

# PDFfit2 and PDFgui: computer programs for studying nanostructure in crystals

C L Farrow<sup>1</sup>, P Juhas<sup>1</sup>, J W Liu<sup>1</sup>, D Bryndin<sup>1,4</sup>, E S Božin<sup>1</sup>, J Bloch<sup>2</sup>,  
Th Proffen<sup>3</sup> and S J L Billinge<sup>1</sup>

<sup>1</sup> Department of Physics and Astronomy, Michigan State University, East Lansing,  
MI 48824-2320, USA

<sup>2</sup> Institute for Theoretical Physics, University of Regensburg, 93040 Regensburg, Germany

<sup>3</sup> Lujan Neutron Scattering Center, Los Alamos National Laboratory, Los Alamos,  
NM 87545, USA

<sup>4</sup> Department of Computer Science and Engineering, Michigan State University, East Lansing,  
MI 48824, USA

E-mail: [billinge@pa.msu.edu](mailto:billinge@pa.msu.edu)

Received 29 March 2007

Published 4 July 2007

Online at [stacks.iop.org/JPhysCM/19/335219](http://stacks.iop.org/JPhysCM/19/335219)

## Abstract

PDFfit2 is a program as well as a library for real-space refinement of crystal structures. It is capable of fitting a theoretical three-dimensional (3D) structure to atomic pair distribution function data and is ideal for nanoscale investigations. The fit system accounts for lattice constants, atomic positions and anisotropic atomic displacement parameters, correlated atomic motion, and experimental factors that may affect the data. The atomic positions and thermal coefficients can be constrained to follow the symmetry requirements of an arbitrary space group. The PDFfit2 engine is written in C++ and is accessible via Python, allowing it to inter-operate with other Python programs. PDFgui is a graphical interface built on the PDFfit2 engine. PDFgui organizes fits and simplifies many data analysis tasks, such as configuring and plotting multiple fits. PDFfit2 and PDFgui are freely available via the Internet.

(Some figures in this article are in colour only in the electronic version)

## 1. Introduction

Increasingly materials scientists and chemists are interested in complex materials which have structure on the nanometre length-scale. Examples are discrete nanoparticles [1], crystals with embedded nanoscale structures [2–4], and nanoporous materials with molecules or nanoparticles intercalated inside the pores [5, 6]. It is difficult to study structure on the nanoscale quantitatively using diffraction methods, because the diffraction gives no sharp Bragg peaks but only broad diffuse features in the scattering. Total scattering, in which both Bragg and diffuse components of the scattering are analysed together, is growing in popularity

for the study of this kind of problem [2, 7]. One approach to analysing the data is to use reverse Monte Carlo [8], the subject of this journal supplement. An alternative approach is to Fourier transform the data to real space to obtain the atomic pair distribution function (PDF) and analyse the data directly in real space [9].

A popular approach for this is to use the profile fitting refinement program PDFfit [10, 11]. This approach assumes that the structure can be described by a relatively small number of atoms in a unit cell, which may be the crystallographic unit cell or it may be a supercell to accommodate symmetry lowering local distortions. The refined parameters are in direct analogy with those determined crystallographically: lattice parameters and unit cell angles, atomic fractional coordinates and anisotropic atomic displacement parameters (ADPs); however, they correspond to the relevant values in the *local* structure. This allows direct comparison between the local structure in a distorted crystal and the average structure determined from the Bragg peaks alone. In the absence of local disorder, the values determined crystallographically and those determined from total scattering studies will be the same. This is not the case in the interesting materials, where the local structure disagrees with the average structure [2, 7, 12]. Here we report a significant redesign and modification of the successful software program PDFfit [13], including a new user-friendly graphical user interface (GUI) with labour-saving data and structure plotting capabilities.

### 1.1. PDF technique

The pair distribution function method was originally developed to study the structure of materials with no long-range order, such as liquids and glasses [14]. The technique is now increasingly applied for investigations of disorder in crystalline materials and for studies of nanomaterials [2]. It is complementary to traditional crystallographic analysis, which assumes a perfect periodic material and considers only intensities from Bragg reflections. The PDF method does not require periodic order in the sample—instead it uses the whole measured spectrum, including Bragg and diffuse components, to extract the total scattering structure function,  $S(Q)$ , which contains coherent scattering intensities from the material [9].

