

PDFgui user guide

1.0 Beta release
04 May 2007

C. L. Farrow, P. Juhás, J. W. Liu, D. Bryndin, E. S. Božin,
J. Bloch, Th. Proffen, and S. J. L. Billinge

Acknowledgements

PDFgui was developed as part of the Distributed Analysis of Neutron Scattering Data (DANSE) project which is funded by NSF under DMR-0520547. The PDFfit2 development was funded by NSF under contract DMR-0304391. The program is part of the DiffPy and DANSE open-source projects at Michigan State University and is available subject to the conditions and terms laid out below. Several of the examples in the tutorial part were made possible and benefited from samples synthesized by J. F. Mitchell, and from data collected and processed by M. Schmidt, P. G. Radaelli, and X. Qiu.

If you use this program to do productive scientific research that leads to publication, we ask that you acknowledge use of the program by citing the following paper in your publication:

1. C. L. Farrow, P. Juhás, J. W. Liu, D. Bryndin, E. S. Božin, J. Bloch, Th. Proffen and S. J. L. Billinge, PDFfit2 and PDFgui: Computer programs for studying nanostructure in crystals, *J. Phys: Condens. Matter*, in press (2007).

Copyright © 2006, 2007, trustees of Michigan State University, all rights reserved.

For more information please visit <http://www.diffpy.org> or contact Prof. Simon Billinge at billinge@pa.msu.edu.

Preface

Using PDFGui and PDFfit2

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of MICHIGAN STATE UNIVERSITY nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

1	Introduction	1
1.1	PDFfit2	1
1.2	PDFgui	2
1.2.1	Design principles	2
1.2.2	Capabilities	2
1.3	Availability	3
1.4	Installation	3
	Unix/Linux/MAC	3
	Windows	4
	Building from Subversion repository	4
1.5	Community	5
2	Quick start	6
2.1	PDFgui layout	6
2.2	Creating a simple fit using a preexisting structure file	6
3	Examples and tutorials	17
3.1	Building structure model using crystal symmetry	17
3.2	Calculating PDF from a structure	19
3.3	Sequential fitting	21
3.3.1	Multistage fitting	21
3.3.2	Sequential fitting of incremental r-series	22
3.3.3	Sequential fitting of temperature series	24
3.3.4	Sequential fitting of doping series	26
3.4	Nanoparticle structure	29
4	Extras	32
4.1	PDF plotting	32
4.2	Displaying the structure	33
4.3	Advanced usage and special needs	33
5	PDFgui reference sheets	35
5.1	PDFgui hot keys	35
5.2	List of PDFfit2 variables	35
	New Variables	35
	Renamed Variables	36
	Preserved Variables	37
	PDF Peak Width	37
	Index	38

1 Introduction

PDFgui is a graphical interface built on the PDFfit2 engine, which is a program as well as library for real-space refinement of crystal structures based on the atomic pair distribution function (PDF) method. PDFgui organizes fits and simplifies many data analysis tasks, such as configuring and plotting multiple fits. PDFfit2 is capable of fitting a theoretical three dimensional 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 symmetry requirements of an arbitrary space group. The PDFfit2 engine is written in C++ and accessible via Python, allowing it to inter-operate with other Python programs.

1.1 PDFfit2

PDFfit2 is a major upgrade to PDFfit, 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 expressing structure variables as simple functions of fitted parameters. PDFfit structure variables include lattice constants, data and phase scale factors, atomic site occupation, anisotropic atomic displacement parameters (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 (<http://www.python.org>). 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.

1.2 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.

1.2.1 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 multitasking 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 (<http://www.wxPython.org>), the Python package for wxWidgets, 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.

1.2.2 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 can be configured to run fits on a remote computer. This uses the secure-shell protocol (ssh2) so that the connections are stable and secure. In this distributed mode the core fitting computation does not compete for system resources with the graphical functions, which speeds up the fit over fast network connections and gives faster user response at the console.

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 atomic 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 show a local distortion. The code for space group definitions was provided by the Python Macromolecular Library (mmLib, <http://pymmlib.sourceforge.net>). PDFgui also supports supercell expansion of a normal unit cell.

PDFgui uses the matplotlib (<http://matplotlib.sourceforge.net>) Python package for 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 meta-data (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 (<http://164.107.79.177/Archive/Graphics/A/>) for 3D visualization. The modular nature of PDFgui allows for other alternatives in the future, such as PyMOL (<http://pymol.sourceforge.net>).

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 fit over different 2θ -ranges of the PDF data.

1.3 Availability

PDFfit2 and PDFgui are open source and distributed under a BSD license. They run on Windows, Mac OS, 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 <http://www.diffpy.org>. News of updates and releases will be posted at this website and on the diffpy-users group at <http://groups.google.com/group/diffpy-users>.

1.4 Installation

PDFgui is included as a part of DiffPy, a Python library for structure analysis from diffraction data. The latest version of DiffPy can be obtained from <http://www.diffpy.org>. Refer to the section corresponding to your operating system for specific installation instructions.

Unix/Linux/MAC

For unix-like operating systems DiffPy is provided as a source archive, which needs to be built and installed. A successful build of DiffPy requires several libraries and applications. Before proceeding any further, make sure the packages below are installed. All of these packages should be available as RPM files for recent Linux systems, and it is then best to install them using the respective package manager such as 'apt-get' or 'yum'. In case an RPM package is not available, refer to the corresponding project page for other installation options.

Required packages:

python <http://www.python.org> Python
interpreter, minimum version 2.3

python-dev
<http://www.python.org> development files for Python

gsl-devel
<http://www.gnu.org/software/gsl> GNU Scientific Library

numpy <http://numpy.scipy.org> Numerical Python, minimum version 0.9.8

wxPython <http://wxpython.org> GUI toolkit for Python, minimum version 2.6, recommended 2.8

python-matplotlib
<http://matplotlib.sourceforge.net> plotting library for Python, minimum version 0.90

Once all of these are in place and working, download the latest ‘.tgz’ archive of the DiffPy sources from <http://www.diffpy.org>. The version number is embedded as part of the file name. Unpack the sources using the command

```
gzip -d -c diffpy-VERSION.tgz | tar xf -
```

where you need to change VERSION to match the name of the archive file. This will create a new ‘diffpy-VERSION’ directory. Change to this directory and execute as a root

```
python setup.py install
```

This installs DiffPy to a default location in python ‘site-packages’ directory and executable scripts pdfgui and pdfit2 to ‘python-prefix/bin/’. If you don’t have root access, you can install DiffPy to other locations by adding suitable options to setup.py script. For a list of available options, see ‘python setup.py --help’ or ‘python setup.py install --help’.

Windows

DiffPy is provided as self-installing archive for Windows systems. Download the latest ‘diffpy-VERSION.exe’ file, launch the executable and follow the prompts. That’s all - PDFgui and PDFit2 should be now ready for use.

Building from Subversion repository

The latest development sources of DiffPy are available in a public Subversion code repository. To obtain the latest code, run

```
svn checkout svn://svn@danse.us/diffraction/diffraction/diffpy
```

You need to have **svn**, a subversion client, installed on your computer. In addition a GNU documentation system Texinfo (<http://www.gnu.org/software/texinfo/>) is required to compile users manual.

Otherwise the installation procedure is the same as for Unix-like operating systems. Change to the ‘diffpy’ directory and execute

```
python setup.py install
```

The checked-out diffpy directory can be kept around for future updates. To bring it up to date with the latest sources, one just needs to execute

```
svn update
```

inside this directory.

1.5 Community

There are several Google groups for support and development of PDFgui and other DiffPy software. Visit the links below for message archives or instructions on subscription and posting.

diffpy-users <http://groups.google.com/group/diffpy-users>

Help on usage of PDFgui, PDFfit2 and other DiffPy packages. This group should become a knowledge base of PDFgui user tips, tricks and troubleshooting. Feel free to ask your question here.

diffpy-dev <http://groups.google.com/group/diffpy-dev>

For discussions about development and changes of PDFgui, PDFfit2 and DiffPy library in general.

2 Quick start

In this chapter the PDFgui layout is briefly described, followed by a simple tutorial example, spanning the gui functionality and aimed at novice users. Users familiar with the basics can proceed to [Chapter 3 \[Examples and tutorials\]](#), page 17, or use [Chapter 5 \[PDFgui reference sheets\]](#), page 35. All the files used in examples are available in the tutorial subdirectory relative to this document.

2.1 PDFgui layout

Once PDFgui is invoked, a PDFgui window comes up. Its layout consists of a menu bar, a tool bar, and a set of four panes. The menu bar contains drop-down menus that provide various aspects of PDFgui functionality. The tool bar features icons for commonly used operations: starting a new project or opening an existing one, saving a project, executing a refinement or calculation, stopping a refinement or calculation, and making a quick plot of a refinement. The four panes consist of fitting, plotting, current action, and PDFfit2 output. The fitting pane is home for Fit Tree and is used in setting up a fit protocol. The Plot Control panel serves the user's needs for graphically displaying the fits, as well as various fit-related parameters. The content of the Current Action panel changes as the refinement is being set up. It is a functional panel through which the user configures the fit attributes, sets the fit constraints, reviews the fit settings, displays fitting results, and also carries out other setup steps. The progress of the PDFfit2 program is displayed in the PDFfit2 output panel. All panels except the Current Action panel are dockable windows that can be dragged across the screen, resized and arranged to accommodate the individual visual needs of the user. The window layout can also be controlled via the "View" drop-down menu on the menu bar. An important part of the PDFgui operativity is also conveniently available through mouse operations such as select and right-click.

2.2 Creating a simple fit using a preexisting structure file

The purpose of this example is to familiarize the novice user with the PDFgui basics. The goal is to create a simple fit of Ni PDF from a preexisting structure file, associate a dataset with the refinement, configure and execute a refinement, and display the result. Files to be used in this exercise are *Ni.stru* and *Ni-xray.gr*. To start, invoke PDFgui.

First a new fit needs to be established. In the Fit Tree pane right-click the mouse button, and select **New Fit**. This will generate a new fit called Fit 1 in the Fit Tree. The Current Action panel will change accordingly, with generic appearance reflecting the fact that the fitting three is not populated with neither structure nor data. Note that the same action could be achieved by selecting the **New Fit** option from the "Fits" drop-down menu on the menu bar.

The next step in populating the fitting tree is to load structural model. This is achieved through a simple sequence of actions: first select the fit icon in the Fit Tree pane. Then right-click the mouse, invoking a drop-down menu. Select **Insert Phase**, which will modify the current action panel. The same action could be performed from the "Phases" drop-down menu on the menu bar by selecting **New Phase**. In the current action panel options are offered to load a structure from a file or to generate it from scratch. In this exercise a preexisting structure file *Ni.stru* is to be loaded.

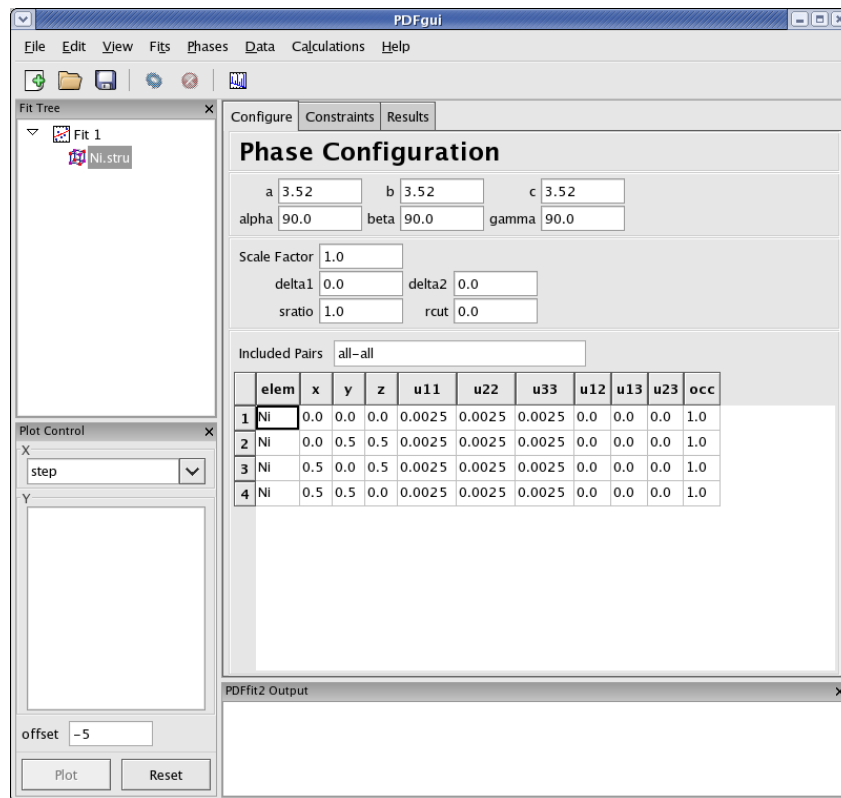


Figure 2.1: PDFgui window with Fit Tree and Plot Control panels to the left and Current Action panel to the right. The Fit Tree panel features the current fit and the loaded Ni structure which is selected. The current action panel displays phase configuration within the “Configure” tab.

