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 DAXM characterization uses white beam to capture a convoluted diffraction patterns from the entire volume. Then a Pt wire, also known as the differential aperture, is used to block the diffraction signals from certain voxels. During the post processing, deconvolution of the diffraction patterns is performed to extract the Laue diffraction pattern for each voxel, thus to achieve spatially revolved characterization of the material.

   Since the final results are essentially Laue diffraction patterns, all the analysis that can be used for Laue diffraction can be applied to the DAXM data as well. Particularly, it is possible to extract the crystal rotation and lattice strain by analyzing the mismatch between ideal diffraction spots (zero rotation, strain free) and measured diffraction spots. The underlying assumption is that the deformation of the crystal can be resolved into two steps: a simple rotation that causes

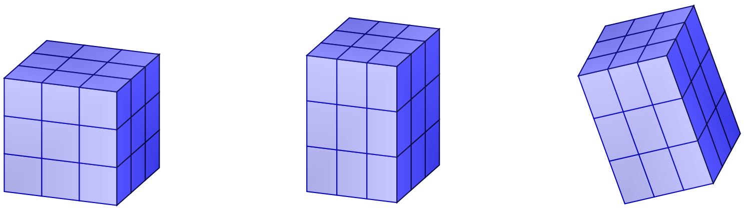

Figure 1: According to continue mechanics theory, the plastic deformation can be resolved into stretching ($\mathbf{U}$) and rotation ($\mathbf{R}$) and , namely $\mathbf{F} = \mathbf{RU}$.

diffraction peaks to shift and a small lattice stretch that causes diffraction peaks to streak(Fig. 1).

Assuming small lattice strain present in the sample, the indexation program executed at APS uses forward prediction method to match the diffraction spots while ignoring the streaking effect present in the diffraction patterns. The results of this indexation is a set of reciprocal lattice vectors with no strain present (strain free). In other words, the diffraction spots calculated from these set of reciprocal lattice vectors will differ from those measured. This mismatch between calculation and measurements can be attributed to experimental error and residual lattice. Since the indexation results (*.xml file) also contains the measured diffraction vectors used to identify the reciprocal lattice vectors, it is possible to further process the data to extract the lattice strain. The strain refinement algorithm used in this package is based on the software package “LaueGo” developed and maintained by Dr.Tischler at APS, ANL. [1]

Assuming that the residual mismatch between the theoretical diffraction peaks and measured diffraction peaks are the results of residual lattice strain, it is possible to construct a strained unit cell that generate a diffraction pattern that fits the measurement better. Comparing the strained unit cell and the ideal unit cell, a deformation gradient ($\mathbf{F}$) can be calculated by

$$\mathbf{B}_{\text{strained}} = \mathbf{F}\ \mathbf{B}_{\text{ideal}}$$

where $\mathbf{B}$ is the basis tensor whose column vectors are unit cell base vectors.

With $\mathbf{F}$ known, the Green strain tensor can be calculate by

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

So the lattice strain for given voxel can be approximated by

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

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

Since there is no explicit method available to find a strained unit cell that generates better diffraction pattern, it is inevitable to use an implicit numeric approach to locate a possible solution. The overall goal of this implicit numeric approach is to minimize the difference between the calculated diffraction peaks and the measured ones. To quantify the mismatch, angular difference between all diffraction peak pairs (calculated and measured),

$$err = \sum \mathbf{q}_{\text{calculated}} \cdot \mathbf{q}_{\text{measured}}$$

where $\mathbf{q}_{\text{calculated}}$ can be calculated by

$$\mathbf{q}_{\text{calculated}} = \mathbf{B}^*_{\text{strained}} \cdot \mathbf{v}_{\text{hkl}}.$$

Here $\mathbf{B}^*_{\text{strained}}$ is the basis in reciprocal space whose columnar vectors are the reciprocal base vectors and $\mathbf{v}_{\text{hkl}}$ is the indexation for the diffraction peak.

The relationship between $\mathbf{B}$ and $\mathbf{B}^*$ is

$$\mathbf{B}^* = 2\pi\mathbf{B}^{-T}$$

Considering most of the searching will be done in the reciprocal space, it is necessary to find the connection between $\mathbf{B}^*$ and $\mathbf{F}$.

Let $\mathbf{B}_2$ be the basis for strained and rotated unit cell, $\mathbf{B}_0$ be the basis for ideal unit cell (zero rotation, strain free), and $\mathbf{B}_1$ be the basis for the unit cell only considering stretch.

$$\mathbf{B}_2 = \mathbf{F}\mathbf{B}_0 = \mathbf{R}\mathbf{B}_1 = \mathbf{R}\mathbf{U}\mathbf{B}_0$$

where $\mathbf{R}$ represents the lattice rotation and $\mathbf{U}$ represents the lattice stretch.

