAXM Dataset

Due to experiment setup of DAXM characterization, there are three different coordinate systems related to our data processing (Fig. 2):

- **XHF**
  This is the configuration closely tied to the sample stage motor, which records sample movement during the scan. The spatial position for each voxel in the DAXM data can be directly calculated from the recording of the motor position (Xsample/Ysample/Zsample) and the wire position(depth).

- **APS**
  This is the official coordinate system at APS, where the beam direction is the z-axis. At beamline 34-ID-E, the indexation results is computed in this coordinate system. In other words, the indexation results, including the reciprocal lattice and diffraction vectors (**q**) are all in this coordinate system.

- **TSL**
  This coordinate system is closely tied to the sample where the sample surface normal is usually consider to be z-axis. It is also the default coordinate system in TSL software, where OIM data is processed. To make the coordinate system transformation easier, the x-axis in the TSL system is aligned with the x-axis in APS coordinate system. However, this could be changed depending on how the sample is mounted in during EBSD and DAXM scan.

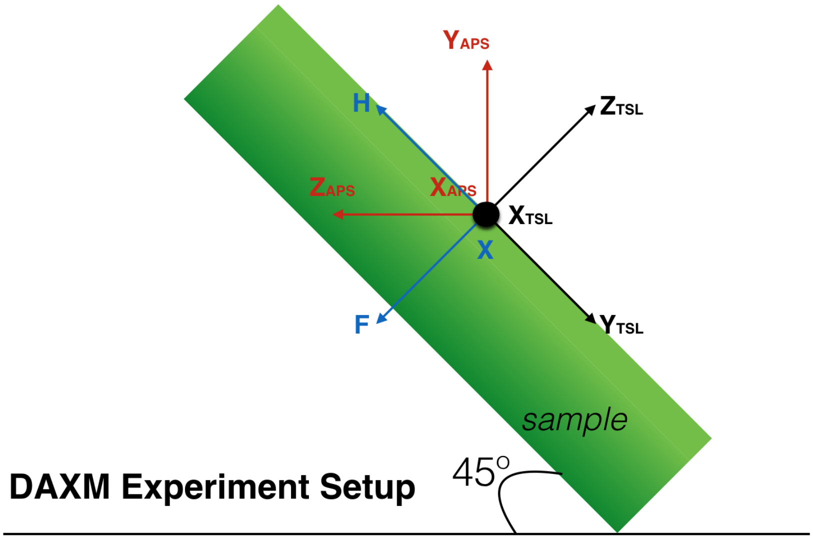

Figure 2: Three different coordinate system in DAXM characterization, assuming x-axis is shared by APS and TSL coordinate system.