Figure 2.1 shows the PDFgui appearance at this stage of the exercise. The Current Action panel has three tabs, “Configure”, “Constraints”, and “Results” that could be selected using mouse. These will be returned to later. The “Configure” panel displays configuration information from the structure file. The top portion contains lattice parameters, phase scale factor, and a set of parameters intended to be used to account for correlated atomic motion effects that typically sharpen the nearest neighbor PDF peak. These are *delta1*, *delta2*, *sratio*, and *rcut*. Their usage is explained later. The bottom part of the panel contains standard unit cell content related information such as atomic species, their fractional coordinates, anisotropic ADPs, and site occupancies.

The next step is to load an experimental data set to be fit. To achieve this Fit 1 is selected on a Fit Tree, and a drop down window is invoked by a right-click of a mouse. From the menu **Insert Data Set** is chosen. The same action could be performed through a menu bar “Data” drop-down menu and selecting **New Data Set**. The Current Action panel changes accordingly, giving an option for a data set to be loaded from a file. In this exercise Ni PDF data obtained using synchrotron x-ray radiation collected at 6-ID-D at the Advanced Photon Source is used. This is contained in a file *Ni-xray.gr*, which is to be

loaded. Note that among the exercise files there is also a file *Ni-neutron.gr*, obtained using neutron radiation at the GPPD diffractometer at the IPNS facility at the Argonne National Laboratory. Both x-ray and neutron datasets were collected at 300 K.

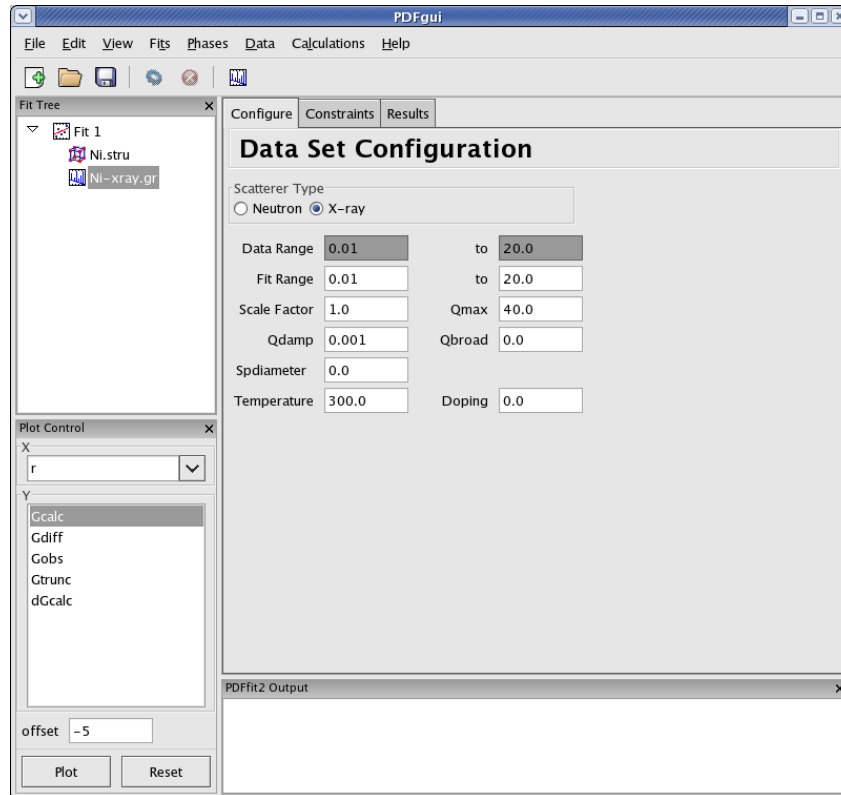


Figure 2.2: Appearance of a PDFgui window after a PDF dataset is loaded. The Fit Tree panel features the current fit, loaded Ni structure, and loaded Ni PDF dataset, $G(r)$, which is selected. The Current Action panel displays data set configuration within the “Configure” tab.

Figure 2.2 shows the PDFgui appearance at this stage of the exercise. The Current Action panel has three tabs, “Configure”, “Constraints”, and “Results” that could be selected using mouse. These will be returned to later. The “Configure” panel displays configuration information from the data file. It should be noted that depending on the software used to prepare the experimental PDF from the raw data, the file may (or may not) contain meta data reflecting the experimental conditions and configuration. For example, software PDFgetX2 and PDFgetN used to prepare PDFs from x-ray and neutron total scattering experiments respectively, are supplying meta data in the header of the data file. The PDFgui is picking up this information and filling the appropriate fields in the Current Action panel for data set configuration. Caution should be exercised by the user to verify that these data indeed correspond to the experimental conditions. In the present example, x-ray radiation is used, and so the x-ray selection is turned on for the scatterer type. The data range, fit range, data scale factor, maximum Q value used in Fourier transform to obtain the ex-

perimental PDF, spherical form factor diameter (used for nanoparticle samples, to account for the PDF signal dampening due to the finite particle size) and the experiment specific parameters are displayed. These comprise parameters describing experimental resolution effects, Qdamp and Qbroad, and experimental conditions, such as temperature and doping (used for bookkeeping and for parametric plots). If no meta data are present in a data file, this information should be supplied by the user. Note also that the changes occurred at this stage in the Plot Control panel, allowing user to plot the data. This is achieved by selecting an observed $G(r)$ and clicking on a plot button that will invoke a plot. Since no fitting has occurred so far, an attempt to plot calculated PDF profile or a difference yields a blank plot.

Having specified the initial structure to be refined, and the data set to be fit, one proceeds to the refinement setup. This is carried out through adjusting the initial parameters and variables, as well as setting up the constraints. The adjustments and constraint setup are done on both the experimental data and the refined structure levels, toggling between the corresponding “Configure” and “Constraints” tabs. In the present example the data related setup will be done first.

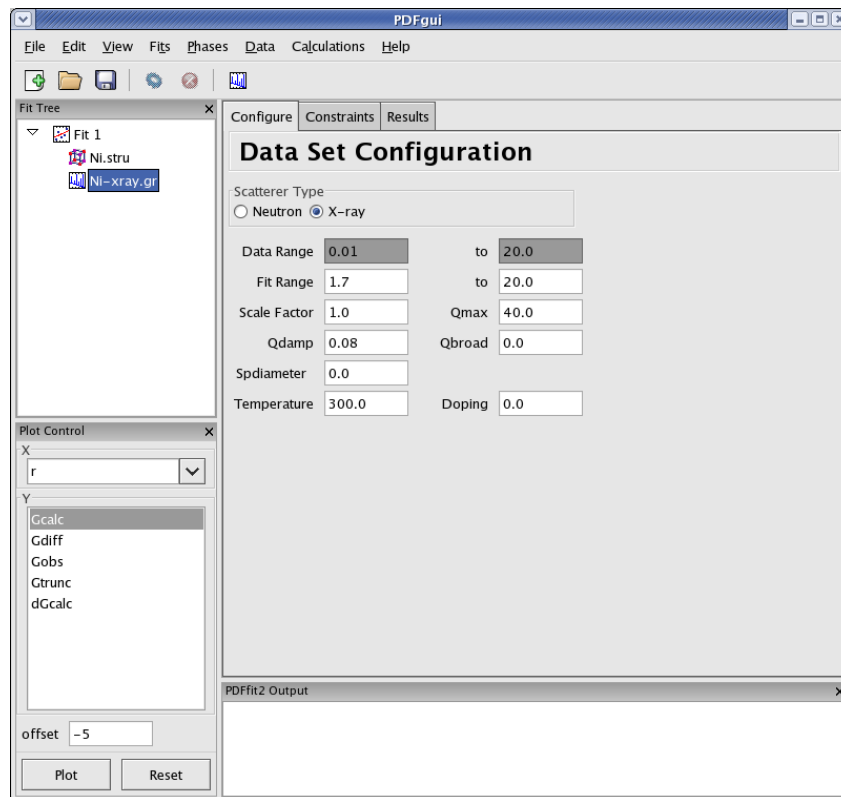


Figure 2.3: Adjusting data set related configuration.

Click on the *Ni-xray.gr* in the Fit Tree. In Figure 2.3 “Data Set Configuration” panel is shown. At this stage we adjust the fitting range, as well as other parameters that reflect the experimental conditions, as appropriate. Initial values for refined parameters can be set and

reviewed when the fit is selected on the Fit Tree, on the Parameters tab. Since there is no physical information in the region of r below the nearest neighbor PDF peak position, and since this region is often affected by noise and experimental artifacts, it is wise to exclude it from fitting. We therefore set the value of the lower boundary of the fitting range to 1.7. In addition, we set $Qdamp$ parameter to a more realistic starting value of 0.08. This parameter is typically obtained through a conventional calibration process at each PDF experiment using a standard sample such as Ni or Si. Next, we select the “Constraints” tab, and type @1 into the “Scale Factor” edit box. This will assign refinement parameter 1 to the data scale factor. Note that this is the syntax used for assigning the refinement parameters in PDFfit2 engine. Similarly, assign parameter 2 to $Qdamp$ by inserting @2 into the appropriate edit box. This is illustrated in Figure [Figure 2.4](#).

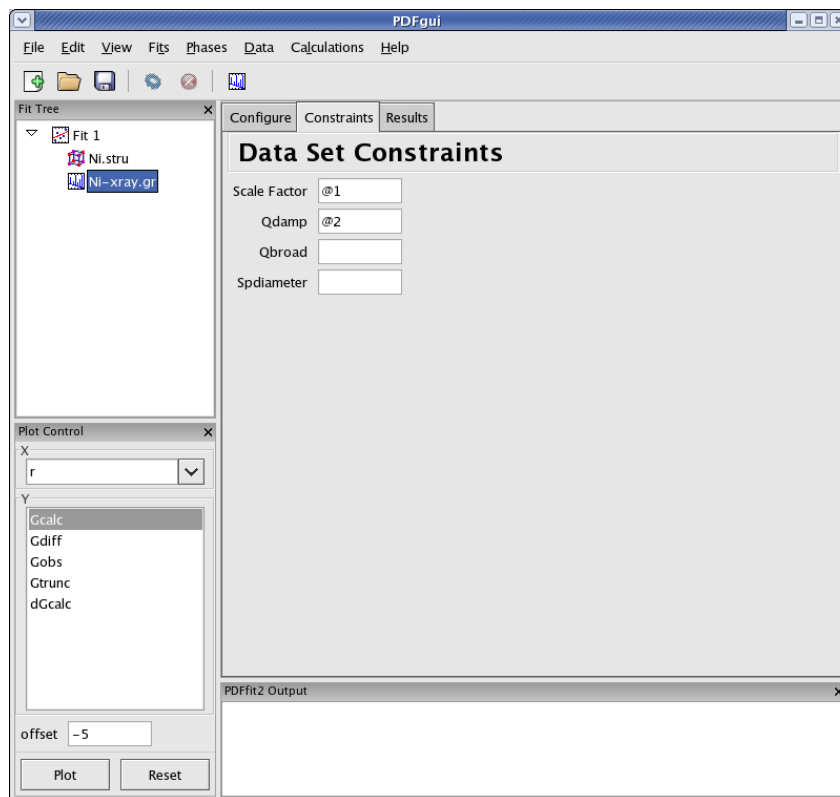


Figure 2.4: Setting up the refinement parameters and constraints.