Convert everything to reciprocal space except for $\mathbf{F}$, [2]

$$\begin{aligned}\mathbf{B}_2 &= \mathbf{F}\mathbf{B}_0\\ \mathbf{B}^*_2 &= \mathbf{F}^{-T}\mathbf{B}^*_0\end{aligned}$$

Now the deformation gradient can be expressed using reciprocal basis,

$$\mathbf{F} = \mathbf{B}_2^{*-T}\mathbf{B}_0^{*T}$$

Thus the strain tensor can be written as

$$\begin{aligned}\epsilon &= \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I})\\ &= \frac{1}{2}((\mathbf{B}_2^{*-T}\mathbf{B}_0^{*T})^T\mathbf{B}_2^{*-T}\mathbf{B}_0^{*T} - \mathbf{I})\\ &= \frac{1}{2}(\mathbf{B}_0^*\mathbf{B}_2^{*-1}\mathbf{B}_2^{*-T}\mathbf{B}_0^{*T} - \mathbf{I})\end{aligned}$$

[2] Inverse transpose does not change the order of matrix multiplication.

Considering that rotation is not reflected in the Green strain tensor,

$$\mathbf{F}^T\mathbf{F} = \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U} = \mathbf{U}^T\mathbf{U}$$

it is possible to skip the last rotation by using $\mathbf{U} = \mathbf{B}_1^{*-T}\mathbf{B}_0^{*T}$

$$\begin{aligned}\epsilon &= \frac{1}{2}(\mathbf{U}^T\mathbf{U} - \mathbf{I}) \\ &= \frac{1}{2}(\mathbf{B}_0^*\mathbf{B}_1^{*-1}\mathbf{B}_1^{*-T}\mathbf{B}_0^{*T} - \mathbf{I})\end{aligned}$$

Since all reciprocal lattice vectors are in APS coordinate system(Section 6.2), the strain tensor calculated here is also in the APS coordinate system.

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 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,

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

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.

Fig. 2 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.

To make the math a little bit easier, let assume that $\mathbf{a}$ is align with x-axis, which gives us

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

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

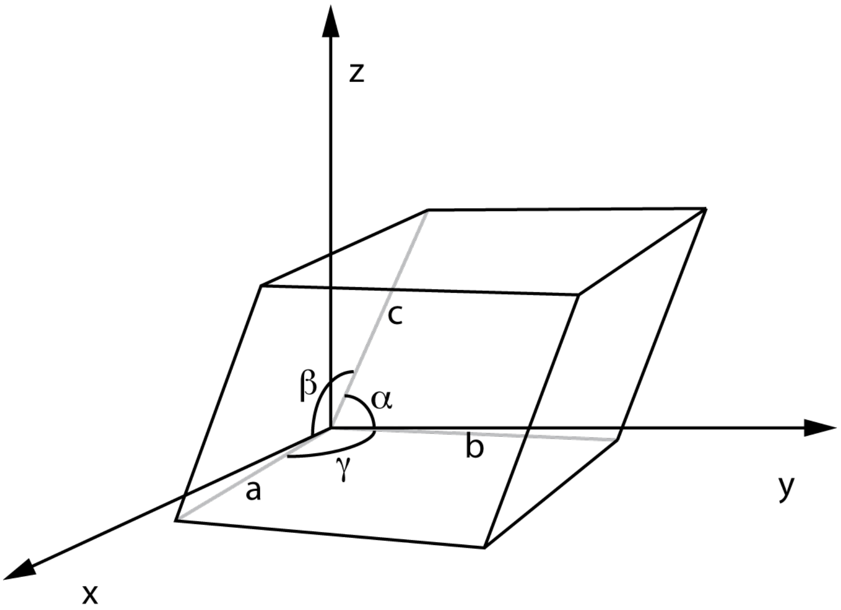

Figure 2: A general unit cell with its six lattice constants

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

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

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

$$a_1c_1 + 0 + 0 = ac_1 = ac\cos\beta$$
$$b_1c_1 + b_2c_2 + 0 = bc_1\cos\gamma + bc_2\sin\gamma = bc\cos\alpha$$

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 this European convention 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. 3):

- **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.

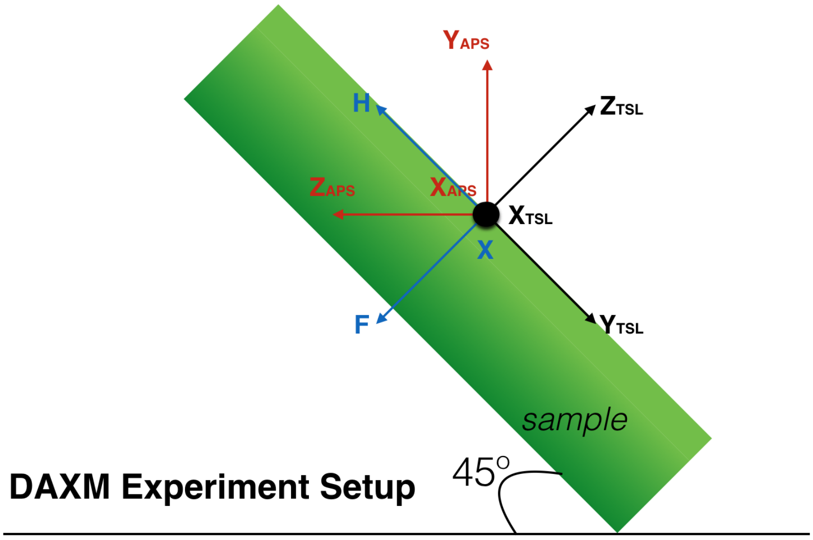

Figure 3: Three different coordinate system in DAXM characterization, assuming x-axis is shared by APS and TSL coordinate system.