PDFmorph user guide

1.0.0 release
June 2024

C. L. Farrow, C. J. Wright, P. Juhás, C.-H. Liu, T. Davis, S. M. Román, A. Yang, and S. J. L. Billinge

Acknowledgments

The PDFmorph app was developed by the Billinge group at Columbia University. The examples in the tutorial part of the manual were made possible by data collected by Soham Banerjee at Brookhaven National Laboratory, Benjamin Frandsen during his PhD at Columbia Unversity, and Long Yang from Tonji University.

The DiffPy development team is the set of all contributors to DiffPy projects. Each member of the development team maintains copyright on their individual contributions to the code base. For a detailed contribution history of `PDFmorph`, see the git logs at `https://github.com/diffpy/diffpy.pdfmorph`.

For more information about the application, please visit `https://www.diffpy.org/diffpy.pdfmorph`. Report any bugs to diffpy-users@googlegroups.com or submit an issue to `https://github.com/diffpy/diffpy.pdfmorph/issues`.

Preface

Redistribution

Disclaimer

Table of Contents

1 Introduction

`PDFmorph` is a Python package that increases the insight researchers can obtain by comparing measured atomic pair distribution functions (PDFs) in a model-independent way. It was originally designed to help a researcher answer a question: "Has my material undergone a phase transition between these two measurements?"

A common approach to compare two PDFs is to plot the difference curve below the two PDFs. However, a significant signal can be seen in the difference curve due to benign effects such as thermal expansion (peak shifts), increased thermal motion (peak broadening), or a change in scale due to differences in the incident flux when computing the PDFs for example. Other commonly-used dissimilarity metrics such as the weighted Rietveld reliability factor R_W are also inflated by these effects. `PDFmorph` does its best to correct for these benign effects prior to computing the R_W and plotting the difference curve.

To use `PDFmorph`, supply one PDF as the "target" (suggested to the one collected under higher temperature) and a second PDF to be morphed. The user can choose from a list of morphs to apply, which includes "stretching" (scaling the r-axis, the abscissa, to simulate isotropic expansion/compression), "smearing" (peak broadening by convlution with a Gaussian), and "scaling" (scaling the ordinate axis). By default, `PDFmorph` will refine the intensity of the morphs to best match the morphed to the target PDF. All morphed PDFs generated by `PDFmorph` can easily be saved and exported (see the `--save` option in Chapter 5 [PDFmorph options], page 19).

`PDFmorph` includes other morphs including those that factor in nanoparticle shape effects. For a full list of available morphs, see Chapter 4 [Available morphs], page 15. For a quick tutorial on using `PDFmorph`, see Section 3.1 [Quick start tutorial], page 4. For a full list of command-line options, see Chapter 5 [PDFmorph options], page 19.

Finally, we note that though `PDFmorph` could be used on other spectra, it has not been extensively tested beyond the PDF.

2 Installation

2.1 Availability

PDFmorph is open source and distributed under the BSD license. It runs on Windows, Mac OS, Linux, and all major Unix systems. The source code is freely available at https://github.com/diffpy/diffpy.pdfmorph.

If you come accross any bugs in the application, please open an issue (https://github.com/diffpy/diffpy.pdfmorph/issues) or email diffpy-dev@googlegroups.com.

2.2 Requirements

PDFmorph is currently run from the command-line, so it is recommended that users first familiarize themselves with their operating system's command-line interface.

PDFmorph currently requires Python 3.10 or higher to run. It also makes use of the following third party libraries:

- NumPy (https://numpy.org) - Python's fast scientific computing library
- Matplotlib (https://matplotlib.org) - Python plotting library
- SciPy (https://scipy.org) - Python's technical computing library
- diffpy.utils (https://www.diffpy.org/diffpy.utils) - DiffPy (https://www.diffpy.org) general utility functions

These dependencies will be installed automatically if you use the conda installation procedure described in the following section (Section 2.3 [Installation instructions], page 2).

2.3 Installation instructions

We recommend installing the package and its dependencies using conda. Miniconda, a minimal installer for conda, can be downloaded here (https://docs.anaconda.com/free/miniconda). Once Miniconda is installed, run conda on the command-line to test that it has been installed properly.

To create and activate a conda environment to use PDFmorph, run the following command.

```
conda create -n pdfmorph_env python=3 --yes
conda activate pdfmorph_env
```

A conda environment must be active to use the packages within. In future sessions, make sure to run conda activate pdfmorph_env to activate the environment before interacting with the PDFmorph program.