Further, we set constraints related to the structural model, by selecting the *Ni.stru* on the Fit Tree, adjusting the initial parameter values if necessary (not done here), and proceeding to the “Constraints” tab. We assign the refinement parameter 3 to all three lattice constants, a , b , and c , reflecting the fact that the structure is cubic. Isotropic ADPs are assigned to all Ni atoms in the refined cell as refinement parameter 4. This can conveniently be done by highlighting all the appropriate boxes together, and type only once @4. The outcome is shown in Figure [Figure 2.5](#).

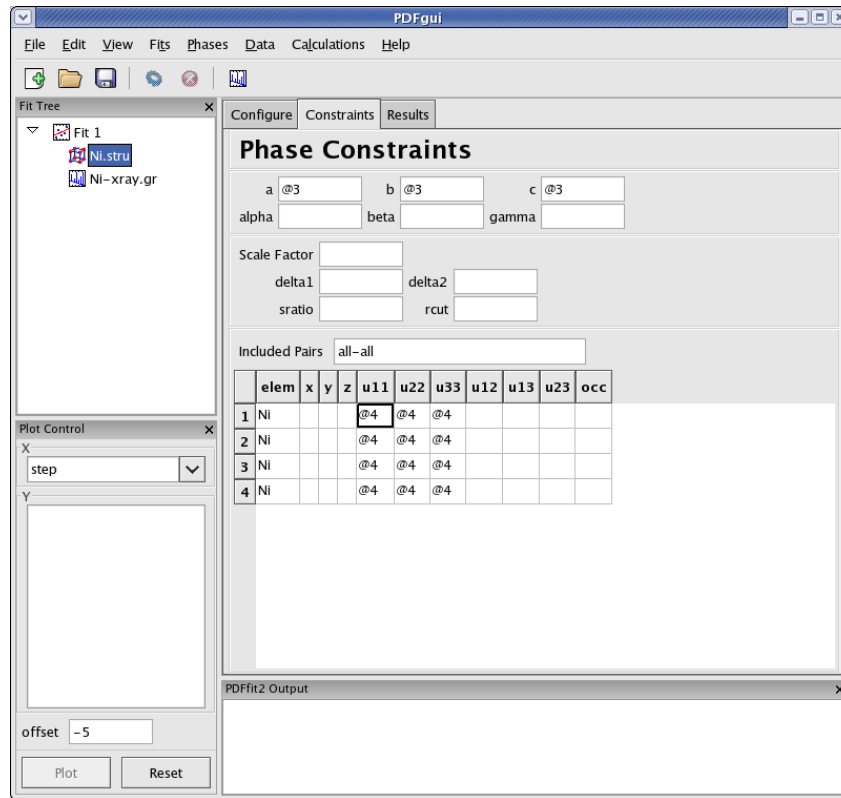


Figure 2.5: Setting up the refinement parameters and constraints.

It should be noted that as a part of the PDFfit2 syntax a refined variable can also be defined as a math expression $f(@n1, @n2, @n3, \dots)$ where $@n1$ stands for fitted parameter, and $n1, n2, \dots$ are arbitrary positive integers enumerating the parameters. This allows simple linking of related variables. For example, since cell lengths a , b , and c are all expressed as $@3$, the refined structure will remain cubic. Also note that the enumeration of the parameters can be arbitrary, enumeration does not have to follow any particular order. The quantities within a fit that are enumerated with the same number will be assigned the same parameter, hence caution should be exercised to avoid unintentional assignment of the same parameter to physically different quantities. The automatic parameter assignment is done in such a way as to disallow for such situations to happen. If assignment is done in part manually, in particular for complex setups, it is recommended to verify that the parameter assignment is done correctly.

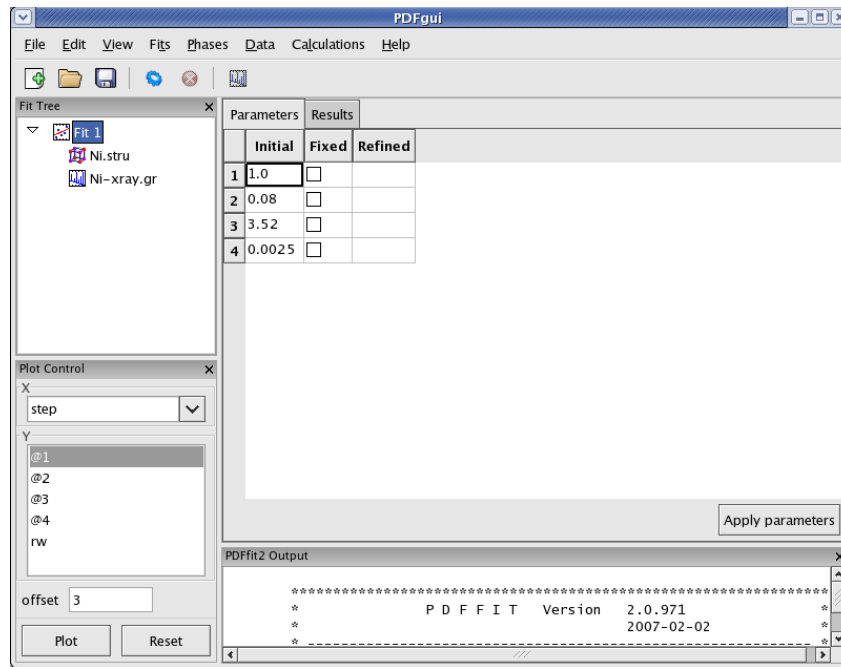


Figure 2.6: Reviewing the fit parameters and conditions.

The setup for the present example is now completed. By selecting Fit 1 on the Fit Tree the Current Action window shows “Parameters” tab, which displays the used refinement parameters for review, [Figure 2.6](#). It shows the initial values, and allows for updates and for refinement parameters to be released or fixed depending on whether the corresponding “Fixed” box is unchecked or checked, respectively. The “Refined” column, which is currently empty, will be populated with the refined values of the parameters. If the initial parameter values are to be changed, using “Apply parameters” button will update the values of the parameters on all relevant setup tabs. The refinement is executed by clicking the “gear” icon on the tool bar. The refinement could be stopped by clicking on the “stop” icon on the tool bar. During the refinement the refinement progress will be directly reported in the PDFfit2 Output panel. This is illustrated in [Figure 2.7](#).

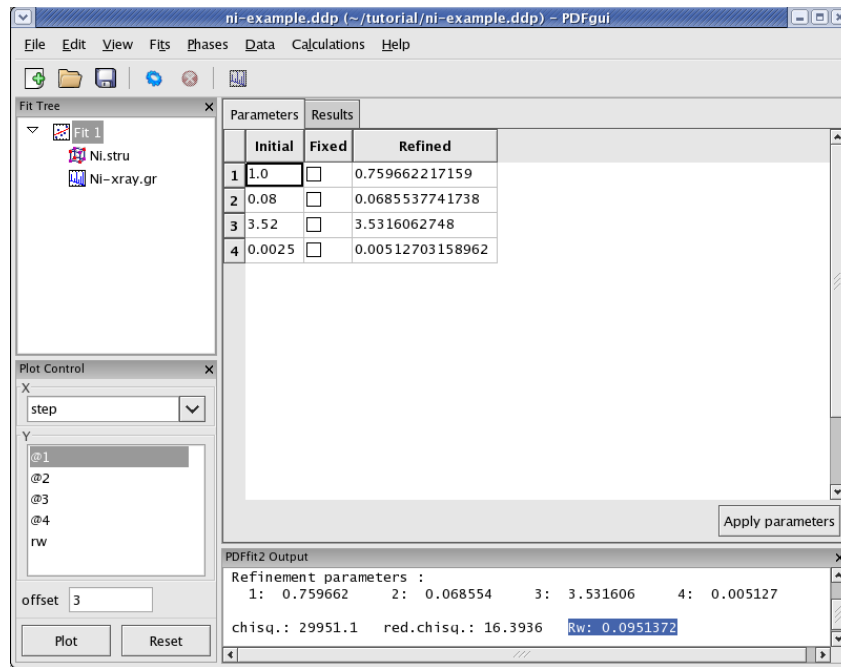


Figure 2.7: While the refinement is running, the refinement progress is displayed in the PDFfit2 Output panel.

After the fitting is completed, the fit summary is provided in the “Results” tab of the Current Action panel when Fit 1 is selected on the Fit Tree. Similarly, if “Results” tab is displayed when *Ni.stru* or *Ni-xray.gr* are selected, values of all corresponding refined parameters for the converged model are displayed.

If the fit results are acceptable, one or more refined values could be copied to become new initial parameters for possible further refinement, where appropriate. This can be done when Fit 1 is selected on the “Parameters” tab. By highlighting refined parameters to be copied, and right-clicking on mouse a drop-down menu is invoked. Selecting **Copy Refined To Initial** will perform the desired initial parameter update, as illustrated in Figure 2.8.

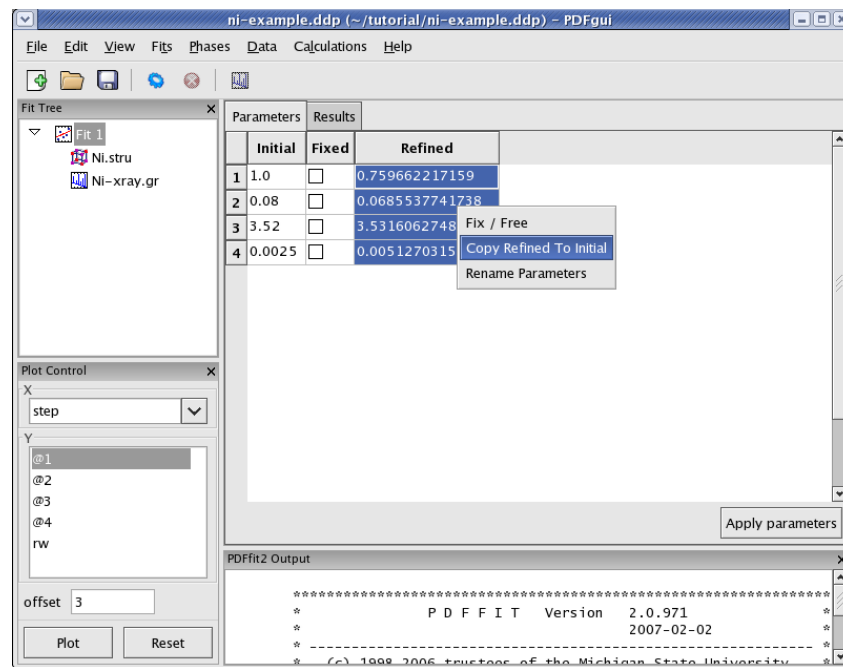


Figure 2.8: Updating the set of initial values of refined parameters.

The fit can be displayed graphically by clicking at the quick plot icon on the tool bar. Alternatively, desired items to be displayed can be selected from the Plot Control and plotted on the screen. Depending on whether the structure or the data are selected on the Fit Tree, either refined structural parameters or the experiment related parameters and fit could be plotted. An example of the fit display is shown in [Figure 2.9](#), and a selected parameter plot vs iteration step in [Figure 2.10](#). If the plotting window remains open while the fitting is in progress, the content will be updated after each fit iteration.

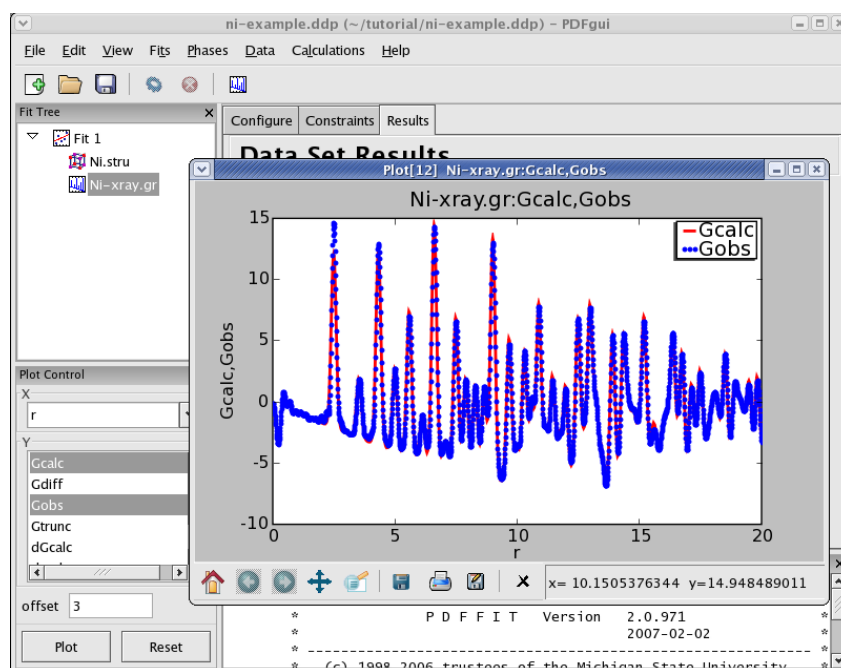


Figure 2.9: An example of PDFgui plotting capabilities: displaying a fit.