The experimental atomic pair distribution function,  $G(r)$ , is obtained by Fourier transformation of  $S(Q)$ ,

$$G(r) = \frac{2}{\pi} \int_0^{\infty} Q[S(Q) - 1] \sin(Qr) dQ, \quad (1)$$

where  $Q$  is the magnitude of the scattering vector. For elastic scattering,  $Q = 4\pi \sin(\theta)/\lambda$ , with  $2\theta$  being the scattering angle and  $\lambda$  the wavelength of the measured radiation. The  $G(r)$  function is related to the atomic structure through the relation,

$$G(r) = 4\pi r [\rho(r) - \rho_0], \quad (2)$$

where  $\rho_0$  is the atomic number density of the material and  $\rho(r)$  is the atomic pair density which gives the mean weighted density of neighbour atoms at radial distance  $r$  from an atom at the origin. Thus,

$$\rho(r) = \frac{1}{4\pi r^2 N} \sum_i \sum_{j \neq i} \frac{b_i b_j}{\langle b \rangle^2} \delta(r - r_{ij}), \quad (3)$$

where the indices  $i$  and  $j$  go over all  $N$  atoms in the sample,  $b_i$  is the scattering factor of atom  $i$ ,  $\langle b \rangle$  is the average scattering factor, and  $r_{ij}$  is the distance between atoms  $i$  and  $j$ . Full details on the determination of an experimental PDF can be found elsewhere [9].

For a known structure model, the experimental PDF can be calculated using

$$G_{\text{calc}}(r) = \frac{1}{Nr} \sum_i \sum_{j \neq i} \left[ \frac{b_i b_j}{\langle b \rangle^2} \delta(r - r_{ij}) \right] - 4\pi r \rho_0. \quad (4)$$

There are many methods of extracting structural information from measured PDFs. These range from the simple analysis of peak positions, widths and intensities to extract bonding information directly [9, 15, 16], reverse Monte Carlo analysis of large structure models [17, 18], and emerging *ab initio* structure solution methods [19], to the common down-hill least-square refinement as implemented in the PDFfit program [10, 11], and PDFfit2 described here.

## 2. PDFfit2

PDFfit2 is a major upgrade to PDFfit [10], and inherits many of its features. PDFfit is capable of fitting a theoretical three-dimensional structure to an experimentally determined PDF. It can simultaneously fit multiple structures, accounting for different structural phases in a material. PDFfit has a constraint system that allows the expression of structure variables as simple functions of fitted parameters. PDFfit structure variables include lattice constants, data and phase scale factors, atomic site occupation, anisotropic ADPs, and atomic vibrational correlations. PDFfit has a built-in FORTRAN-style command language that understands simple `for` loops and some built-in arithmetic functions.

The original PDFfit was written in FORTRAN-77, which imposes some limitations on the program. For example, it uses fixed-size arrays for internal storage. This precludes the analysis of structures with large cells without modifying the code. Though the constraint system is powerful, it requires that a constraint equation be accompanied by its first derivative. This places the burden of determining the derivatives on the user, which can introduce errors that lead to instability in the convergence. Furthermore, the code is monolithic, not easily extensible, and hard to integrate with external programs.

The primary focus of PDFfit2 development was to remedy the limitations of PDFfit while extending its functionality. The old PDFfit engine has been completely rewritten in C++, and many bugs have been fixed. The new engine uses dynamic memory allocation so that the size of the structure or extent of the fit-range of the PDF is limited only by the physical memory available. The constraint system has also been upgraded. The program *automatically* computes the analytical derivatives of the constraints that are required by the minimization routine. This simplifies user input and reduces the possibility of errors. In addition, new fitting parameters for handling dynamic atomic correlations and experimental resolution have been introduced as well.

