pyRS: A Python package for the reduction and analysis of neutron residual stress data

Chris M. Fancher[1, 2], **Jean Bilheux**[2], **Wenduo Zhou**[3], **Ross E. Whitfield**[3], **Jose Borreguero**[3], **William F. Godoy**[3], **Steven Hahn**[3], and **Peter F. Peterson**[3]

1 Materials Science & Technology Division, Oak Ridge National Laboratory, Oak Ridge, TN **2** Neutron Scattering Division, Oak Ridge National Laboratory, Oak Ridge, TN **3** Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN

DOI: 10.21105/joss.03935

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Submitted: 19 November 2021
Published: 20 November 2021

Statement of need

The 2nd Generation Neutron Residual Stress Facility (NRSF2) residual stress mapping instrument at the High Flux Isotope Reactor (HFIR) at ORNL was recently rebuilt with a modern detector and control system. Upgrading from a LabVIEW-based control software (SPICE) to an Experimental Physics and Industrial Control System (EPICS) based control software with the neutron Event Distributor (nED) data acquisition system brought additional experimental flexibility.(Vodopivec & Vacaliuc, 2017; White et al., 2019) The transition from a control system that measured discrete data to an event-based data structure.(Peterson et al., 2015) deprecated the data reduction and analysis software. The design of `pyRS` relied on years of experience with the previous reduction and analysis software (NRSF View) to ensure the new software builds upon the strengths and improves on the weakness of NRSF View. The lack of a unified analysis and visualization of residual stress data was among the most significant needs.

Summary

`pyRS` is a python software package that was designed to meet the data reduction and analysis needs of the neutron residual stress mapping user community at Oak Ridge National Laboratory (ORNL). `pyRS` incorporates separate modules that provide a streamlined workflow for reducing raw neutron events into 1D intensity vs. scattering angle and subsequent analysis to extract the interatomic spacing and intensity for residual stress and texture analysis. Users can access the modules through either a graphical or command-line interface. `pyRS` saves data into a single HDF5 file.(The HDF Group, 1997-NNNN), in which the metadata, reduced diffraction data, and peak analysis results are passed between different modules.

Overview of pyRS

`pyRS` was designed with three distinct graphical user interfaces (GUIs) that enable users to 1) reduce neutron event data, 2) perform single-peak fitting of reduced data, and 3) combine single-peak fitting results for residual stress analysis and subsequent visualization. Figure 1 provides an overview for how data flow through and where users interact with `pyRS`.

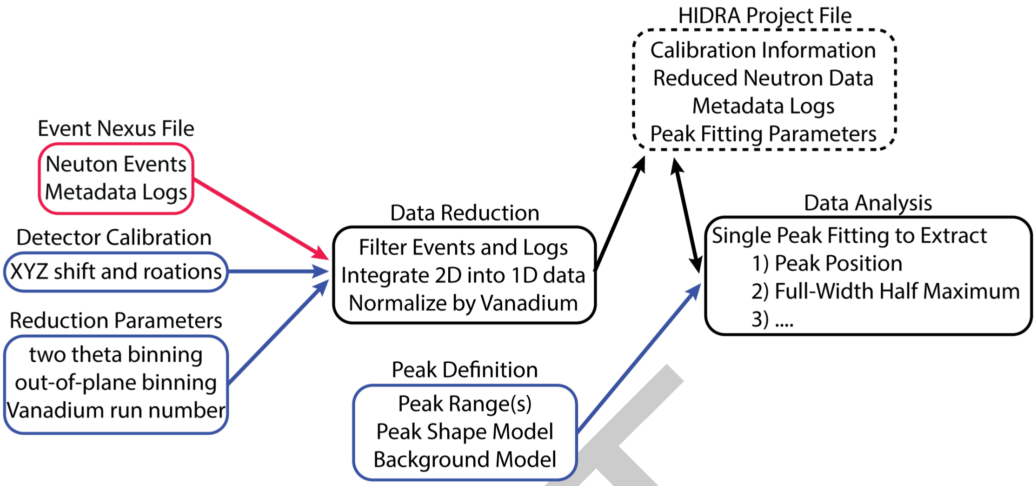

Figure 1: Overview of how `pyRS` takes in raw neutron data (red) and user inputs ($blue$) into the Data Reduction and Data Analysis components. The Data Reduction creates a HIDRA Project File that is then appended with analysis results. Note that the user can specify the inputs through a graphical or python scripting interface.