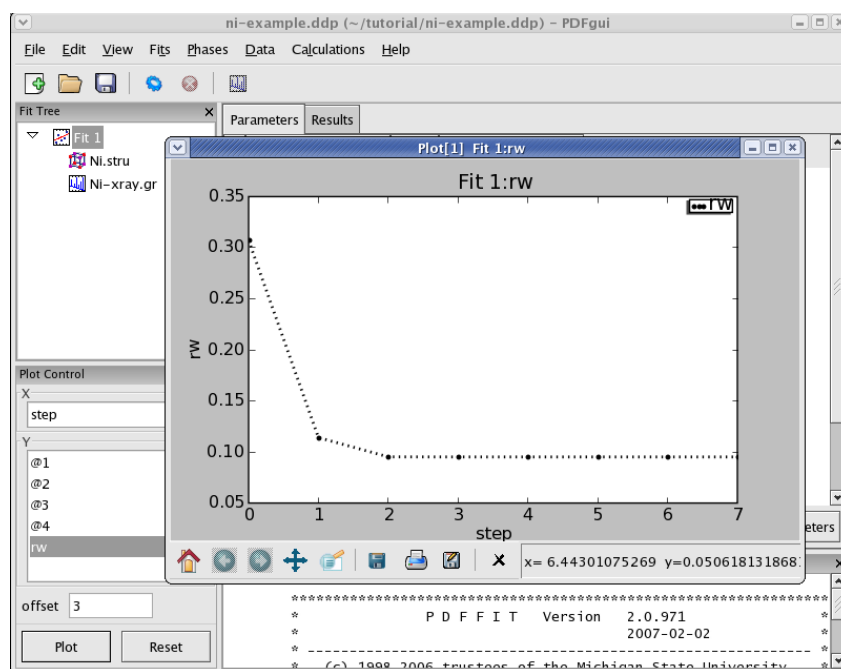


Figure 2.10: An example of PDFgui plotting capabilities: displaying a parameter.

PDFgui is furnished with “Journal” capability, which can be found under the “View” menu, and represents a convenient way for generating, storing, and using for further reference of project related notes, as illustrated in Figure 2.11. These notes can be exported as a text file, or saved along with the entire project.

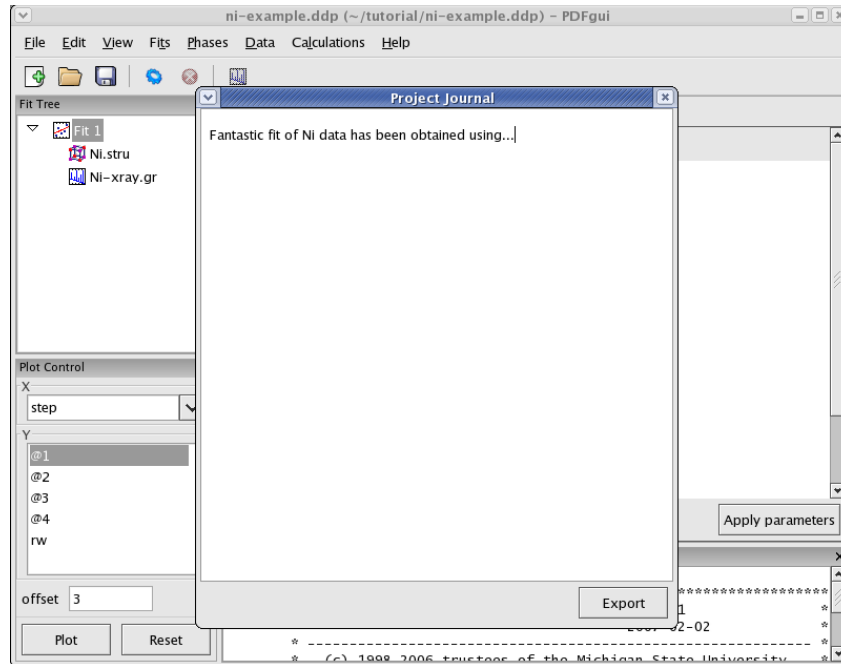


Figure 2.11: Using “Journal” feature can be a convenient way for taking notes.

The project can be saved at any stage in its present configuration through choice of **Save Project as** or **Save Project** as appropriate from the “File” drop-down menu. The PDFgui project file has “ddp” extension. In addition to saving a project, various parts of the project, both structure related and data related, can be exported to external files by making an appropriate selection from the “Phases” and “Data” drop-down menus. The phases (starting or converged) are saved as standard PDFFIT “.stru” files. The model PDF profile can be exported through “Data” menu and will be saved as a five-column “.fgr” file. The first four columns are r , $G(r)$, dr , and $dG(r)$, and the fifth column is the difference curve between the data and the model. Note that the model PDF and the difference are only calculated within the user-specified fitting range.

PDFgui allows for PDFfit2 scripts written to operate refinements in the expert mode to be imported into the gui. This option, along with the option of exporting gui generated scripts and exporting the refinement results is available through the “Fits” menu on the menu bar.

3 Examples and tutorials

In this chapter we present series of examples and tutorials aimed at users already comfortable with the gui, to provide training in advanced gui features designed for most common modeling situations, such as building the structure from scratch, calculating the PDF based on a given structure, linking two fits, and creating and executing a series of fits on a sequence of data sets.

3.1 Building structure model using crystal symmetry

The purpose of this example is to demonstrate to the user the symmetry expansion capabilities of PDFgui. The goal is again to create a simple fit of Ni PDF, but this time from scratch rather than from a preexisting structure file. Further goal is to associate a dataset with the refinement, configure and execute a refinement, and display the result. The focus will be on the symmetry expansion features only, and the steps that are described earlier in [Section 2.2 \[Simple fit\], page 6](#) are left for the user to recreate. File to be used in this exercise is *Ni-xray.gr*. To start, invoke PDFgui.

To begin, a new fit again needs to be established by right-clicking the mouse button on the Fit Tree pane, and selecting **New Fit**. This will generate a new fit called Fit 1 as in the previous example. The next step in populating the fitting tree is to introduce a structural model. This time a new phase will be added and the model built up from scratch. Select the fit icon in the Fit Tree pane. Then right-click the mouse, invoking a drop-down menu. Select **Insert Phase**, which will modify the current action panel. The same action could be performed from the “Phases” drop-down menu on the menu bar by selecting **New Phase**. In the current action panel options are offered to load a structure from a file or to generate it from scratch. Select “New” to build up structure from scratch. The Fit Tree will be populated with a new item “New Phase”. Rename this phase to “Ni fcc”. The Current Action panel now displays default phase configuration with “dummy” values and no atoms. Edit the lattice parameters and set them to reflect the symmetry, in this case set the lengths to 3.52 Angstroms and all the angles to 90 degrees. The “Tab” keyboard shortcut feature can be used for moving quickly from one form-field to the next. To introduce new atoms right-click with your mouse onto *elem* tab in the Current Action panel, and select desired number of atoms in the asymmetric unit via the dialog box that pops up. In Ni case select one row only. After action is invoked, a new row will show up in the table. This can also be performed by hitting “Control”, “Shift” and “+” simultaneously. Type in *Ni* for the element name. It is important that the element name is typed in correctly, as this is further used to search a database for the scattering characteristics of that site. Further, highlight the three diagonal ADPs, and type in the initial value 0.0025.

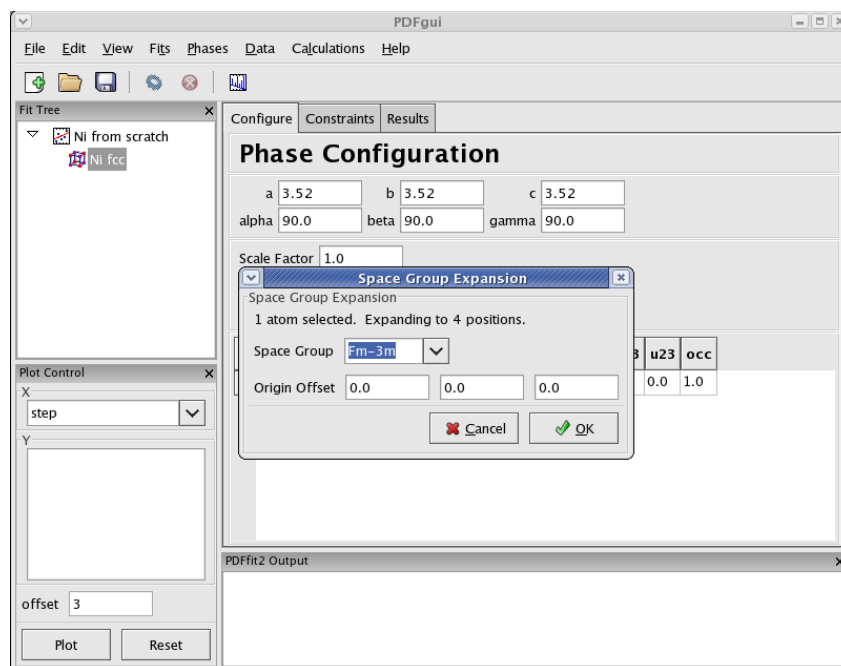


Figure 3.1: Expanding the unit cell using space group information.

Now right-click with the mouse on the element name, and select “Expand space group...” option. A dialog box will appear, as shown in [Figure 3.1](#), requesting space group information and the choice of origin. Choose “Fm-3m” and hit enter. You can also type “Fm-3m” or “225”, the space group number. This will expand the unit cell accordingly showing 4 Ni sites.

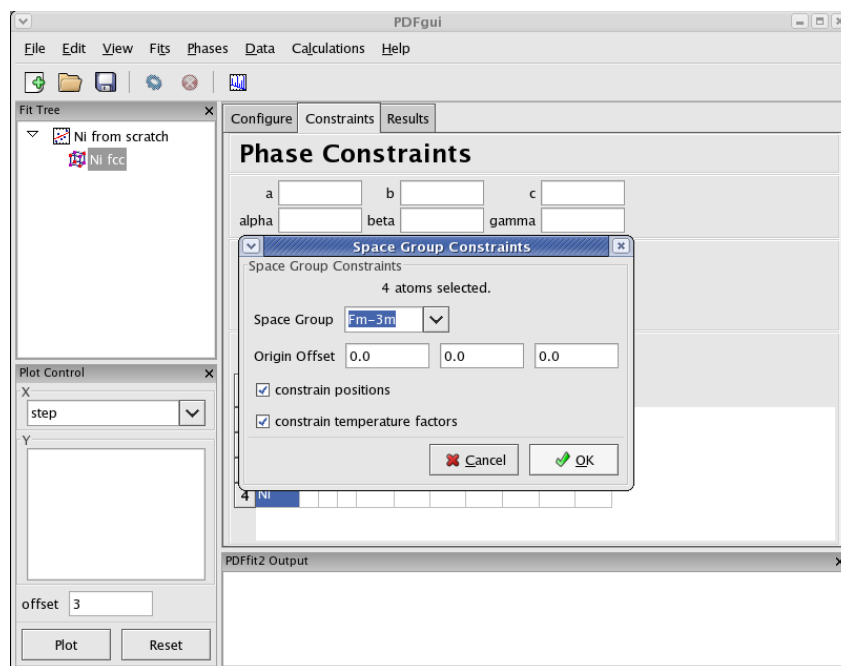


Figure 3.2: Setting up symmetry constraints to be used in a refinement.