Once active, you can install PDFmorph into your environment from the conda-forge (https://conda-forge.org) channel of Anaconda packages by running

```
conda config --add channels conda-forge
conda install diffpy.pdfmorph
```

This installation only needs to be done once. The next time you activate the environment, these packages will already be installed. To see a list of all installed packages within an environment, run

```
conda list
```

When you are finished with the session, exit the environment by running

```
conda deactivate pdfmorph_env
```

If you do not wish to use conda to install, `PDFmorph` is also available on the Python Package Index (`https://pypi.org`) or can be installed from source at `https://github.com/diffpy/diffpy.pdfmorph`. Note that these installation methods will require you to first install the dependencies listed in the previous section (Section 2.2 [Requirements], page 2).

3 Tutorials

This section includes a quick start tutorial to acquaint users with the core features of `PDFmorph`. Also included are a few extra tutorials covering some more specific `PDFmorph` functionalities.

Before you start this tutorial, make sure you have installed all necessary software and dependencies (Section 2.3 [Installation instructions], page 2). If you are looking for a full summary of `PDFmorph` command-line options, see Chapter 5 [PDFmorph options], page 19, or run `pdfmorph --help` in a `PDFmorph`-equipped `conda` environment.

3.1 Quick start tutorial

In this tutorial, we will demonstrate how to use `PDFmorph` to compare two PDFs measured from the same material at different temperatures. The morphs showcased include "stretch", "scale", and "smear". Throughout this tutorial, entries surrounded by angle brackets <> represent placeholder names. The user is free to use their own names in stead of those provided.

1. Open your terminal (Linux and MacOS) or Command Prompt (Windows).
2. If not already active, activate your `conda` environment with `PDFmorph` installed using

   ```
   conda activate <PDFmorph_env>
   ```

   - See Chapter 2 [Installation], page 2, for how to set up a `conda` environment with `PDFmorph` installed.
   - If you need to list your available `conda` environments, run

     ```
     conda env list
     ```
3. Create a directory to store the tutorial PDF files using

   ```
   mkdir <PDFmorph_tutorial_directory>
   ```

   and enter the directory with the `cd` command

   ```
   cd <PDFmorph_tutorial_directory>
   ```
4. Download the quick start tutorial files `tutorialData.zip` (`https://www.diffpy.org/diffpy.pdfmorph/tutorialfiles.html`) into your current directory and extract the files. After this is done, type `ls` in the command-line to list your current directory contents and make sure there is a directory named `tutorialData`.
   - The files in this dataset were collected by Soham Banerjee at Brookhaven National Laboratory in Upton, New York. Note that these files have a `.gr` extension, indicating that they are measured PDFs. The `.cgr` file extension indicates a calculated PDF such as those generated by the PDFgui (`https://www.diffpy.org/products/pdfgui.html`) program.
   - Each file is PDF data collected on Iridium Telluride with 20% Rhodium Doping $IrRhTe_2$ at different temperatures with the first file (01) collected at $10K$ and the last (44) at $300K$. The samples increase in temperature as their numbers increase. The "C" in their names indicates that they have undergone cooling.
5. Enter the `tutorialData` directory using the `cd` command and find the data labeled `darkSub_rh20_C_##.gr` within this directory.

```
(pdfmorph_env) ~ $cd PDFmorph_tutorial_directory/
(pdfmorph_env) PDFmorph_tutorial_directory $ls
tutorialData  tutorialData.zip
(pdfmorph_env) PDFmorph_tutorial_directory $cd tutorialData
(pdfmorph_env) tutorialData $ls
__MACOSX  tutorialData
(pdfmorph_env) tutorialData $cd tutorialData/
(pdfmorph_env) tutorialData $ls
darkSub_rh20_C_01.gr  darkSub_rh20_C_09.gr  darkSub_rh20_C_17.gr  darkSub_rh20_C_25.gr  darkSub_rh20_C_33.gr  darkSub_rh20_C_41.gr
darkSub_rh20_C_02.gr  darkSub_rh20_C_10.gr  darkSub_rh20_C_18.gr  darkSub_rh20_C_26.gr  darkSub_rh20_C_34.gr  darkSub_rh20_C_42.gr
darkSub_rh20_C_03.gr  darkSub_rh20_C_11.gr  darkSub_rh20_C_19.gr  darkSub_rh20_C_27.gr  darkSub_rh20_C_35.gr  darkSub_rh20_C_43.gr
darkSub_rh20_C_04.gr  darkSub_rh20_C_12.gr  darkSub_rh20_C_20.gr  darkSub_rh20_C_28.gr  darkSub_rh20_C_36.gr  darkSub_rh20_C_44.gr
darkSub_rh20_C_05.gr  darkSub_rh20_C_13.gr  darkSub_rh20_C_21.gr  darkSub_rh20_C_29.gr  darkSub_rh20_C_37.gr
darkSub_rh20_C_06.gr  darkSub_rh20_C_14.gr  darkSub_rh20_C_22.gr  darkSub_rh20_C_30.gr  darkSub_rh20_C_38.gr
darkSub_rh20_C_07.gr  darkSub_rh20_C_15.gr  darkSub_rh20_C_23.gr  darkSub_rh20_C_31.gr  darkSub_rh20_C_39.gr
darkSub_rh20_C_08.gr  darkSub_rh20_C_16.gr  darkSub_rh20_C_24.gr  darkSub_rh20_C_32.gr  darkSub_rh20_C_40.gr
(pdfmorph_env) tutorialData $
```