- Data Reduction
  - *Filter Events and Logs*: The High-Intensity Diffractometer for Residual Stress Analysis (HIDRA) stores raw measured neutron events in HDF5 files using an event data structure using the NeXus standard schema.(Könnecke et al., 2015) The event NeXus data structure stores information about the pixel position and detection time with respect to the start of a measurement. HIDRA leverages this flexibility to encode scan_index metadata signals that `pyRS` uses to filter events into separate datasets. `pyRS` reduces measured events based on how scan_index increments throughout a single NeXus file. `pyRS` reconstructs the measured 2-dimensional diffraction datasets by first filtering the event index array based on the scan_index time, then histogramed based on pixel position (np.histogram).(Harris et al., 2020) Metadata events are filtered using the Mantid framework time-filtering algorithm.(Arnold et al., 2014). Users can specify to exclude unwanted logs.
  - *Integrate 2D into 1D data*: Calibration information about the position of the detector in space (XYZ shifts and rotations about the engineering position) are used to determine the angular position of the detector pixels. Pixel angular position and intensity data are histogramed to construct raw Intensity vs. scattering angle datasets based on the default or user-defined angular range.
  - *Normalize by Vanadium (optional)*: raw Intensity vs. scattering angle is normalized by the incoherent scattering intensity from a Vanadium sample if a Vanadium run number is defined.
  - A HIDRA project file is created that stores the calibration information, Intensity vs. scattering data, and metadata logs
- Data analysis
  - *Peak Fitting Analysis*
    * Reduced 1D data are analyzed using single-peak fitting to extract information about the position, intensity, full-width half maximum of N peaks within the detector field of view. Users can define specific peak fitting ranges using the graphical interface or using a JSON formatted text file. Users can export the graphically select peak ranges into a JSON file for later use. Peak fitting results are automatically appended into the loaded HIDRA project file. Alternatively, users can export a CSV summary of the results.

- *Residual Stress Analysis*
    * Residual stress analysis requires peak fitting results for 2 or 3 orthogonal directions. `pyRS` does not limit users to only defining a single HIDRA project file per direction. `pyRS` can merge multiple project files based on the spatial position metadata logs. `pyRS` determines residual stresses using a simple linear elasticity model to related the resulting stress from the calculated strain from the measured Bragg peaks typical in neutron scattering experiments, by:

$$\sigma_{ii} = \frac{E}{(1+\nu)}\left[\varepsilon_{ii} + \frac{\nu}{1-2\nu}\left(\varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33}\right)\right] \tag{1}$$

where

$$\varepsilon_{ii}(x,y,z) = \frac{d^{ii}_{hkl}(x,y,z)}{d^0} - 1 \tag{2}$$

and

$$d_{hkl}(x,y,z) = \frac{\lambda}{2sin\theta(x,y,z)} \tag{3}$$

in which:

* σ_{ii}: orthogonal residual stresses
* ε_{ii}: orthogonal calculated strains
* d: atomic lattice d-spacing from Bragg's Law
* d_0: nominal atomic lattice d-spacing
* hkl: crytallographic plane indices
* (x,y,z): spatial coordinates
* λ: measured wavelength
* θ: angle measured from a normal surface

Acknowledgements

The authors are grateful for the contributions of E. Andrew Payzant, Garrett Granroth, Jeff R. Bunn and Anibal Ramirez Cuesta for reviewing the paper. A portion of this research used resources at the High Flux Isotope Reactor, a DOE Office of Science User Facility operated by the Oak Ridge National Laboratory. Work at Oak Ridge National Laboratory was sponsored by the Division of Scientific User Facilities, Office of Basic Energy Sciences, US Department of Energy, under Contract no. DE-AC05-00OR22725 with UT-Battelle, LLC. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

References

Arnold, O., Bilheux, J. C., Borreguero, J. M., Buts, A., Campbell, S. I., Chapon, L., Doucet, M., Draper, N., Ferraz Leal, R., Gigg, M. A., Lynch, V. E., Markvardsen, A., Mikkelson, D. J., Mikkelson, R. L., Miller, R., Palmen, K., Parker, P., Passos, G., Perring, T. G., … Zikovsky, J. (2014). Mantid—Data analysis and visualization package for neutron scattering and <math altimg="si0002.gif" overflow="scroll"> <mi mathvariant="normal">μ</mi> </math> SR experiments. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *764*, 156–166. https://doi.org/10.1016/j.nima.2014.07.029

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Könnecke, M., Akeroyd, F. A., Bernstein, H. J., Brewster, A. S., Campbell, S. I., Clausen, B., Cottrell, S., Hoffmann, J. U., Jemian, P. R., Männicke, D., Osborn, R., Peterson, P. F., Richter, T., Suzuki, J., Watts, B., Wintersberger, E., & Wuttke, J. (2015). The NeXus data format. *Journal of Applied Crystallography*, *48*(1), 301–305. https://doi.org/10.1107/S1600576714027575

Peterson, P. F., Campbell, S. I., Reuter, M. A., Taylor, R. J., & Zikovsky, J. (2015). Event-based processing of neutron scattering data. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *803*, 24–28. https://doi.org/10.1016/j.nima.2015.09.016

The HDF Group. (1997-NNNN1997-NNNN). *Hierarchical Data Format, version 5*. http://www.hdfgroup.org/HDF5/

Vodopivec, K., & Vacaliuc, B. (2017). High Throughput Data Acquisition With Epics. *Conference on Accelerator and Large Experiment Control Systems*, 213–217. https://doi.org/10.18429/JACoW-ICALEPCS2017-TUBPA05

White, G., Slac, M. S., Park, M., Johnson, A. N., Rivers, L., Shen, G., Anl, S. V., Bnl, K. S., Island, L., York, N., Cosylab, M. S., Dls, T. C., Ess, T. K., Gareth, D., Euxfel, H., Junkes, H., Gregor, M., Frib, K., Lansing, E., … Self, K. (2019). THE EPICS SOFTWARE FRAMEWORK MOVES FROM CONTROLS TO PHYSICS. *International Particle Accelerator Conference (10th)*, 1216–1218. https://doi.org/10.18429/JACoW-IPAC2019-TUZZPLM3