In order to set the symmetry constraints for the refinement, select “Constraints” tab, highlight all the atoms, and right-click to invoke a menu. Note that hitting “Control” and “A” simultaneously will select all the atoms if the grid is active. (If the Fit Tree is active, it will select all tree items). Click on “Symmetry constraints...”, and a dialog box will appear asking for the space group to be used and whether you want the positions and/or the thermal parameters to be constrained, as shown in [Figure 3.2](#). The default is to use the space group used for expansion, and to constrain everything according to the symmetry. Since all the positions are special, these will not be refinable by symmetry, and the thermal parameters will be isotropic. The program will assign parameters according to symmetry using default names that can be renamed and/or reassigned by the user. To be consistent with the naming from the first example, one could assign the lattice parameters as @3 and the isotropic ADPs as @4. Note that parameters can be quickly renamed or consolidated in the “Parameters” panel of the Fit node using the right-click menu. It is important to note that the table reflecting constraints is *the only place* that program refers to for the symmetry. What is written there will be used, and if the table is tampered with, then the original symmetry obtained using symmetry expansion feature will *not* be preserved. Therefore, the expansion tool represents a convenience tool and nothing more than that. The remaining steps of this example are identical to these described in [Section 2.2 \[Simple fit\]](#), page 6.

3.2 Calculating PDF from a structure

There is often a need for obtaining a calculated PDF profile for a given structure instead of performing a fit. In order to carry out a calculation, an underlying structure needs to be specified. This can either be loaded from a file or generated from scratch. Suppose that we have a Ni structure populating a Fit Tree using steps described in either [Section 2.2 \[Simple fit\]](#), page 6 or in [Section 3.1 \[Symmetry expansion\]](#), page 17, and that we would like to calculate Ni PDF using neutron radiation. Highlight the Ni structure on Fit Tree. On menu bar, select **New Calculation** from “Calculations” drop-down menu. Current activity panel will display information very similar to that when a data set is loaded, as shown in [Figure 3.3](#).

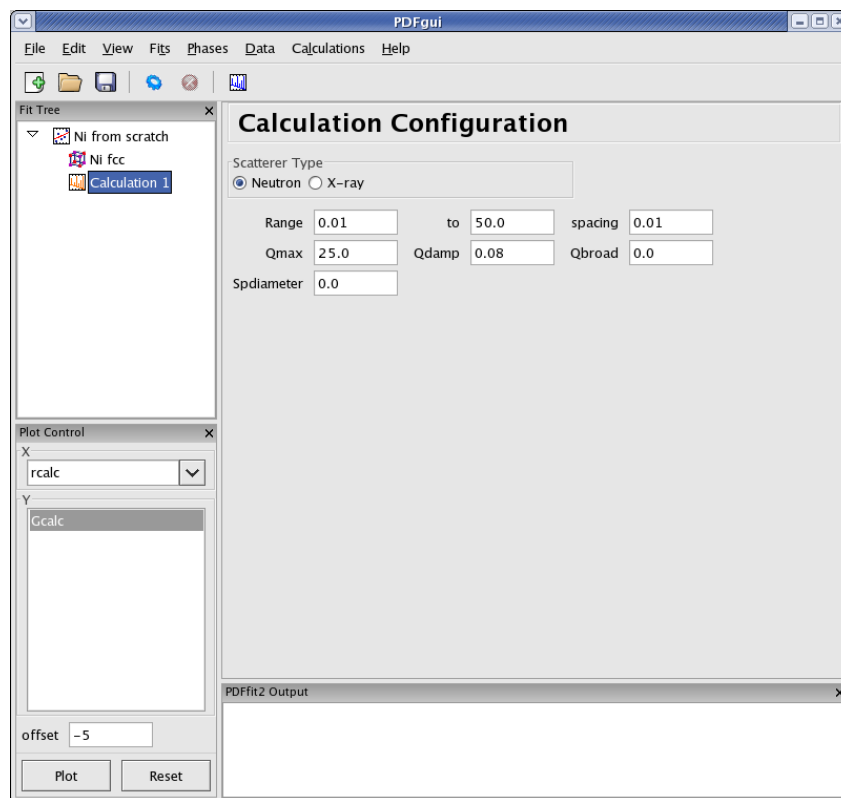


Figure 3.3: An example of the calculation configuration panel.

User has to specify conditions to be used for the calculation, such as radiation type, calculation range and corresponding r -grid size, as well as instrument resolution and maximum momentum transfer parameters. For the later two, the default values of parameters could be used, or values could be specified that closely mimic the experimental conditions on some particular instrument of interest. After the conditions are set, the gear icon on the tool bar can be used to execute the calculation (or alternatively select **Run Selected Calculation** from the “Calculations” menu). For our exercise, select neutrons for radiation. The defaults for the range and other parameters would work just fine. To get an experiment data lookalike calculation, select for example 0.01 for the r -grid size, and use 0.08 and 25.0

for resolution and maximum momentum transfer parameters respectively. Execute the calculation. Observe the result by clicking on the quick plot icon on the tool bar. To export the calculated PDF, use the **Export Selected Calculation** choice on the “Calculations” menu, after highlighting the performed calculation in question on the Fit Tree.

3.3 Sequential fitting

In many practical situations there is a need to set a series of refinements that are linked in a sequence. Whether motivated by following certain fitting protocol on a single data set, or such that numeric fluctuations of refined parameters are avoided in a temperature series data, or any other similar situation, sequential fitting capability is of clear importance and is supported in PDFgui. In what follows we will describe multi-stage fitting capabilities of PDFgui, followed by description of three pre-made macros that implement sequential fitting: fitting of r -series on a single data set, temperature series on a set of data corresponding to various temperatures, and a doping series on a set of data corresponding to samples whose chemical content changes in some systematic way. These macros can be accessed from the menu bar, “Fits” menu, **Macros** selection on the drop-down menu.

3.3.1 Multistage fitting

Here we describe how to string together several fits. To begin, create a fit as explained earlier. The next step is to copy the fit. This is achieved by right-clicking on the existing fit name “Fit 1” in the the Fit Tree, and selecting “Copy” option from the pop-up menu. Once this is done, a paste fit operation is done by right-clicking at the empty space on the Fit Tree, and selecting “Paste Fit” option from the pop-up menu. This will create “Fit 1_copy”, a copy of “Fit 1” in the Fit Tree. The next step is to link the fits. Select “Fit 1_copy” on the Fit Tree. In the “Parameters” panel, select the entire “Initial” column. Type “=Fit 1” and then press “Enter”. The “Initial” values of the parameters should now read “=Fit 1:n”, where n is the index of the parameter. This brings us to the linking syntax, which is “=name:index”. Here, “name” is the name of another fit to which the link is made, and “index” is the index of a parameter in that fit. If “index” is omitted, it will default to the index of the parameter one is linking from. A linked parameter uses the refined value of the link as its *initial* value. This is useful when one runs several related fits. An example of this is shown in [Figure 3.4](#).

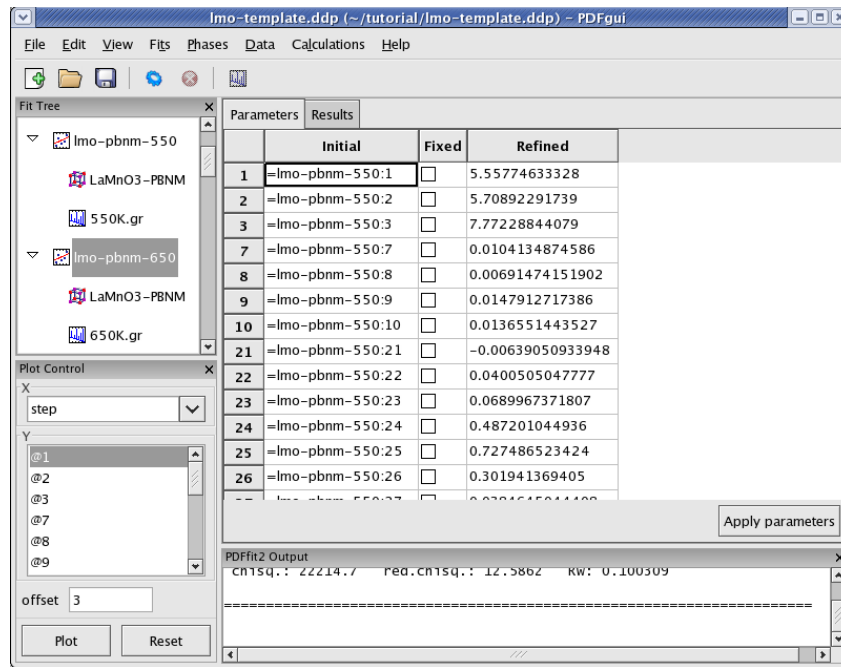


Figure 3.4: An example of linked fits. The output of a converged fit will be fed into the succeeding fit in the sequence as an input.

Additional fit parameters can be introduced when needed to the new fit. If our Ni example was used and copied in this exercise, one can select *Ni.stru* phase of “Fit 1_copy” and introduce parameter *delta2* by inserting “@5” in the appropriate box of the “Constraints” tab of that phase. This is a quadratic atomic correlation factor, a parameter related to the correlated motion of atoms, and as such should help in sharpening up the nearest neighbor PDF peak in the model PDF profile. Highlight the fits on the Fit Tree, and run them in the same way as described earlier. Only the highlighted fits will be executed. The fitting will proceed in stages, so the first fit is executed first, and, after it is converged, the second one. After the fitting of the sequence is over, you may plot the results as described earlier. This time hold down “Control” and select the data sets from both “Fit 1” and “Fit 1_copy”. You can change the offset in the plotting window to 0 if you would like to display the profiles on top of each other. Notable is a sharper nearest neighbor PDF peak in the second fit, improving the fit to the Ni data compared to the “Fit 1” achievement.

We note here that there is also a linear atomic correlation factor *delta1*. This one is appropriate to use in cases of high temperature, while *delta2* is more appropriate for the case of low temperatures. Alternative way to include the correlated motion effects on PDF is to introduce *sratio* parameter that defines low-*r* to high-*r* PDF peak ratio, and *rcut* limit needs to be specified that defines a cutoff distance. The two approaches of accounting for correlated motion should *not* be used simultaneously. Details of this and other issues related to the PDF method can be found elsewhere.

3.3.2 Sequential fitting of incremental r -series

In certain modeling situations user could benefit from fitting a data set through a series of refinements that differ one from another by the corresponding fitting ranges. This could either be a case when upper r -limit is to be changed incrementally, or when a fixed width r -window is to be defined for a box-car fitting approach. An example of this would be a situation when details of crossover from local to average structure are examined. PDFgui has a pre-written macro that allows for this modeling situation to be accommodated. We will illustrate these capabilities through a simple Ni example. Generate a complete fit, including a phase and a dataset, as explained in [Section 2.2 \[Simple fit\]](#), [page 6](#). Select the fit from the Fit Tree. From the “Fits” menu, select **Macros**, and choose “ r -Series” option. The Current Action panel will display simple setting requirements, arranged in two rows of three dialog boxes each, that need to be filled with values to be specified by the user. An example of this is shown in [Figure 3.5](#).

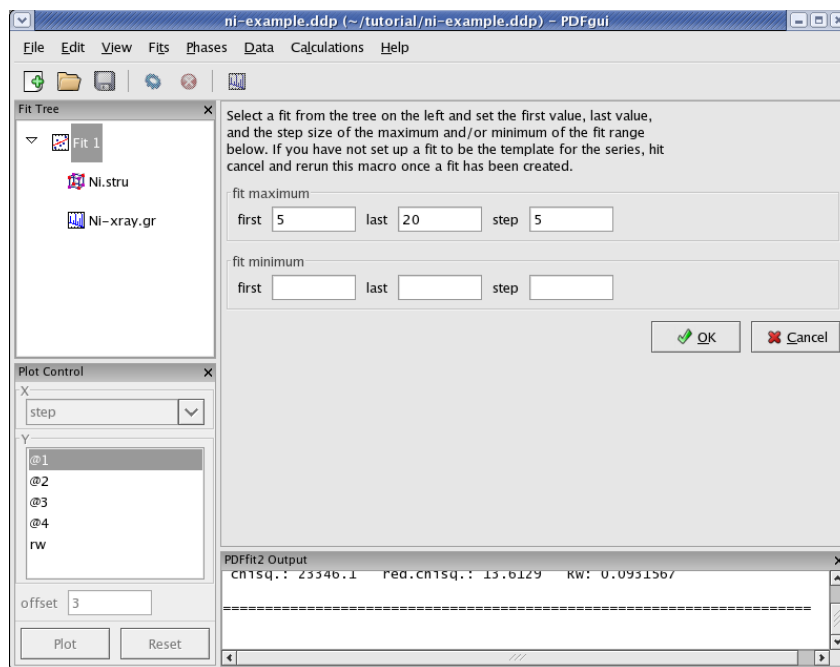


Figure 3.5: Appearance of the setup panel for specifying an incremental r -series fit conditions.