Figure 3.1: Entering the `tutorialData` directory and searching for the PDF data (Linux Terminal Output).

6. Let us try try using the `PDFmorph` program to plot two PDFs against each other. Type the following command into the command-line

   ```
   pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr
   ```

   Underneath the two PDFs, we can see a substantial difference curve and large R_W value of 0.407

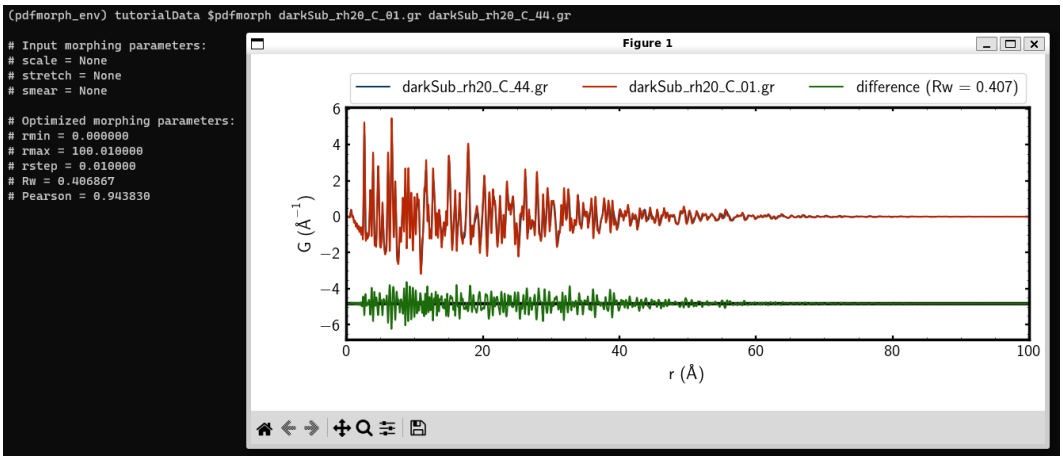

Figure 3.2: Using `PDFmorph` to compare two PDFs without morphing.

7. Now, we will start the morphing process.
   - Note that the morphs will apply only to the first PDF provided (which will be `darkSub_rh20_C_01.gr` in our case). We will henceforth refer to the first PDF as the "morphed" and the second (unmorphed) PDF as the "target". Our goal will be to apply morphs to see if we can reduce the difference between the morphed and target PDFs.
   - In this tutorial, we will apply the scale factor, Gaussian smear, and stretch morphs. To learn more about why we wish to apply these three morphs in particular, see Section 4.1 [Temperature-related morphs], page 15.
8. We will begin by applying a scaling factor of 2 by typing the command

```
pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=2 -a
```

Now, the difference is much larger $R_W = 1.457$. We should modify our initial value for the scaling factor until we see a reduction in R_W. We can try 0.9

```
pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.9 -a
```

and see that the difference has dropped to $R_W = 0.351$, lower than when comparing the unmorphed PDF. For `PDFmorph` to optimize the scale factor, drop the `-a` option

```
pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.9
```

Given a reasonable initial guess, `PDFmorph` will find the optimal value for each morph. In this example, `PDFmorph` should display `scale=0.799025` meaning it has found this to be the optimal value for the scale factor. The difference $R_W = 0.330$ has further decreased.

