5/12/2022

Azimuthor – Guide

Paulo Ricardo Garcia – pauloricardoafg@yahoo.com.br
João Zerba – joao.zerba@lnls.br

Azimuthor is a graphical user interface (GUI) for azimuthal average of X-ray/neutron/electron scattering data using PyFAI python library. The idea of this GUI is to make the integration process more intuitive and, in this way, accessible and practical to the user. The input are the geometry parameters, the mask and the scattering image. These parameters can be obtained using FIT2D or pyFAI tools

The mask file should be in .msk or npy formats. The scattering images file should be in HDF5, H5, CXI, NPY and Tiff.

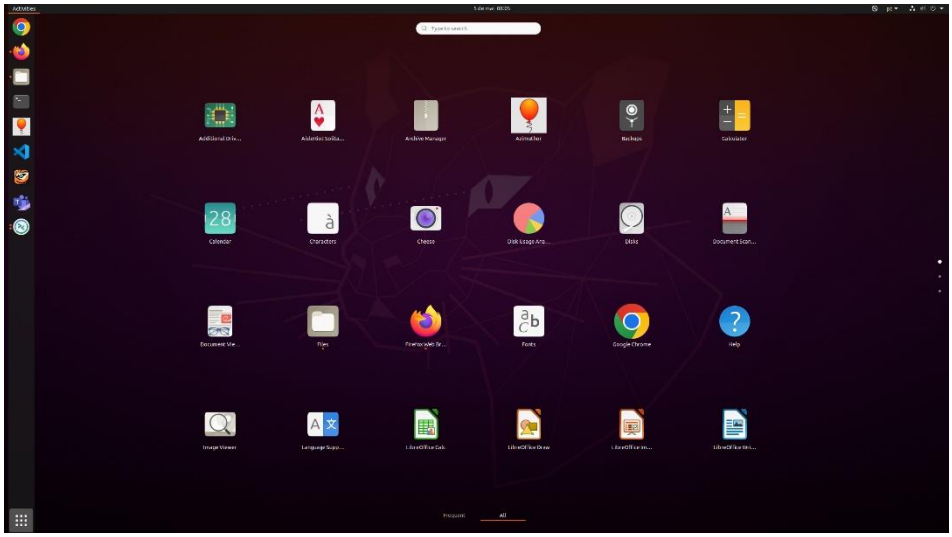
The software can work in single and batch modes, the last one is suitalble when dealing with multiple files. It has an interactive display in which the integrated curves may be vizualised and selected for exportation. The data are stored in hdf5 and dat files.

Sumário

1. Opening the software 2
2. Setting the integration conditions 2
3. Single File Mode 4
4. Batch Mode 5
5. Plot tab 6
6. Data structure 6
References 7

1. Opening the software

Azimuthor can be found in Applications > Azimuthor:



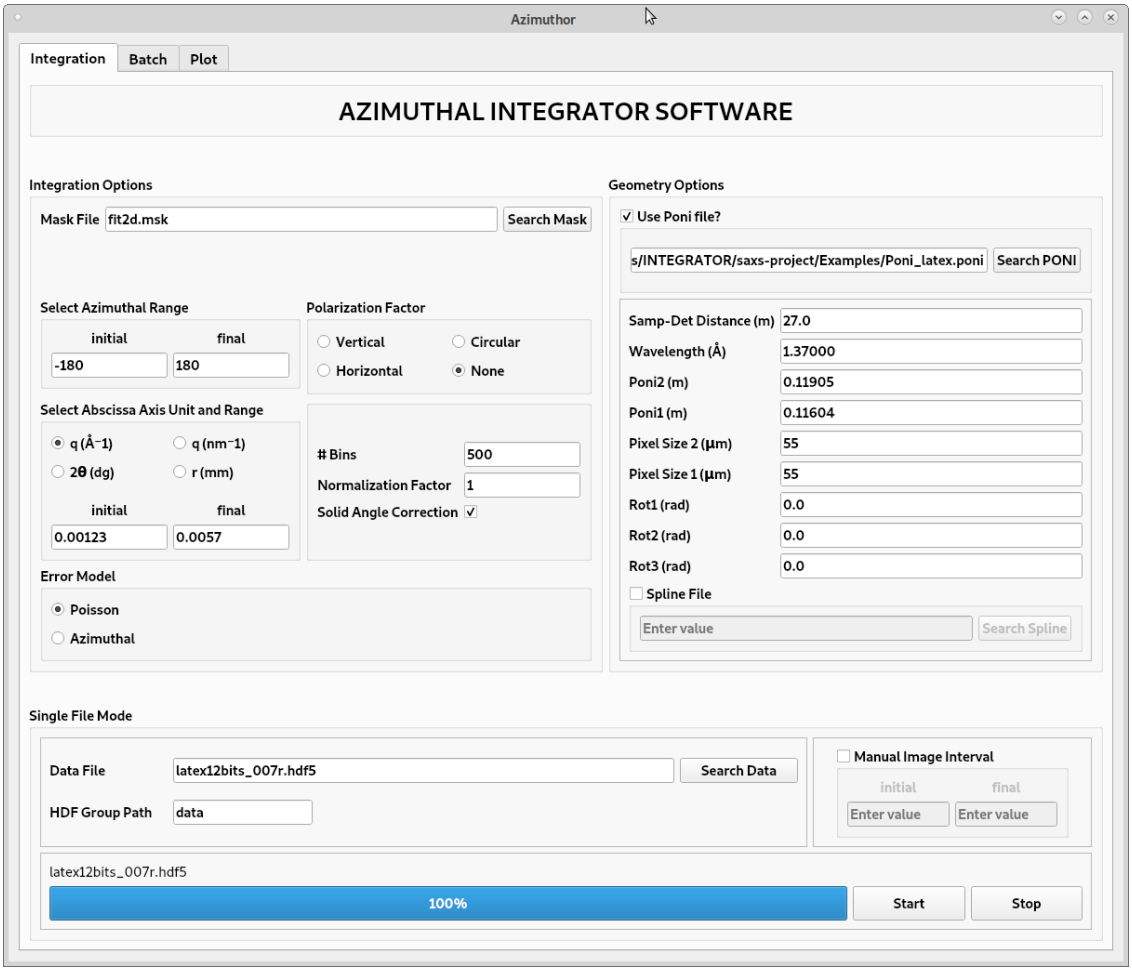
With the balloon icon  . This can be added to Favorites by the user.

It can also be accessed with a Terminal console. To do that, press CRTL + ALT + T to open a Terminal application. Then, just type:

- *Azimuthor*

2. Setting the integration conditions

When opening Azimuthor the ***Integration tab*** will be displayed. This tab contains three large fields: ***Integration Options***, in which the integration conditions are set; ***Single File Mode,*** in which the integration of a single file can be performed, and ***Geometry Options,*** containing the geometry parameters of the experiment.



The variables of the **Integration options** are:

- ***Mask File:*** press the Search Mask button and choose the file of the mask to be used. If the mask was done with Fit2D please hit the yes button on the pop-up window. Masks done in FIT2D need to be vertically flipped;
- ***Select Azimuthal Range:*** Set the azimuthal angle interval in degree. To integrate the full range of the azimuthal angle it is necessary to consider -180° to 180° (pyFAI requirement);
- ***Polarization Factor:*** Polarization of the radiation wave. It can be vertical, horizontal, circular or none;
- ***Abscissa Axis Unit Range:*** Abscissa unit of the output integrated curves. It can be q (nm^{-1}), q ($Å^{-1}$), the scattering angle 2θ (deg) or the integration radius r (nm). If you leave the fields ***initial*** and ***final*** empty pyFAI will automatically set the abscissa range for the integration;
- ***Bins:*** Number of points of the integrated curves;

- ***Normalization Factor:*** Multiplicative factor;
- ***Solid Angle Correction:*** Perform the solid angle correction for each pixel;

In the ***Geometry options*** field, the geometry variables may be defined either by loading a PONI calibration file or manually filling the fields. If you use a PONI file, check the checkbox ***Use Poni file?*** and click on the button ***Search PONI*** to load the PONI file. If you are not going to use a PONI file, leave the checkbox unchecked and fill the geometry fields.