The first row deals with the increment setup of the upper r of the refinements. User should specify the first and the last fit maximum r -value, and corresponding step (increment), all in units of Angstroms. In the second row information is needed to specify the lower r refinement limit. Again, the user sets the first and the last value, and step. This allows for a box car of either fixed or variable width to be defined. If the second row is left blank, the corresponding refinement series will be with incremental maximum r only, and fixed lower limit. For the purpose of this exercise let's perform incremental fitting of Ni-data with 4 steps total, and fixed lower limit. To achieve this, type 5 for the first upper limit, 20 for the last upper limit, and the step of 5 Angstroms. Leave the second row blank.

Once this is done, make sure that the Ni fit, which in this situation serves as a template, is highlighted, and then click “OK” in the Current Action panel. This will generate 4 new fits below the original fit, and these four fits will constitute your series, with desired incremental limits. You select all four of them and execute the refinement. Once the sequence is done, you can review the results using Plot Control.

3.3.3 Sequential fitting of temperature series

More frequent modeling scenario is handling a large number of data sets originating from a single sample collected at various temperatures. One of the common modeling schemes in such cases is to perform sequential fitting of such data series, which is known to yield well behaved modeling parameters. PDFgui has a pre-written macro that allows for this modeling situation to be accommodated.

We will describe the procedure using an example of a temperature series of data collected on LaMnO₃ at various temperatures from 300 K to 1150 K at NPDF instrument at LANSCE at Los Alamos National Laboratory. This material exhibits Jahn-Teller (JT) order-disorder phase transition just above 700 K, where the long range orbital order is lost at high temperature, but the local JT distortion survives. The formal space group does not change at this transition. To begin, from the “File” menu use **Open Project** selection to open *lmo-template.dpp* project from the tutorial directory. This project consists of a fit called *lmo-pbnm*, which will serve as a template and which contains *Pbnm* phase of LaMnO₃ and a 300 K data set. The fit refinement is set up to cover 1.7-19.5 Angstroms range, and all the parameter values are set to their converged values for this temperature. The fit setup uses isotropic ADPs for all atomic sites. In the same directory data corresponding to various temperatures exist, in particular *300K.gr*, *550K.gr*, *650K.gr*, *700K.gr*, *720K.gr*, *730K.gr*, *740K.gr*, *750K.gr*, *800K.gr*, *880K.gr*, *980K.gr*, *1050K.gr*, *1100K.gr*, and *1150K.gr*. We will establish a T-series fit sequence. Select **Macros** from the “Fit” menu on the menu bar, and choose “Temperature Series”. The Current Action panel will reflect the selection. A tool is provided that allows for data sets to be added. Click on “Add” button. Using “Shift” and mouse-select operation you should select all the data sets mentioned in the above list, except the 300 K one, as this is already in the template fit, and then press “Open” button. All the data sets will be loaded. Since the files contain meta data, as mentioned earlier, the gui is going to pick up temperature information from the files. User should verify that the information is correct. The data can be ordered by temperature by clicking the “Temperature” header.

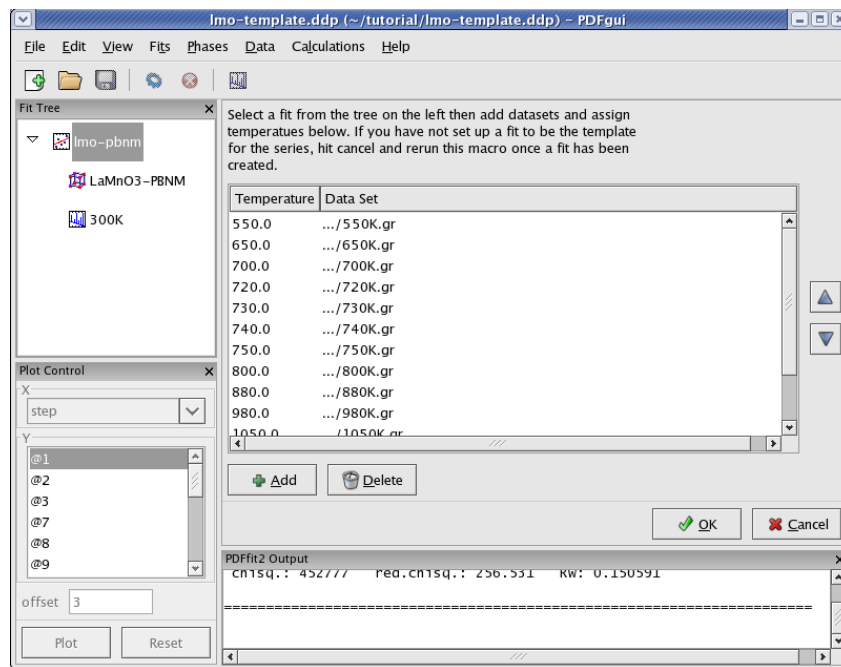


Figure 3.6: Setting up a T-series sequential refinement for LaMnO₃. Ordering by temperature will ensure that the fits are linked correctly.

Order the data by temperature (Figure 3.6), as otherwise the series of fits that is to be automatically generated in the next step will be linked in an arbitrary way in which the data were loaded, rather than to reflect the scientific logic. There should be 13 data sets. Next, ensure that the template fit is selected on the Fit Tree. If this is the case, the “OK” button becomes clickable. Clicking on the “OK” button will generate a sequence of linked fits in the Fit Tree in the order of the temperature increase. These fits are linked, which can be verified by inspecting the initial values on the “Parameters” tab of each fit. Highlight all the fits in the Fit Tree to start the sequential refinement of the T-series. The results can then be displayed in Fit Control such that various converged fit parameters are plotted versus temperature. In terms of the science behind the LaMnO₃ example, it is interesting to plot isotropic ADP of the oxygen at general position. Select all the phases on the Fit Tree by selecting one and then clicking on “Control”, “Shift” and “A” keys simultaneously. This will select all the phases at once. On the Plot Control choose the temperature for x axis, and select the Uiso of O2 (for example atom 20). Clicking on “Plot” button a plot will be displayed featuring Uiso vs T.

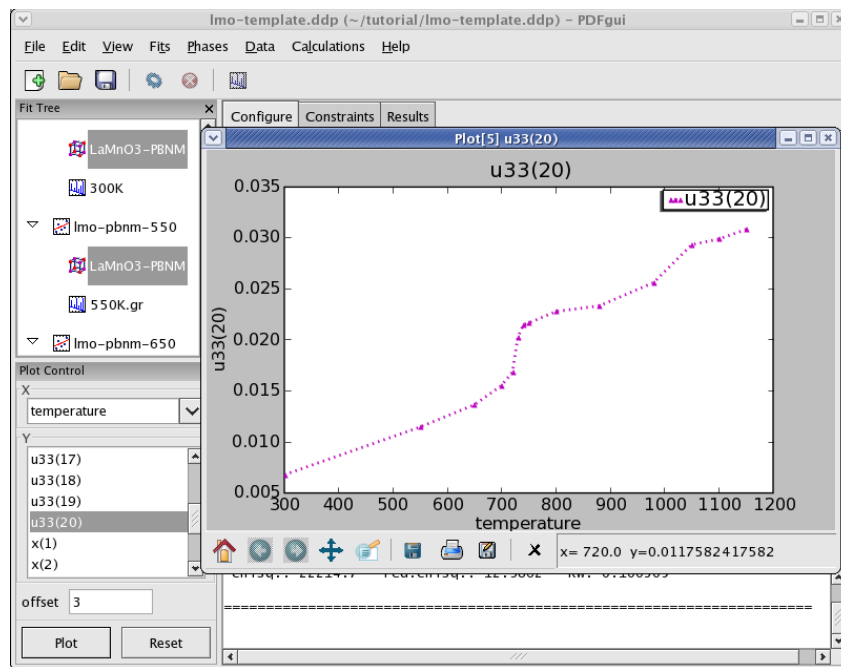


Figure 3.7: Displaying the refinement results as a function of external parameter: T-series refinement of LaMnO₃, example of isotropic ADP of oxygen atom on general position in *Pbnm* setting. Notable are the offsets just above 700 K (Jahn-Teller transition), and at around 1000 K when sample converts from orthorhombic to rhombohedral symmetry.

Despite quite high temperatures, an onset of the static offset above the transition temperature is clearly marked by this parameter, as apparent in Figure 3.7. Curious user could repeat the same T-series refinement restricting the refinement range upper limit to say 5-6 Angstroms and observe the outcome. The refinement sequence execution should be very quick in this case.

3.3.4 Sequential fitting of doping series

Fitting a series of PDF data that correspond to a set of samples with related chemistry, such as various doping series, represents another important sequential modeling aspect that is supported in PDFgui. PDFgui has a pre-written macro that allows for this modeling situation to be accommodated.

We will describe the procedure using an example of a series of data collected on LaMnO₃ and series of Ca-doped LaMnO₃ samples with various Ca content from 0 to 0.28 at GEM instrument at ISIS, UK. To begin, from the “File” menu use **Open Project** selection to open *lcmo-template.dpp* project from the tutorial directory, which has an initial setup for this exercise. Note the letter *c* in the file name, in contrast to the name used in the previous example. This project consists of a fit called *lcmo-pbnm*, which will serve as a template and which contains *Pbnm* phase of LaMnO₃ and a 10 K data set, *x000t010q35.gr*. The difference here with respect to the template used in the previous example is that Ca sites are introduced in the structure, but are assigned zero occupancy. However, existence of the

Ca dopant species in the structure of the template is essential for the macro to operate. Also, upper limit used in Fourier transform for obtaining this set of data is 35 inverse Angstroms, in contrast to 32 inverse Angstroms in previous example. Since different instrument was used, the value of Q_{damp} is different than in previous example.

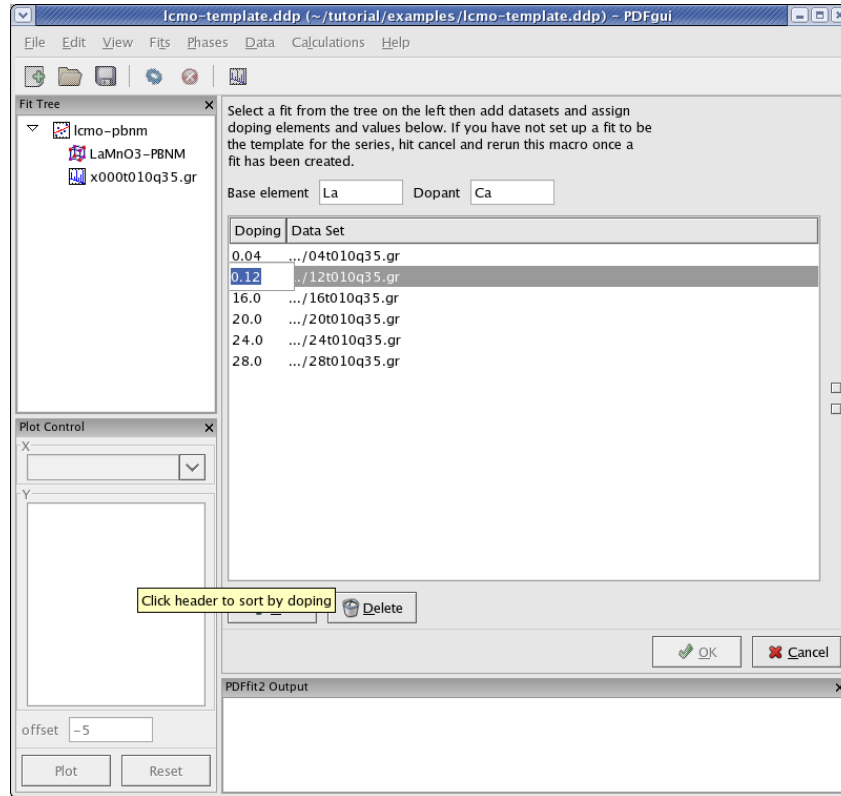


Figure 3.8: After loading of the Ca-doping data series of LaMnO₃ system, proper doping assignment needs to be carried out, as the doping levels introspected from the file names in this example incorrectly reflect the scientific situation. Note that dopant atom has to be present in the template seed used to generate the linked sequence of fits.