- The `-a` option prevents `PDFmorph` from refining the morph. It is the choice of the user whether or not to run values before removing `-a` when analyzing data with `PDFmorph`. By including it, you can attempt to manually move towards convergence before allowing the program to optimize by removing it; when including it, you may reach a highly optimized value on the first guess or diverge greatly.

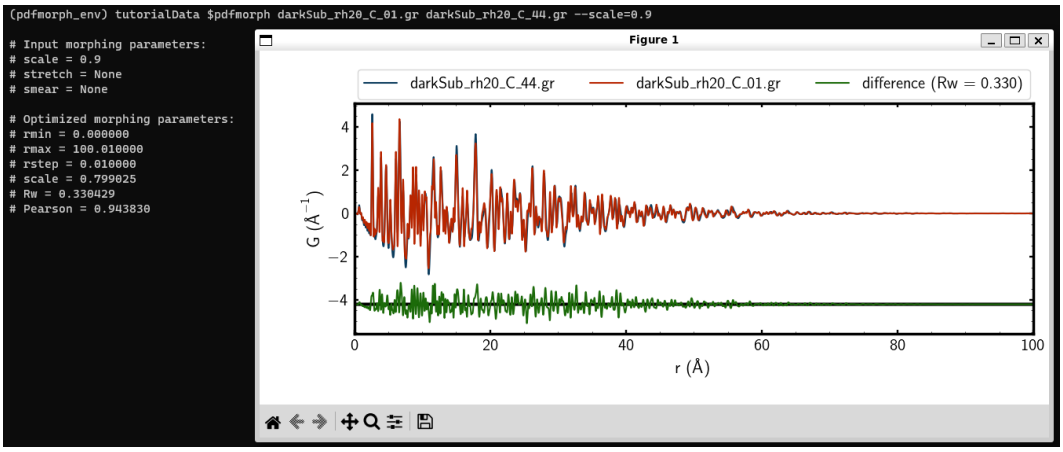

Figure 3.3: `PDFmorph` has found an optimal value (minimizes R_W) for the scale factor.

9. Now we will add on a Gaussian smear morph with an initial smear factor of 0.5. The absolute value of the smear factor is the standard deviation of the Gaussian that is convoluted with the morphed PDF.

```
pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.8 \
--smear=0.5 -a
```

- The \ in the above command is only used to break our long command into multiple lines for readability on this document. In your Terminal/Command Prompt, you can write the entire command in a single line.

We have tailored our initial scale factor to be close to the optimal value given by `PDFmorph`, but we should see that the $R_W = 0.897$ has increased substantially due to the smear factor. Remove the $-a$ from above and run it again to refine the smear factor.

- Note that the warnings that the Terminal/Command Prompt displays are largely numerical in nature and do not indicate a physically irrelevant guess. These are somewhat superficial and in most cases can be ignored.

After refining, we should see that R_W has dropped back to 0.330, and the optimized smear factor is quite small (`smear = 0.002940`) indicating that the smear has hardly had an effect on our PDF. To see an effect, restrict the `rmin` and `rmax` values by typing

```
pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.8 \
--smear=0.5 --rmin=1.5 --rmax=30
```

The optimized smear factor is now a non-insignificant `smear=-0.084138` and $R_W = 0.204$.

- We restricted the r (abscissa) values because some of the Gaussian smear effects are only visible in a fixed r range. We chose this r range by noting where most of our relevant data was that was not exponentially decayed by instrumental shortcomings.

- Also note that a negative smear factor is equivalent to its absolute value. You can see that

  ```
  pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.8 \
  --smear=-0.84138 --rmin=1.5 --rmax=30 -a
  ```

  and

  ```
  pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.8 \
  --smear=+0.84138 --rmin=1.5 --rmax=30 -a
  ```

  give the exact same results.

10. Finally, we will examine the stretch morph. The r axis will be stretched by $1+s$ where s is the stretch factor.

    ```
    pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.8 \
    --smear=-0.84 --stretch=0.5 --rmin=1.5 --rmax=30 -a
    ```

    Again, the R_W has increased. Before letting `PDFmorph` refine automatically, try seeing which direction (higher or lower) the initial stretch factor should go to decrease R_W. If you cannot, type

    ```
    pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.8 \
    --smear=-0.84 --stretch=0.005 --rmin=1.5 --rmax=30 -a
    ```

    to observe a decreased difference and remove the `-a` to get the optimized stretch factor of `stretch=0.001762` and smaller $R_W = 0.117$. We have reached the optimal fit for our PDF!