It is important to notice that the unit system used is different in each case. If a PONI file is loaded the variables will be loaded as defined in PyFAI library. Otherwise, the variables will be loaded as defined in Fit2D. Next table shows the variables and units for each case:

With PONI (PyFAI)		**Without PONI (Fit2D)**	
Variable	***Unit***	***Variable***	***Unit***
Sample to detector distance	meter	Sample to detector distance	meter
Wavelength	Å	Wavelength	Å
Poni 2	µm	X-center	px
Poni 1	µm	Y-center	px
X-Pixel	µm	X-Pixel	µm
Y-Pixel	µm	Y-Pixel	µm
Rot 1	degree	Tilt	degree
Rot 2	degree	Tilt Plan Rot	degree
Rot 3	degree	-	-

*Poni = Point of normal incidence. Consult the PyFAI documentation for more information.

The user can load a ***spline file*** describing the detector distortions. More information about the variables can be found in the PyFAI documentation in the web.

3. Single File Mode

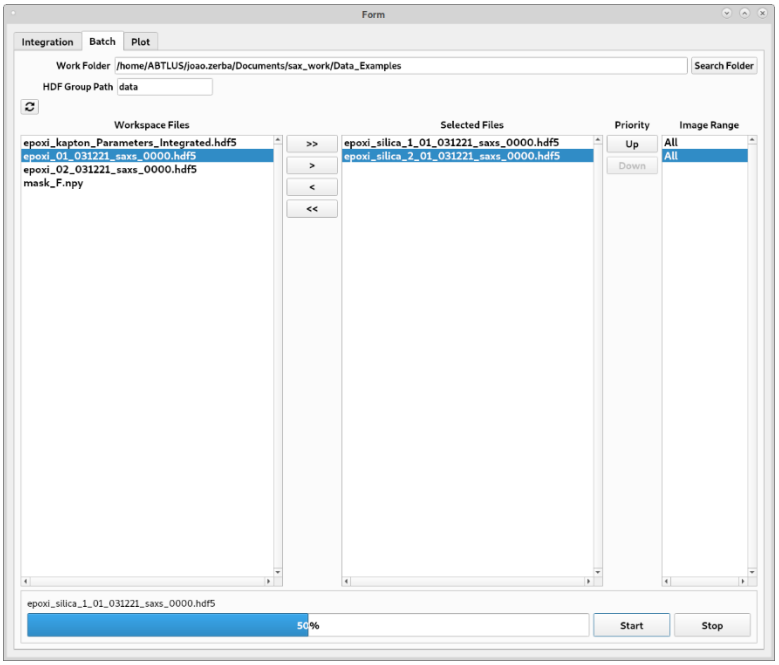
In the single file mode, the integration of just one file can be performed. The file can be loaded in the ***Data File*** field. The files format supported are hdf, numpy (npy) and tif. If an hdf file is loaded the user can define the hdf group where the scattering image(s) is(are) stored in the ***HDF Group Path*** field.

It is possible to define an interval of images to be integrated. In this case, check the ***Image Interval*** checkbox and write the interval limits in the ***initial*** and ***final*** fields.

Then click on **start** button. Once the progress bar is completed go ***Plot tab*** where the results will be displayed.