In the same directory data corresponding to various Ca contents exist, collected at 10 K temperature, in particular *x004t010q35.gr*, *x012t010q35.gr*, *x016t010q35.gr*, *x020t010q35.gr*, *x024t010q35.gr*, and *x028t010q35.gr*. It should be noted at this point that the data files do not contain any relevant meta data in the file headers. However, the doping level is somehow encrypted into the file names. We will establish a doping series fit sequence. Select **Macros** from the “Fit” menu on the menu bar, and choose “Doping Series”. The Current Action panel will reflect the selection. Base element and dopant need to be specified. A tool is provided that allows for data sets to be added. Click on “Add” button. Using “Shift” and mouse-select operation you should select all the data sets mentioned in the above list, and then press “Open” button. All the data sets will be loaded. The gui will introspect both the file names and files themselves in attempt to obtain the doping (or in previous example temperature) information. Since the files do not contain meta data, as mentioned

earlier, the gui is going to pick up doping information from the file names. User should verify that the information is correct. In this particular case the doping information will not be correctly picked up, as for example 004 from the name is meant to be 0.04 doping, and the gui would try to interpret it as 4.0 doping. Similarly 028 would be interpreted as 28.0, while it was intended to mean 0.28 doping. These values should be edited and fixed manually by clicking on the corresponding values and simply typing in the correct values (Figure Figure 3.8). The data can be ordered by doping by clicking the header. After you are done with editing, order the data by doping, as otherwise the series of fits that is to be automatically generated in the next step will be linked in an arbitrary way in which the data were loaded, rather than to reflect the scientific logic. There should be 6 data sets (with the initial $x=0.0$ data set there will be 7 chained fits total after this setup is done) . Next, ensure that the template fit is selected on the Fit Tree. If this is the case, the “OK” button becomes clickable. Clicking on the “OK” button will generate a sequence of linked fits in the Fit Tree in the order of the Ca content increase. These fits are linked, which can be verified by inspecting the initial values on the “Parameters” tab of each fit. Highlight all the fits in the Fit Tree to start the sequential refinement of the doping series.

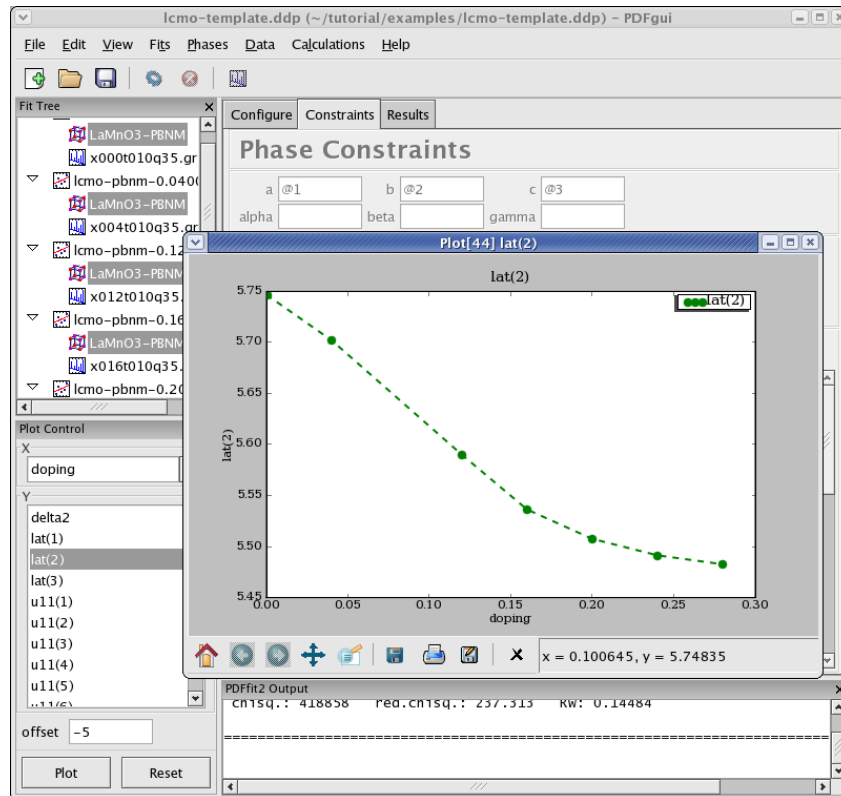


Figure 3.9: Sequence of refined parameters, such as lattice constants, can be plotted vs doping using PDFgui plotting facilities. Figure features lattice parameter b in $Pbnm$ space group setting for series of Ca-doped LaMnO₃ samples for doping concentrations between 0 and 0.28 at 10 K temperature.

After the convergence is achieved for all the fits in the Fit Tree, the results can be displayed in Fit Control such that various converged fit parameters are plotted versus Ca content. An example is provided in Figure [Figure 3.9](#) featuring one of the lattice parameters.

3.4 Nanoparticle structure

Determining the structure of a nanoparticle is notoriously difficult. Diffraction experiments on nanoparticle samples yield broad diffraction patterns that are hard to analyze using conventional crystallographic approach. One of the options that remain is to approach the problem through the PDF analysis. The PDF of a nanoparticle features sharp peaks conveying structural information. The PDF signal gets damped at higher distances due to the convolution caused by the finite nanoparticle size. For certain simpler cases when nanoparticles can be assumed to have spherical shape, characteristic parameters such as nanoparticle diameter can be obtained.

One of the features of this PDF modeling suite is a capability of modeling the effect of the finite nanoparticle size using a spherical form factor. Relevant PDF parameter is *Spdiameter* which is the diameter of the nanoparticle. The PDF analysis is rather difficult, as this parameter is highly correlated with various other parameters one would like to refine, such as anisotropic ADPs, scale factors, correlated motion parameters and so on. The refinement procedure is therefore rather delicate and the solutions are not as robust as we are used to in cases of crystalline materials.

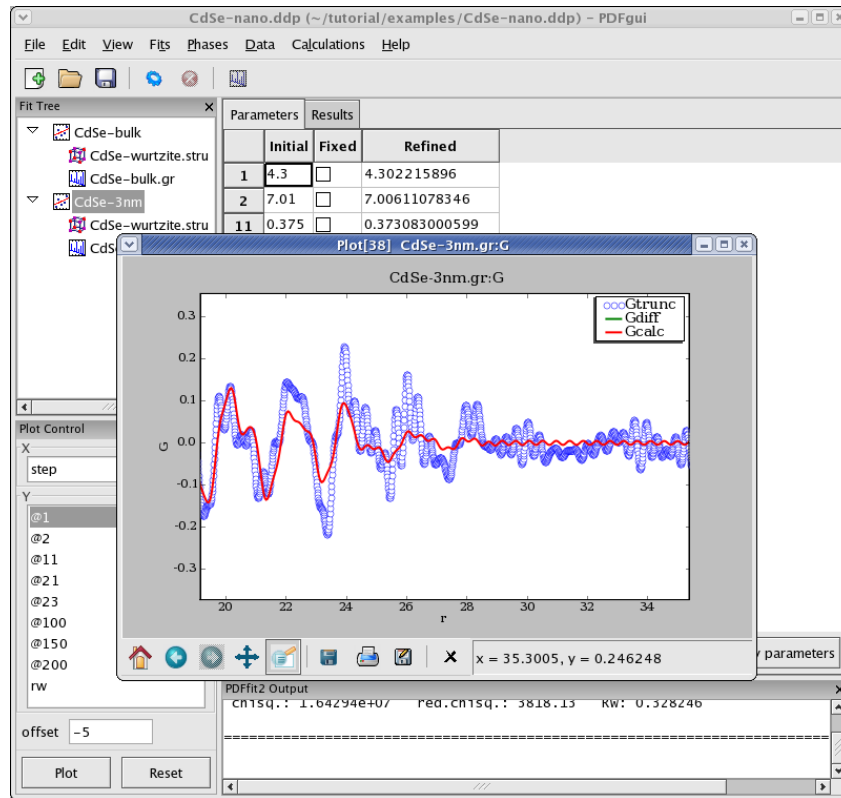


Figure 3.10: Fitting the structure of a nanoparticle: 3nm CdSe nanoparticle example.

To illustrate the program capabilities we present a case of CdSe nanoparticle approximately 3nm in size. It is useful to have PDF data for a crystalline reference, where available/applicable. In this exercise we start from a prepared project file *CdSe-nano.ddp*. This project contains two fits: the first one is a bulk CdSe reference, and the other pertains to the CdSe nanoparticle. For consistency the PDFs of both bulk and nano samples were obtained using Q_{max} of 14 inverse Angstroms, although the bulk material PDF could have been processed using a higher value. Synchrotron x-ray radiation was used to obtain the data at 300 K, based on an experiment carried out at 6-ID-D at the Advanced Photon Source at Argonne National Laboratory. The structure used for both data sets is wurtzite, space group P6₃mc. From calibrations on Ni standard Q_{damp} value of 0.0486 was obtained and is used here. We first carry out a refinement on the bulk reference. This is carried out over a range from 1.7 to 19.8 Angstroms, using 7 parameters: lattice parameters a and c (@1 and @2 respectively), selenium z fractional coordinate (@11), isotropic ADPs for Cd and Se (@21 and @23 respectively), the data scale factor (@100), and finally correlated motion related quadratic term coefficient $\delta 2$ (@200). The converged fit results in parameter values that can be further used for reference when modeling the nanoparticle data. We note that while the fit is reasonable, the values of the isotropic ADPs are enlarged. The fit can be further improved if anisotropic ADPs are introduced, although the z -direction related components

will remain enlarged due to the stacking disorder present in the structure. The referent value of 5.69 for *delta2* will be used as a starting value for the nanoparticle fit.

In the nanoparticle refinement we will use the same starting values for all the parameters, except for *delta2* and the nanoparticle diameter. The former is set to 5.69, and the later to 25 Angstroms. In other cases an approximate value of the spherical nanoparticle size is usually known, and it is the best to start from a reasonably good guess. Refining the nanoparticle data reveals nanoparticle diameter of approximately 30 Angstroms, as further illustrated in [Figure 3.10](#). Enlarged values of isotropic ADPs are again observed, and the fit is reasonably good. Further improvements can be obtained by introducing anisotropic ADPs, where again values related to the z-direction will remain abnormally large most probably due to the stacking related disorder. The fitting protocol is not very robust, as mentioned earlier. Enthusiastic user is encouraged to play around with the refinement conditions and witness the difficulties. Going into the greater details of the structural properties of nanosized CdSe is beyond the scope of this tutorial. The successful fitting scenario depends on particular details of a structural problem one is determined to solve. The problem of determining the structure of a nanoparticle remains difficult. The PDFgui suite is not intended to necessarily provide *the* solution, it is rather a helpful tool in the process of determining new details and exploring the space of possible solution candidates, yielding success in some instances.

4 Extras

4.1 PDF plotting

Plotting capabilities of PDFgui are provided through the Plot Control panel and the quick-plot icon on the tool bar. The Plot Control allows for selection of x and y coordinates for plotting. The actual quantities that could be assigned to the coordinates is determined by selection of either Fit, or Phase, or Data on the Fit Tree. The choices for x and y coordinates varies depending on what is selected on the Fit Tree. Special options like index, temperature and doping are available as choices for x in cases of plotting multiple fit results from sequential fitting protocols. The plot window provides essential functionality such as zoom, pan, cursor coordinate tracking, and shifts. Features such as saving, exporting and printing are also available. The principal intent of the plotting functionality is to allow quick access to the fitting results to enhance the scientific process. If data is selected on the Fit Tree, the user can plot various aspects of the PDF function, such as data, model and difference PDF profiles as a function of inter-atomic distance r . If the fit or the phase are selected, then various parameters, both structural and internal can be plotted instead. Since several formal plotting examples were given through the tutorial exercise, and having simplicity of usage in mind, no other plotting examples are provided, hoping that the usage is sufficiently simple for users to master individually with ease. An example plot of R_w vs refinement step is shown in Figure 4.1 for Ni example.