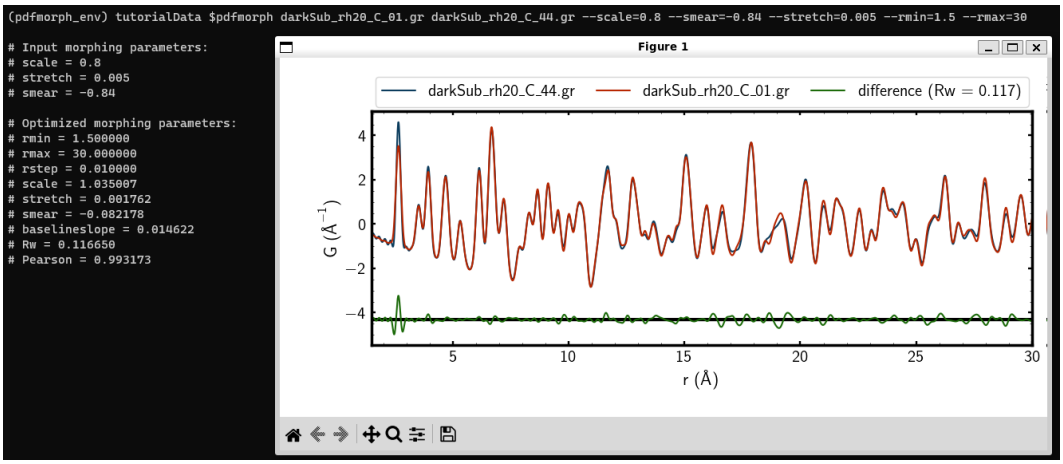

Figure 3.4: The optimal fit after applying the scale, smear, and stretch morphs.

11. To save your new morphed PDF as a file named `morph.gr`, use the save command

    ```
    pdfmorph darkSub_rh20_C_01.gr darkSub_rh20_C_44.gr --scale=0.8 \
    --smear=-0.84 --stretch=0.005 --rmin=1.5 --rmax=30 \
    --save=morph.gr
    ```

    Type `ls` to confirm a new file named `morph.gr` has been saved.

12. Now, try it on your own! If you have personally collected or otherwise readily available PDF data, try this process to see if you can morph your PDFs to one another. Note that many of the parameters provided in this tutorial are unique to it, so be cautious about your choices and made sure that they remain physically relevant.

3.2 Extra tutorials

PDFmorph has some more functionalities not showcased in the basic workflow above (run `pdfmorph ---help` for an overview of all `PDFmorph` functionalities). Tutorials for some of these are included below. Additional files are used for these tutorials. Please download and extract the extra tutorial data `additionalData.zip` (`https://www.diffpy.org/diffpy.pdfmorph/tutorialfiles.html`) before proceeding.

3.2.1 Performing multiple morphs

It may be useful to morph a PDF against multiple targets: for example, you may want to morph a PDF against a sequence of PDFs measured at various temepratures to determine whether a phase change has occured. PDFmorph currently allows users to morph a PDF against all files in a selected directory and plot resulting R_W values from each morph. It is advised that the lowest temperature PDF be that morphed and the higher temperature PDFs act as targets as the smear morph is only able to account for increases in thermal motion.

1. Within `additionalData`, enter (using `cd`) the `morphMultiple` directory. Inside, you will find multiple PDFs of $SrFe_2As_2$ measured at various temperatures.