Instead of rewriting the PDFfit command interpreter, which is used to define the fitting problem and to control and run the refinement, its functions are carried out using the Python language<sup>4</sup>. Python is a powerful, cross-platform, open-source interpreted programming language (i.e. it does not need to be compiled to run, similar to scripting) that emphasizes object-oriented and modular design. PDFfit2 scripts written in Python syntax take the place of PDFfit macros, and the Python interpreter can handle everything that the old interpreter could and more. Using Python as an interpreter allows PDFfit2 to be combined with and enhanced by other Python libraries. We make use of this capability with PDFgui as described below.

## 3. PDFgui

The PDFfit2 engine can be used either directly from the Python command line or as part of larger and more complex software applications. The first application built on PDFfit2 is PDFgui, a graphical environment for PDF fitting.

<sup>4</sup> Python Software Foundation.

### 3.1. Design principles

PDFgui has been designed to provide users with an easy-to-use yet powerful interface for fitting structure models to PDF data. It makes use of an object-oriented, component-based architecture, which makes it highly extensible and maintainable. This allows for powerful usability features such as real-time plotting and remote execution of the fitting program whilst visualizing the results locally. PDFgui has been designed with multi-tasking in mind. It is multi-threaded, so that the work being done by the PDFfit2 engine does not interfere with the tasks of the user interface.

PDFgui is written in the Python programming language. Python features a relaxed and friendly syntax, supports ‘write once, run anywhere’ portability, and has extensive libraries and modules for virtually every task. Software codes written in a variety of programming languages can be bound into Python, which allows them to be used together. Python is becoming a popular choice in the scientific computation community.

PDFgui’s interface is built using wxPython [20], the Python package for wxWidgets, which is a mature cross-platform GUI library. Graphical applications written in wxPython provide a look and feel native to the platform on which they are run. PDFgui is designed to run on Windows, Mac OS, Linux, and all major Unix systems.

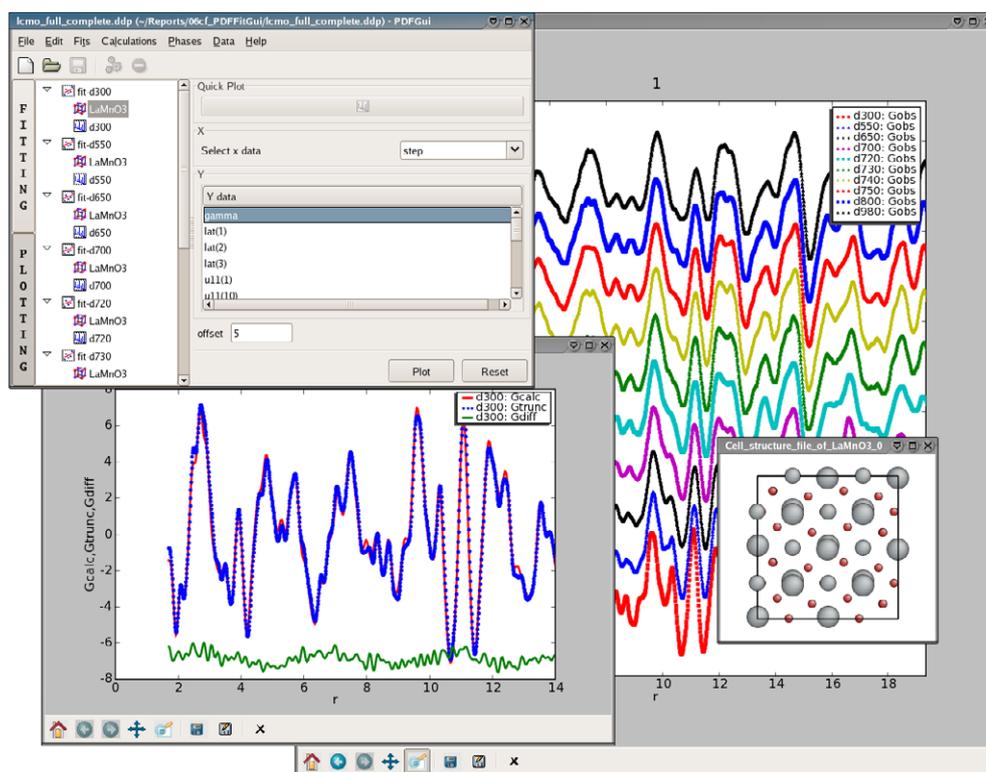
### 3.2. Capabilities

PDFgui contains all of the functionality of PDFfit2 along with additional enhancements for usability. Mundane tasks are handled by the program and difficult tasks are made simple. PDFgui can manage multiple fits at once. Each fit can have multiple experimental data sets and structure models. Fits in a sequence can call upon other fits for their starting parameters, and configured fits can be queued to run while the user is away. All the initial, final, and intermediate data are stored in a platform-independent project file that can be loaded on any computer. All management tasks, such as fit creation, configuration, modification, and visualization, can be done through the graphical interface.