4. Batch Mode

In the batch mode, it is possible to integrate several files sharing the same parameters (set at the ***Integration*** tab). To select the files for the batch integration first choose the work folder. It is also needed to fill in the ***HDF Group Path*** field. Then use the directional buttons or double click on the files at the vertical column ***Workspace Files*** to be selected or at the ***Selected Files*** space to remove them. It is possible to change the priority of integration using the ***Up*** and ***Down*** buttons.



As default, the **Image Range** of integration is set to ***All*** images of the file. If not all the images are intended to be integrated, it is possible to use the "printer notation" to slice portions of images of each dataset. The index starts at 0. Example:

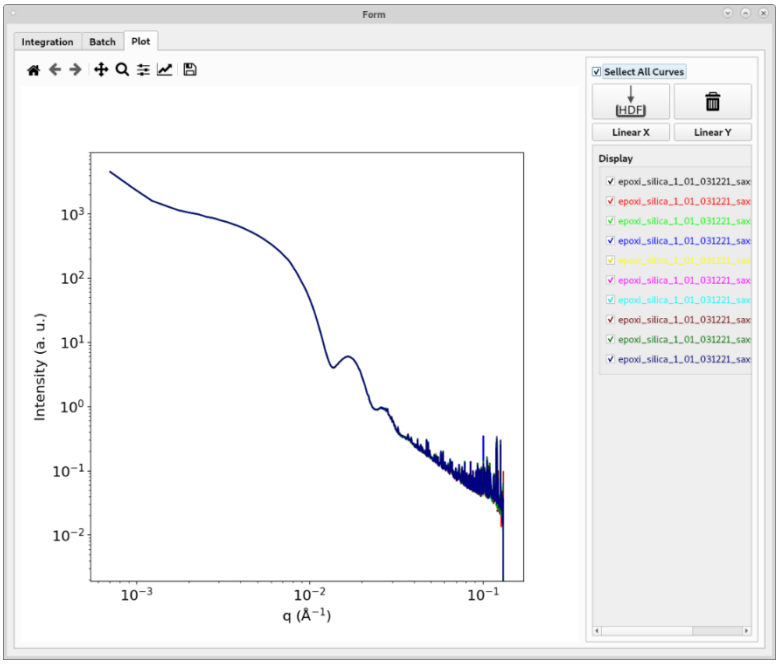
Image Interval	Description
7	It is going to integrate the image 7 of the dataset.
7–23	It is going to integrate the images between 7 and 23 positions.
7–23; 45–99;237-458	It is going to integrate three portions of the dataset by dividing each slice by semicolon (**;**).
45–99;7;237-458	It is going to integrate all images between 45 and 99, image 7 and all images from 237 to 458.

Then click on **start** button. Once the progress bar is completed go ***Plot tab*** where the results will be displayed.

5. Plot tab

In this tab it is possible to view and interact with the integration curves, to toggle between linear and logarithm axis representation, to hide and delete an unchecked curve, and to save the data.

