*MStack*: Stacking disorder tools for *Python*

Peter C. Metz\*

*Inamori School of Engineering, Alfred University, 2 Pine St., Alfred, NY 14802*

*\*Contact:* [*pcm1@alfred.edu*](mailto:pcm1@alfred.edu) *| (315) 350 1585*

Abstract

In order to refine stacking disorder models in real and reciprocal space, *MSTACK* has been written to extend two established profile generators: *DIFFaX*, a reciprocal space intensity distribution calculator built on a stochastic stacking disorder model description; and *DiffPy-CMI*, a suite of tools including pair distribution function calculators. *MSTACK* includes tools to expand the stochastic stacking model parameters typical of *DIFFaX* into supercell models suitable for calculation of stacking disordered pair distribution function data, and to drive refinement of layer structure models from real and reciprocal space data.

*MSTACK* has been designed with advanced refinement tools in mind. *MSTACK* is built on the code *lmfit* which permits the user to include arbitrary constraint equations enabling parametric refinement. Further, *MSTACK* is designed to be compatible with any minimizer method in the *SciPy* package, enabling the application of global minimization techniques. Currently implemented minimization methods include the L-BFGS-B non-linear optimization algorithm and the Differential Evolution algorithm. Finally, *lmfit* and *MSTACK* enable the user to apply Markov-Chain Monte Carlo analysis, via the package *emcee*, to the resulting fit. This Bayesian statistical analysis tool has been used predominantly in the astronomical community to interpret data with substantial noise where the error in the data is uncertain. Application of this tool to PDF data suggests many model parameters are not normally distributed- an insight that is expected to have substantial impact on the future of scattering analysis of nanostructured material.

Acknowledgements

This software is based upon work supported by the National Science Foundation under

Grant No. DMR-1409102. Example data, where indicated, was collected using the resources

of the Advanced Photon Source, a U.S. Department of Energy (DOE) Office of Science User

Facility operated for the DOE Office of Science by Argonne National Laboratory under

Contract No. DE-AC02-06CH11357.

Table of Contents

[Abstract 1](#_Toc495313398)

[Acknowledgements 2](#_Toc495313399)

[Table of Contents 3](#_Toc495313400)

[1 Installation and dependencies 4](#_Toc495313401)

[1.1 *Python* dependencies 4](#_Toc495313402)

[1.2 Installing and compiling *DIFFaX* 4](#_Toc495313403)

[1.3 Installation of *MSTACK* 5](#_Toc495313404)

[2 Quick Start 6](#_Toc495313405)

[3 What can’t currently be done 7](#_Toc495313406)

[3.1 Calculation of G(r) for interleaved layer structures 7](#_Toc495313407)

[3.2 Full portability between DIFFaX transition matrix and PDF calculator 7](#_Toc495313408)

[4 General Refereces 8](#_Toc495313409)

# Installation and dependencies

*MSTACK* is written in *Python 2.7*, and depends on *DiffPy-CMI* which is currently implemented in Linux only. The easiest way to obtain the correct dependencies is to install [*Anaconda*](https://www.continuum.io/downloads), which in addition to *Python* contains a useful development environment (*Spyder*), and the science computing notebook *IPython* as well as the various science computing packages required.

## *Python* dependencies

Once Python is installed, *DiffPy-CMI* can be installed by following the instructions on their [website](http://www.diffpy.org/products/diffpycmi/). In brief, the following commands can be issued from a bash shell

conda config --add channels diffpy

conda install diffpy-cmi

conda update diffpy-cmi

The additional dependencies can be installed from the Python Package Index, *e.g.*

pip install lmfit

pip install emcee

pip install corner

pip install tabulate

pip install dill

\*pip install mpi4py

\*pip install shwimmbad

\*Required for use on MPI-enabled computing cluster

## Installing and compiling *DIFFaX*

In order to use the program *DIFFaX*, a binary will need to be compiled in Linux. *DIFFaX* is distributed by Mike Treacy and can be downloaded from the [*DIFFaX* website](http://www.public.asu.edu/~mtreacy/DIFFaX.html). Copy the *DIFFaX* tarball to your working directory, untar the directory, and compile the source code. This will require the installation of a suitable Fortran compiler. In Ubuntu

sudo apt-get install gfortran

f77 DIFFaX.f –o DIFFaX.sh

./diffax.sh

If there are permissions errors, try giving the user execution permission

chmod gu+x ./diffax.sh

## Installation of *MSTACK*

*MStack* can be installed using *pip*. Unpack the tarball into an appropriate directory, then run

pip install –e /*path*/mstack\_0.\*

Or, simply copy the *mstack* directory into your working directory.

# Quick Start

Once the program is installed, it is recommended to work in an *ipython* notebook. From a terminal

ipython qtconsole

will open a *ipython* notebook with a graphical backend enabled. By default plots will be embedded in the notebook, however interactive plotting can be enabled with the *ipython* magic

%matplotlib qt # interactive plotting

%matplotlib inline # inline plotting

The QT backend is required for plotting of the fit residual as a function of iteration.

Standard bash commands like cd and pwd are integrated into the *ipython* environment for user convenience.

Navigate to the examples directory and run one of the demonstration files. If all the dependencies are satisfied this should raise no errors. The basic structure and refinement objects are demonstrated in *~/Examples/make\_structure\_objects*.*py.*

~! The data sets and code utilized in the dissertation P. Metz,2017 has been archived on FigShare (hyperlink). Try perusing these examples.

~! Simple case: calculate the PDF and powder diffraction profile for graphene layers

~! Adding parameter constraints

# What can’t currently be done

## Calculation of G(r) for interleaved layer structures

* Requires refactoring of interface module
* Current supercell expander takes asymmetric unit of a layer type and expands it with a layer transition vector
* Simplest solution- N positional asym units for 1x1xN supercell
* Other options: indexed layer structures passed with explicit ordering. E.g.:
  + structure.number = 1, etc.
  + sequence = (1,2), …(1,2,3,2),…(1, 3, 5, 2, 4, 1, 1, 1, 1, 1, 1, 5, 1)….etc.

## Full portability between DIFFaX transition matrix and PDF calculator

* Goes with problem outlined above
* Currently, G(r) approximated as weighted sum of non-interacting states (e.g. nlayers = 1 for k layer types)
* Full portability requires refactoring to interpret number of unique layer structures, number of independent neighbor sheet relations, and constructors for the appropriate supercells.

# General Refereces

1. P. Juhás, C.L. Farrow, X. Yang, K.R. Knox, and S.J.L. Billinge, “Complex Modeling: a strategy and software program for combining multiple information sources to solve ill posed structure and nanostructure inverse problems” *Acta Crystallogr. Sect. A Found. Adv.* **71** [6] 562–568 (2015).

2. M. Newville, T. Stensitzki, D.B. Allen, and A. Ingargiola, “LMFIT: Non-Linear Least-Squares Minimization and Curve-Fitting for Python” *Zenodo* (2014).

3. D. Foreman-Mackey, D.W. Hogg, D. Lang, and J. Goodman, “emcee : The MCMC Hammer” *Publ. Astron. Soc. Pacific* **125** [925] 306–312 (2013).

4. D. Foreman-mackey, “corner . py : Scatterplot matrices in Python” *J. Open Source Softw.* **1** [2] 1–2 (2016).

5. M.M.J. Treacy, J.M. Newsam, and M.W. Deem, “A General Recursion Method for Calculating Diffracted Intensities from Crystals Containing Planar Faults” *Proc. R. Soc. A Math. Phys. Eng. Sci.* **433** [1889] 499–520 (1991).

1–5