Python Analysis Tool for Epitaxial thin film X-ray diffraction data (PATEX)

Version 0.1 (Alpha) (Feb 2024) | Created by Zach Drinkwater

Python Analysis Tool for Epitaxial thin film X-ray diffraction data (PATEX) has been developed for the quick processing of XRD data collected in theta – 2 theta scans. This document outlines the functionality and limitations of PATEX.

As of Version 0.1, PATEX consists of a Python module (patex.py) as well as a Jupyter Notebook (PATEX\_notebook.ipynb). To use the tool, open patex\_notebook.ipynb in Jupyter Notebook and follow the instructions.

The purpose of this notebook and the module xrd\_module.py is to provide a data analysis pipeline for thin film XRD data. This code fits curves to the peaks in the data and then uses the fitted curves to estimate peak location, width, height. Peaks locations are used to find the out of plane lattice parameter using Bragg's law

This code fits curves to the peaks in the XRD data and then uses the fitted curves to estimate peak location, width, height with errors. Peaks locations are used to find the out of plane lattice parameter using Bragg's law. In the future, the stress will be calculated using the Williamson-Hall method. More stress testing is needed to further explore the functionality and limitations.

The following documentation covers what is included in the patex.py module. The Jupyter Notebook gives an example of how the module can be used.

**patex.py**

This is an object-oriented module. It requires NumPy, Matplotlib, and SciPy to function.

The module is centered around the object:

**patex.XRDPeakData**

* An XRDPeakData object is created with three initial parameters:
  + **filename (string):** the name of the data file.
  + **lam (float):** the wavelength of the X-ray used in XRD in metres.
  + **skiprows (integer):** the number of rows to skip in the data file.
* An example of creating an object is as follows:
  + myDataObject = patex.XRDPeakData(‘my\_data.txt’, 1.5406e-10, 3)

Methods of patex.XRDPeakData

The methods can be broken down into two groups. The first group are those that a user of the tool would find useful. The second group consists of methods used by methods belonging to the first group that a user would seldom use, unless they were trying to edit the module.

Group 1:

* **patex.XRDPeakData.filter\_data(sigma)**

Filters the data using a 1D Gaussian with order sigma (integer).

* **patex.XRDPeakData.plot\_raw\_peak\_data(log)**

Plots the raw data and takes log base 10 of the intensity data if log (bool) is True.

* **patex.XRDPeakData.plot\_local\_data(idx, idx\_halfrange)**

Plots local data around an index value (1 idx = 0.01 degrees). A user might find the function patex.idx\_to\_angle useful.

* **patex.plot\_peak\_data(min\_peak\_intensity, max\_peak\_intensity, idx\_halfrange, distance, prominence)**

Plots local data around all peaks found by SciPy, with no further filtering of peaks. Parameters are:

* + **min\_peak\_intensity** (float): the minimum peak height
  + **max\_peak\_intensity** (float): the maximum peak height (not very useful)
  + **idx\_halfrange** (int): the half range in index count
  + **distance (int):** the minimum index count between peaks
  + **prominence (float):** a factor that SciPy uses to look at the relative size between peaks and surround intensity count
* **patex.XRDPeakData.fit\_all\_peaks(peak\_params, curve='g', film=True, film\_left=True, num\_peaks=4, plot=True, Q=False)**
  + The primary workhorse of the module. This either fits all substrate OR all film peaks and returns the peak information. It changes the data object’s peak\_data attribute also. We go into more detail for this method as it is the most important. The parameters are:
    - **peak\_params (list)**: A list of parameters for confining which peaks to identify. This includes:
      * **min\_peak\_intensity (float):** the minimum peak height
      * **max\_peak\_intensity (float):** the maximum peak height (not very useful)
      * **idx\_halfrange (int)**: the half range in index count
      * **distance (int):** the minimum index count between peaks
      * **prominence (float):** a factor that SciPy uses to consider at the relative size between peaks and surrounding intensity
      * **width (list):** a list with two floats. These are the minimum and maximum peak widths
    - **curve (string):** Which type of curve to use. 'g' is Gaussian, 'l' is Cauchy-Lorentz, and 'v' is Voigt. The default is 'g'.
    - **film (bool):** Whether to fit film or substrate peaks. The default is True (film).
    - **film\_left (bool):** Whether the film peaks are on the left (True) or right (False) of the substrate peaks. The default is True.
    - **num\_peaks (int):** The expected number of substrate OR film peaks. The default is 4.
    - **plot (bool):** Whether to output a plot of the fitted curves. The default is True.
    - **Q (bool):** If True, then the fitting is done in reciprocal lattice units for the purpose of Williamson-Hall analysis. Otherwise, the fitting is done in degrees. The default is False
  + RETURNS:
    - **peak\_data (Numpy Array):** an array containing the peak location, intensity, FWHM, location error, intensity error, and FWHM error for each peak.
* **patex.XRDPeakData.get\_lattice\_parameter(peak\_data, plot)**
  + This method calculates the out of plane lattice parameter for the fitted data. It should be noted that fit\_all\_peaks must be called before this method will work.
    - **peak\_data (Numpy Array):** This is the output array from fit\_all\_peaks.
* **patex.XRDPeakData.williamson\_hall – WORK IN PROGRESS**

Group 2:

* **patex.XRDPeakData.cut\_range(idx, idx\_halfrange)**
  + This method is used during fitting to cut down the range of values being analysed to a suitable range around the identified peak.
* **patex.XRDPeakData.fit\_gaussian((theta\_range, intensity\_range, p0)**
  + Fits data to a Gaussian curve over the range theta\_range (NumPy array) and intensity\_range (Numpy Array) with initial parameter guesses p0 (list)
* **patex.XRDPeakData.fit\_lorentz**
  + Fits data to a Cauchy-Lorentz curve over the range theta\_range (NumPy array) and intensity\_range (Numpy Array) with initial parameter guesses p0 (list)
* **patex.XRDPeakData.fit\_voigt**
  + Fits data to a Voigt curve over the range theta\_range (NumPy array) and intensity\_range (Numpy Array) with initial parameter guesses p0 (list)
* **patex.XRDPeakData.fit\_pseudo\_voigt – WORK IN PROGRESS**

Other Functions:

* patex.idx\_to\_angle(idx)
  + Assuming the smallest XRD angle is 15 degrees, it converts an index unit to degrees.
* patex.line(x, m, c)
  + Returns
* patex.gaussian(x, A, x0, sigma)
  + Returns
* patex.cauchy\_lorentz(x, A, x0, gamma)
  + Returns
* patex.voigt(x, A, x0, sigma, gamma)
  + Given and w(z) is the real part of the Faddeeva function, returns

Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Changes | Date | Name |
| Version 0.1 | Created PATEX and documentation | February 2024 | Zach Drinkwater |
|  |  |  |  |