PDFgui supports space group operations. Users can define an asymmetric unit and let PDFgui expand it to a full cell with all symmetry-related positions. PDFgui can also generate symmetry constraints for atom positions and anisotropic ADPs. Users just need to specify the space group, and the program will identify equivalent sites and generate constraint equations for their coordinates and temperature factors to keep the structure consistent with the symmetry requirements. This can be done either for all atoms in the structure or for an arbitrary subset—for example, when it is known that only a certain species shows a local distortion. The code for space group definitions was provided by the Python Macromolecular Library (mmLib) [21]. PDFgui also supports supercell expansion of a normal unit cell.

PDFgui uses the matplotlib [22] Python package for two-dimensional (2D) plotting of data and results. Matplotlib has a friendly interface, so the user can quickly and easily view the results of a fitting. PDFgui lets users plot data from a series of fits and plot it against selected metadata (temperature, doping, etc), plot the results of several fits in the same window, plot the PDF in real time as the fitting is running, plot the parameters or variables in real time as the refinement evolves, and save plots in common image formats or export the data to a text file. PDFgui uses AtomEye [23] for 3D visualization. The modular nature of PDFgui allows for other alternatives in the future, such as PyMOL [24].

PDFgui supports built-in macros for advanced fits. For example, for a set of experimental data from one system at different temperatures or doping levels, PDFgui can expand a template fit to a series of related fits. Another PDFgui macro makes it easy to set up boxcar fits, where the same model is fitted over different  $r$ -ranges of the PDF data.



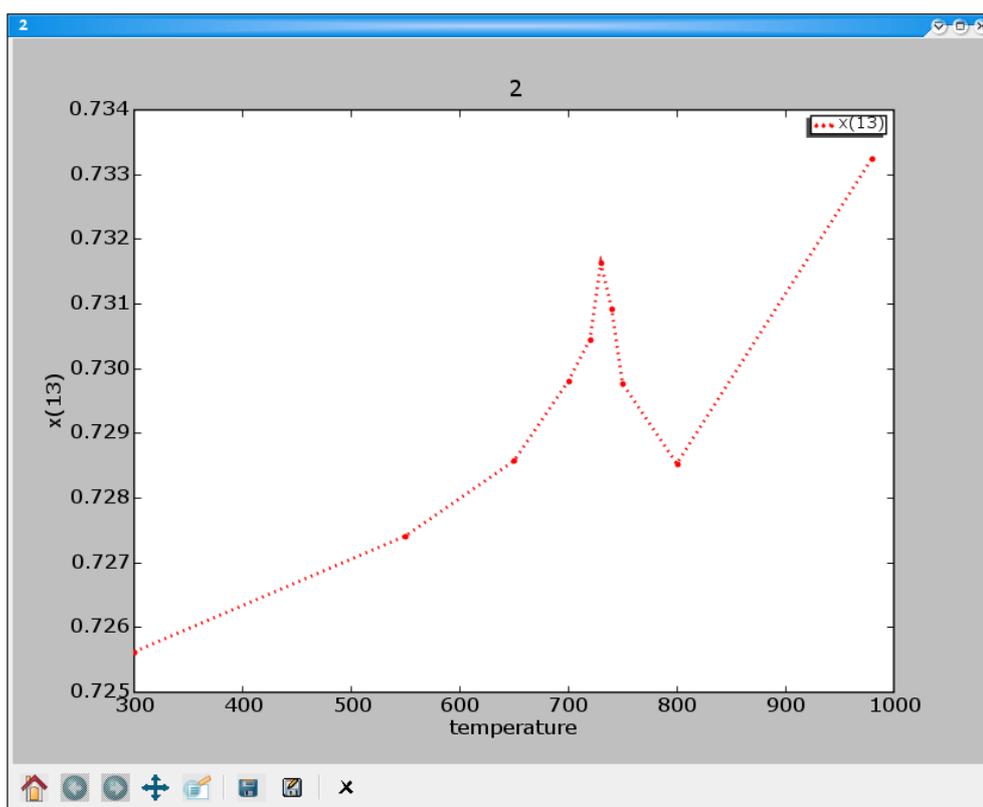
**Figure 1.** The PDFgui plot window, plots of the  $\text{LaMnO}_3$  PDF fit at 300 K and the data at all temperatures, and the refined structure at 300 K.