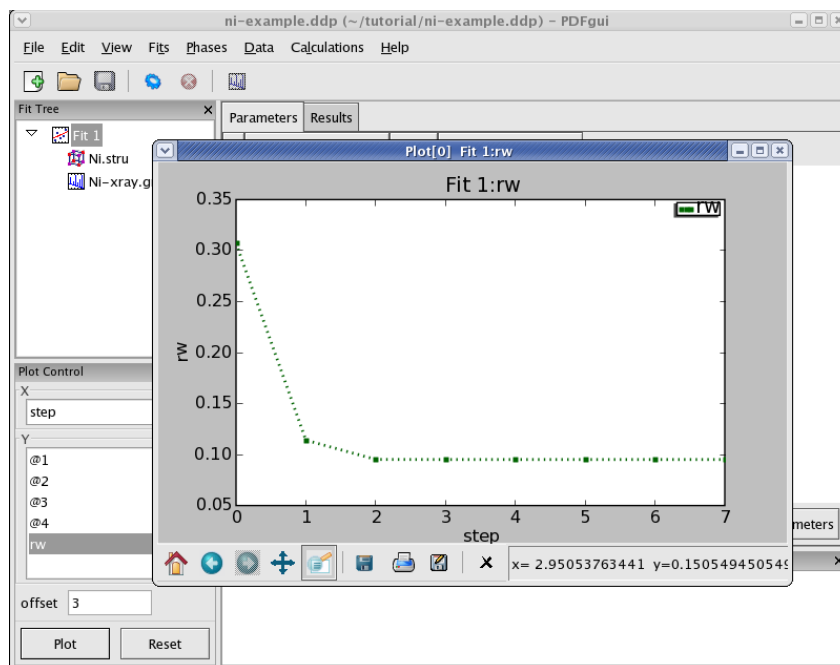


Figure 4.1: Plotting window featuring R_w vs refinement step for Ni example. The basic functionality for manipulating the plot is provided through icons on the tool bar of the plotting window.

4.2 Displaying the structure

For those users having AtomEye structural visualizer available in their system configuration, PDFgui allows for initial or refined structures to be visualized by passing required structural information to AtomEye that is invoked and corresponding 3D structure displayed on screen. This is achieved by highlighting a desired phase on the Fit Tree, and then selecting **Plot Initial Structure** or **Plot Refined Structure** from the “Phases” drop-down menu. The quickplot button will also invoke AtomEye with the refined structure, or initial structure if the refined structure does not yet exist. Corresponding structure visualization is displayed, which can be adjusted, rotated and zoomed in and out through the mouse functionality. The size of the displayed atoms can be controlled via “Page Up” and “Page Down” keys. For advanced AtomEye usage please consult corresponding AtomEye user documentation. An example Ni structure visualization is shown in [Figure 4.2](#).

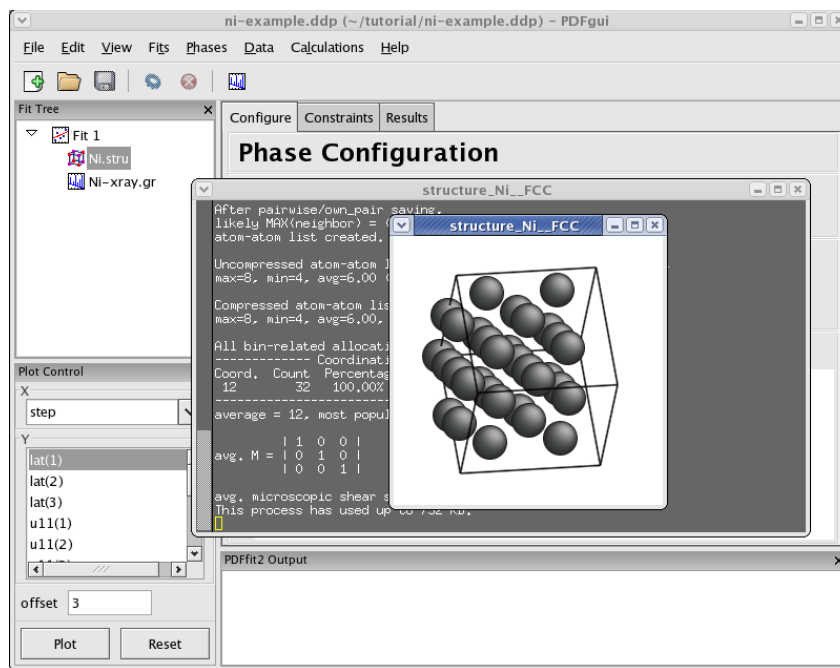


Figure 4.2: Using AtomEye functionality (if installed on your system) for 3D visualization of the initial and refined PDF structures: example of Ni structure.

4.3 Advanced usage and special needs

The PDFgui is designed to accommodate most common modeling situations. However, it is not encapsulating all the capabilities available within the modeling engine, such as calculation of differential PDFs, handling atoms with special scattering properties, etc. Advanced usage of PDFfit2 engine to resolve any such special modeling need that user may have is available through usage of Python scripts in the expert command line mode, similar to that featured in the PDFFIT program, where various special situations could be accounted for. Handling these situations requires detailed knowledge of the PDFfit2

syntax based on Python, which is beyond the scope of this user guide and will be described elsewhere.

5 PDFgui reference sheets

5.1 PDFgui hot keys

Fit Tree

Ctrl+A Select all items in the Fit Tree.

Ctrl+Shift+A Once an object is selected on the Fit Tree, using this hot key will select all the objects on the tree that are of the same type. For example, if a single data set is selected on the tree, hitting this hot key will make all the data sets belonging to all the fits in the tree to become selected. This is particularly useful for simultaneously plotting various fit results across the Fit Tree.

Ctrl+C Copy selected item.

Ctrl+X Delete selected item. Note that this does not place a copy of the item in the clipboard.

Delete Same as **Ctrl+X**

Ctrl+V Paste previously copied item. Note that this will only paste the item to a legal position in the tree. For example, a phase node cannot be pasted into an empty tree.

Phase configuration grid

Ctrl+A Select all items.

Delete Delete row. This will delete any wholly selected row.

Ctrl++ Add an atom to the grid.

Ctrl+- Same as **Delete**.

Phase constraints grid

Ctrl+A Select all items.

Delete Delete contents of selected cells.

5.2 List of PDFfit2 variables

The following is the list of PDFfit2 variables, including their default values in parentheses, and a brief description and a note, where appropriate. Note that some of the variables used in PDFFIT are renamed in PDFfit2. The reference of these changes is provided for the convenience and orientation of those users that are used to the old naming scheme.

New Variables

spdiameter [Dataset]
(float Å, default 0 Å)
spdiameter is a particle diameter for PDF shape damping function. Shape damping is not applied when *spdiameter* equals zero.

anisotropy(n) [Phase]
anisotropy(n) (bool, inferred from *uij(n)* values)
Flag for anisotropic thermal displacements of atom n. Setting of *anisotropy(n)* updates the *uij(n)* or *uisoequiv(n)* values.

uisoequiv(n) [Phase]
 (float Å², calculated from *uij(n)* values)
 Isotropic thermal displacement of atom n or equivalent displacement for anisotropic atom. For anisotropic sites the setting of *uisoequiv(n)* scales values of the *uij(n)* elements. *uisoequiv(n)* can be constrained only for isotropic atoms.

Renamed Variables

dscale [Dataset]
 (float unitless, *dscal[s]* in PDFFIT)
 Scale factor for the selected dataset.

qdamp [Dataset]
 (float Å⁻¹, default 0 Å⁻¹, *qsig* in PDFFIT)
 PDF Gaussian dampening envelope due to limited Q-resolution. Not applied when equal zero. The Gaussian envelope is of the form

$$B(r) = e^{-\frac{(rQ_{damp})^2}{2}}$$

qbroad [Dataset]
 (float Å⁻¹, default 0 Å⁻¹, *qalp* in PDFFIT)
 PDF peak broadening from increased intensity noise at high Q. Not applied when equal zero. See definition of [PDF Peak Width], page 37 for detailed explanation.

uij(n) [Phase]
 (float Å², ij=(11, 22, 33, 12, 13, 23), *u[i,n]* in PDFFIT)
 Elements of anisotropic displacement tensor of atom n.

occ(n) [Phase]
 (float unitless, *o[n]* in PDFFIT)
 Occupancy of site n.

pscale [Phase]
 (float unitless, *cscal[p]* in PDFFIT)
 Scale factor of the current phase.

delta1 [Phase]
 (float Å, default 0 Å, *gamm* in PDFFIT)
 Coefficient for (1/r) contribution to the peak sharpening. See definition of [PDF Peak Width], page 37 for detailed explanation.

delta2 [Phase]
 (float Å², default 0 Å², *delt* in PDFFIT)
 Coefficient for 1/r² contribution to the peak sharpening. See definition of [PDF Peak Width], page 37 for detailed explanation.

sratio [Phase]
 (float unitless, default 1, *srat* in PDFFIT)
 Sigma ratio for bonded atoms. Reduction factor for PDF peak width accounting for correlated motion of bonded atoms.

Preserved Variables

qmax [Dataset]

(float Å⁻¹, default 0 Å⁻¹)

qmax cutoff is an experimental parameter having fixed value determined during the PDF data processing in the Fourier transform step. Finite data range used in the Fourier transform is a source of termination ripples. Termination ripples are *not* applied in PDF calculation when *qmax* is set to zero. The effect is calculated using the following equation

$$G'(r) = \frac{1}{\pi} \int_0^\infty G(r') \left(\frac{\sin Q_{max}(r - r')}{r - r'} - \frac{\sin Q_{max}(r + r')}{r + r'} \right) dr'$$

lat(i) [Phase]

(float Å or degrees, i=1,2,...,6)

Lattice parameters a, b, c, alpha, beta, gamma of the current phase. Can be also used as lat('a') etc.

x(n), y(n), z(n) [Phase]

(float unitless)

Fractional coordinates for atom n.

rcut [Phase]

(float Å, default 0 Å)

Radius cutoff for applying *sratio* sharpening factor.

PDF Peak Width

The PDF peak width contains contributions from thermal and zero point displacements as well as static disorder. For large distances *r* the motion of the two contributing atoms is uncorrelated, for small distances, however, the motion can be strongly correlated leading to a sharpening of the first peak(s) in the observed PDF. The program PDFfit2 provides three different correction terms for the PDF peak width. The final width is given by

$$\sigma_{ij} = \sigma'_{ij} \sqrt{1 - \frac{\delta_1}{r_{ij}} - \frac{\delta_2}{r_{ij}^2} + Q_{broad}^2 r_{ij}^2}$$

Here the primed sigma is the peak width without correlation, given by the structural model. The first two terms correct for the effects of correlated motion. Within the scope of the users guide, we just mention that the term *delta2/r²* describes the low temperature behavior, and term *delta1/r* describes the high temperature case. Since the two parameters are highly correlated, one will in practice choose which one to refine. The last term in the equation models the PDF peak broadening as a result of the Q resolution of the diffractometer. In many cases this term will only be significant for refinements over wider *r*-ranges. Note that the Q resolution also results in an exponential dampening of the PDF peaks which is modeled using the parameter *qdamp*.

Index

A

acknowledgements	2
advanced usage	33
availability	3

B

building structure from scratch	17
---------------------------------------	----

C

calculating PDF	20
Community	5
copyright	2

D

displaying the structure	33
doping series	26

E

examples	17
extras	32

F

fit	6, 12
-----------	-------

H

hot keys	35
----------------	----

I

installation	3
introduction	1

L

list of variables	35
-------------------------	----

M

mail-list	5
multistage fitting	21

N

nanoparticle diameter	29
nanoparticle size	8
nanoparticle structure	29

P

PDF plotting	32
PDFfit2	1
PDFfit2 variables	35
PDFgui	2
PDFgui hot keys	35
PDFgui layout	6
PDFgui reference sheets	35

Q

quick start	6
-------------------	---

R

r-series	23
reference sheets	35
refinement	6, 12

S

sequential fitting	21, 23, 24, 26
simple fit	6
special needs	33
symmetry expansion	17

T

temperature series	24
tutorials	17

U

user groups	5
-------------------	---

V

variables	35
-----------------	----