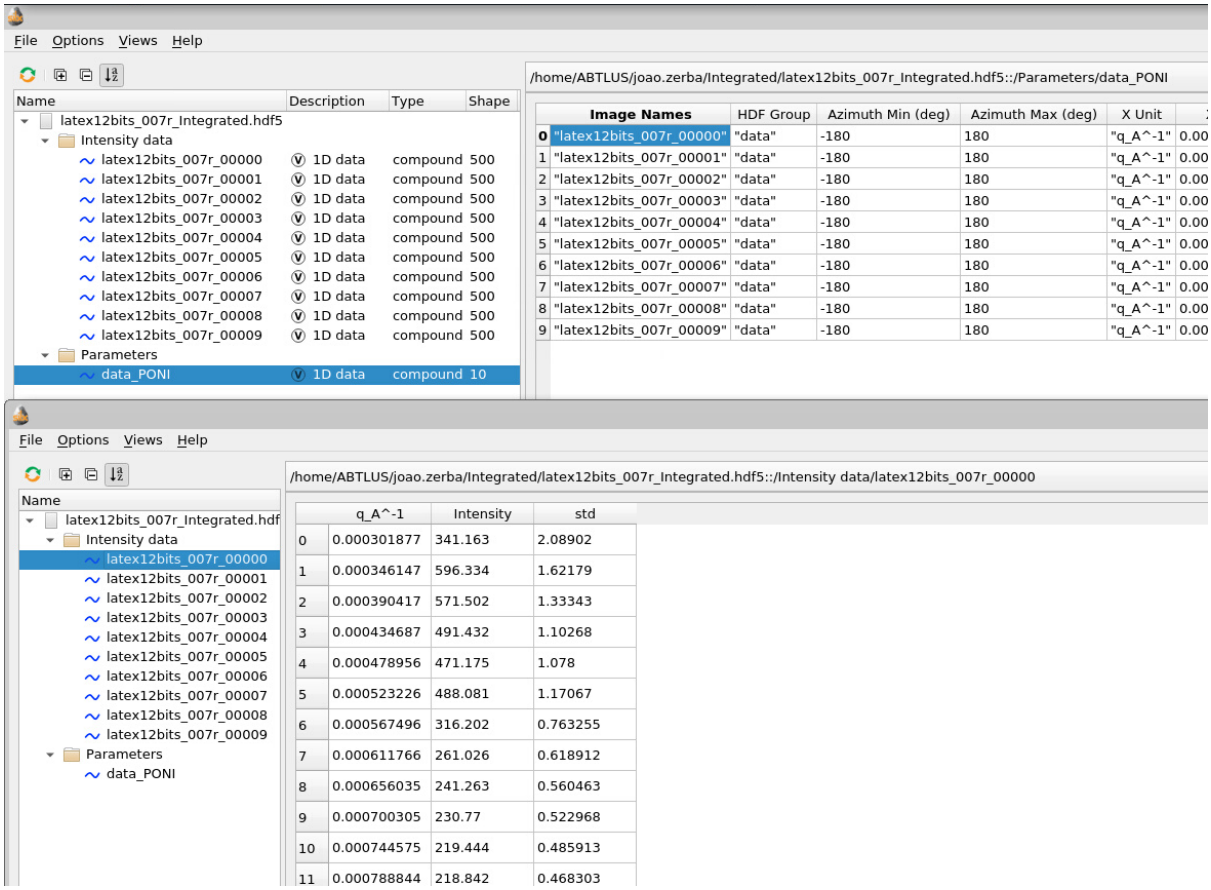
- To hide a curve just uncheck it;
- To delete a curve uncheck it and click on the ***trash can button***;
- To save the **checked curves** and integration data click on the ***download HDF button***. It will save not only in hdf5 format but also in dat, in the Home directory of the user.



6. Data structure

When saving the data, a folder called ***Integrated*** is created in the Home directory, where all the HDF and dat files will be saved.

The HDF file is organized in two groups, ***Intensity data*** and ***Parameters***. The first one stores the curves as a data frame of 3 columns: the axis unit, the intensity, and the standard deviation. The second stores all the information used for the integration for each curve.



References

Fast Azimuthal Integration using Python. Available on https://pyfai.readthedocs.io/en/master/. Accessed on May 12th,2022

Ashiotis, G., Deschildre, A., Nawaz, Z., Wright, J. P., Karkoulis, D., Picca, F. E., & Kieffer, J. (2015). The fast azimuthal integration Python library: pyFAI. *Journal of applied crystallography*, *48*(2), 510-519.

Hammersley, A. P. (2016). FIT2D: a multi-purpose data reduction, analysis and visualization program. *Journal of Applied Crystallography*, *49*(2), 646-652.

Kieffer, J., & Karkoulis, D. (2013, March). PyFAI, a versatile library for azimuthal regrouping. In *Journal of Physics: Conference Series* (Vol. 425, No. 20, p. 202012). IOP Publishing.