### 3.3. Example

The capabilities of PDFgui are demonstrated here on an example fitting of a temperature series of neutron PDF data from  $\text{LaMnO}_3$ . The specifics of using PDFgui and PDFfit2 are detailed in the respective user manuals, available with the code.

$\text{LaMnO}_3$  has a perovskite structure consisting of corner-shared  $\text{MnO}_6$  octahedral units [25, 26]. At room temperature this material takes on orthorhombic symmetry (space group  $Pbnm$ ) where the Jahn–Teller (JT) distorted  $\text{MnO}_6$  octahedra contain four short and two long Mn–O bonds and the JT long-bonds are ordered in space in a checker-board fashion [13]. As temperature is increased through 750 K, a structural phase transition occurs to a pseudo-cubic phase, without a change in space group [27]. Using neutron powder diffraction data collected at the NPDF diffractometer [28] at the Lujan Center at Los Alamos National Laboratory, PDFgui was used to create a temperature series fit to investigate the orthorhombic to pseudo-cubic phase transition. More detail about this analysis can be found in [29] and [30]. The average crystallographic structure for  $\text{LaMnO}_3$  was used as a starting point for the fits. In practice, this can be done by typing in the asymmetric unit, or loading it from a CIF-format file and letting the program expand it.

Each fit in the series was part of the same project and configured identically. The La and Mn atoms were constrained to have isotropic thermal factors, as were the axial and polar oxygen atoms. Using the PDFfit2 constraint mechanism allows one to include explicitly known



**Figure 2.** The full PDFgui plot window showing the fractional  $x$ -coordinate of an axial oxygen versus temperature.

correlations between the physical parameters of the fit and leads to less uncertainty in the resulting refinement. To speed up the convergence of the fits, the starting values used for a given temperature were taken from the converged values of the previous temperature in the series. Chaining fits together in such a way places the fitting parameters in the basin of attraction for the minimum in the parameter space. Figure 1 shows the fit organization and plot interface of PDFgui, a stacked plot of all of the project data, a plot of the 300 K fit results, and the refined structure from the 300 K fit.

One of the goals of this series of fits was to track the orientation of the  $\text{MnO}_6$  octahedra. A major strength of PDFgui is the ability to plot any structure variable or a parameter used in a fit. This allows users to easily and creatively investigate the complex correlations among the fit parameters, without having to manually mine data out of output files. An example of this can be seen in figure 2. The figure shows a screen shot of a PDFgui plot window displaying the refined fractional  $x$ -coordinate of one of the planar oxygens. The expected structural phase transition is clearly indicated by a spike in the value of this parameter at the transition temperature.

#### 4. Availability

PDFfit2 and PDFgui are open source and distributed under a BSD license. They run on Windows, Linux, and all major Unix systems. The source code is freely available. For more

information, please contact professor Simon Billinge (billinge@pa.msu.edu) or consult the web page [31]. News of updates and releases will be posted at this website and on the total-scattering e-mail list. Instructions for joining the e-mail list can be found on the web page.

## Acknowledgments

We would like to acknowledge HyunJeong Kim, He Lin and Wenduo Zhou for help and useful discussions during the development and all the Billinge group members for help in testing. PDFgui was developed as part of the Distributed Analysis of Neutron Scattering Data (DANSE) project, which is funded by the US National Science Foundation (NSF) under DMR-0520547. The PDFfit2 development was funded by the NSF under contract DMR-0304391.

## References

- [1] Gilbert B, Huang F, Zhang H, Waychunas G A and Banfield J F 2004 *Science* **305** 651–4
- [2] Billinge S J L and Kanatzidis M G 2004 *Chem. Commun.* **749–60**
- [3] Lin H, Božin E S, Billinge S J L, Quarez E and Kanatzidis M G 2005 *Phys. Rev. B* **72** 174113
- [4] Vensky S, Kienle L, Dinnebier R E, Masadeh A S, Billinge S J L and Jansen M 2005 *Z. Kristallogr.* **220** 231–44
- [5] Petkov V, Billinge S J L, Vogt T, Ichimura A S and Dye J L 2002 *Phys. Rev. Lett.* **89** 075502
- [6] Billinge S J L, McKimney E J, Shatnawi M, Kim H, Petkov V, Wermeille D and Pinnavaia T J 2005 *J. Am. Chem. Soc.* **127** 8492–8
- [7] Tucker M G, Squires M P, Dove M T and Keen D A 2001 *J. Phys.: Condens. Matter* **13** 403–23
- [8] McGreevy R L and Pusztai L 1988 *Mol. Simul.* **1** 359–67
- [9] Egami T and Billinge S J L 2003 *Underneath the Bragg Peaks: Structural Analysis of Complex Materials* (Oxford: Pergamon, Elsevier)
- [10] Proffen T and Billinge S J L 1999 *J. Appl. Crystallogr.* **32** 572–5
- [11] Billinge S J L 1998 *Local Structure from Diffraction* ed Billinge S J L and M F Thorpe (New York: Plenum) p 137
- [12] Tucker M G, Goodwin A L, Dove M T, Keen D A, Wells S A and Evans J S O 2005 *Phys. Rev. Lett.* **95** 255501
- [13] Proffen T, DiFrancesco R G, Billinge S J L, Brosha E L and Kwei G H 1999 *Phys. Rev. B* **60** 9973
- [14] Warren B E 1990 *X-ray Diffraction* (New York: Dover)
- [15] Wright A C 1998 *Glass Phys. Chem.* **24** 148–79
- [16] Jeong I-K, Proffen T, Mohiuddin-Jacobs F and Billinge S J L 1999 *J. Phys. Chem. A* **103** 921–4
- [17] McGreevy R L 2001 *J. Phys.: Condens. Matter* **13** R877–913
- [18] Toby B H and Egami T 1992 *Acta Crystallogr. A* **48** 336–46
- [19] Juhas P, Cherba D M, Duxbury P M, Punch W F and Billinge S J L 2006 *Nature* **440** 655–8
- [20] <http://www.wxpython.org>
- [21] Painter J and Merritt E A 2004 *J. Appl. Crystallogr.* **37** 174–8
- [22] <http://matplotlib.sourceforge.net>
- [23] Li J 2003 *Modelling Simul. Mater. Sci.* **11** 173–7
- [24] DeLano W L 2002 *The PyMOL User's Manual* (San Carlos, CA: DeLano Scientific) <http://www.pymol.org>
- [25] Elemans J B A A, Van Laar B, Van der Veen K R and Loopstra B O 1971 *J. Solid State Chem.* **3** 238–42
- [26] Rodriguez-Carvajal J, Hennion M, Moussa F, Moudén A H, Pinsard L and Revcolevschi A 1998 *Phys. Rev. B* **57** R3189–92
- [27] Chatterji T, Fauth F, Ouladdiaf B, Mandal P and Ghosh B 2003 *Phys. Rev. B* **68** (5) 052406
- [28] Proffen T, Egami T, Billinge S J L, Cheetham A K, Louca D and Parise J B 2002 *Appl. Phys. A* **74** s163–5
- [29] Qiu X, Proffen Th, Mitchell J F and Billinge S J L 2005 *Phys. Rev. Lett.* **94** 177203
- [30] Božin E S, Qiu X, Schmidt M, Paglia G, Mitchell J F, Radaelli P G, Proffen T and Billinge S J L 2006 *Physica B* **385/386** 110–2
- [31] <http://www.diffpy.org>