- These PDFs come from *Atomic Pair Distribution Function Analysis: A Primer* (`https://github.com/Billingegroup/pdfttp_data`) by Simon Billinge and Kirsten Jensen. Contributions to the dataset have been made by Benjamin Frandsen and Long Yang.

```
(pdfmorph_env) PDFmorph_tutorial_directory $cd additionalData
(pdfmorph_env) additionalData $ls
additionalData
(pdfmorph_env) additionalData $cd additionalData/
(pdfmorph_env) additionalData $ls
morphMultiple  morphShape
(pdfmorph_env) additionalData $cd morphMultiple
(pdfmorph_env) morphMultiple $ls
SrFe2As2_150K.gr  SrFe2As2_168K.gr  SrFe2As2_186K.gr  SrFe2As2_204K.gr  SrFe2As2_222K.gr  SrFe2As2_240K.gr
SrFe2As2_156K.gr  SrFe2As2_174K.gr  SrFe2As2_192K.gr  SrFe2As2_210K.gr  SrFe2As2_228K.gr  SrFe2As2_246K.gr
SrFe2As2_162K.gr  SrFe2As2_180K.gr  SrFe2As2_198K.gr  SrFe2As2_216K.gr  SrFe2As2_234K.gr
(pdfmorph_env) morphMultiple $
```

Figure 3.5: The files within the `morphMultiple` directory.

2. Let us start by getting the R_W of `SrFe2As2_150K.gr` compared to the other, higher-temperature PDFs in the directory. Run

   ```
   pdfmorph SrFe2As2_150K.gr . --multiple
   ```

   - The `--multiple` tag tells `PDFmorph` to compare the morphed file `SrFe2As2_150K.gr` against all PDFs in a directory. The directory we have supplied is `.`, which is shorthand for the current working directory. In our case, this is the `morphMultiple` directory.

```
(pdfmorph_env) morphMultiple $pdfmorph SrFe2As2_150K.gr . --multiple

# Input morphing parameters:
# scale = None
# stretch = None
# smear = None

# Labels: [Target] [Pearson] [Rw]
SrFe2As2_156K.gr 0.999755 0.023024
SrFe2As2_162K.gr 0.999360 0.037521
SrFe2As2_168K.gr 0.998671 0.054065
SrFe2As2_174K.gr 0.997625 0.072189
SrFe2As2_180K.gr 0.996714 0.085544
SrFe2As2_186K.gr 0.995607 0.098680
SrFe2As2_192K.gr 0.993501 0.117966
SrFe2As2_198K.gr 0.989150 0.150522
SrFe2As2_204K.gr 0.987273 0.163814
SrFe2As2_210K.gr 0.985657 0.175156
SrFe2As2_216K.gr 0.984146 0.185372
SrFe2As2_222K.gr 0.982336 0.197483
SrFe2As2_228K.gr 0.980159 0.211229
SrFe2As2_234K.gr 0.978108 0.224387
SrFe2As2_240K.gr 0.975488 0.239560
SrFe2As2_246K.gr 0.971949 0.257996
```

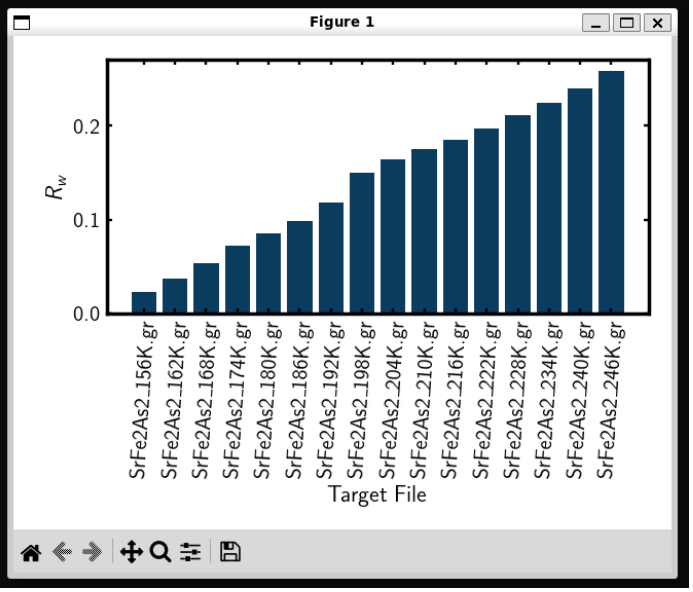

Figure 3.6: Bar chart of R_W values for each target file. Target files are listed in ASCII sort order.

3. After running this we get a chart of R_W values for the morphed file compared to each

target file. By default, a bar chart is generated. However, this plot can be difficult to interpret. We may instead wish to generate a line chart of R_W values against some numerical abscissa such as temperature. At the top of each `.gr` file in this directory is a list of parameters. These are of the form `<parameter_name> = <parameter_value>` and are located above the `r` versus `gr` table. Included in these parameters is the temperature at which each PDF was measured at.

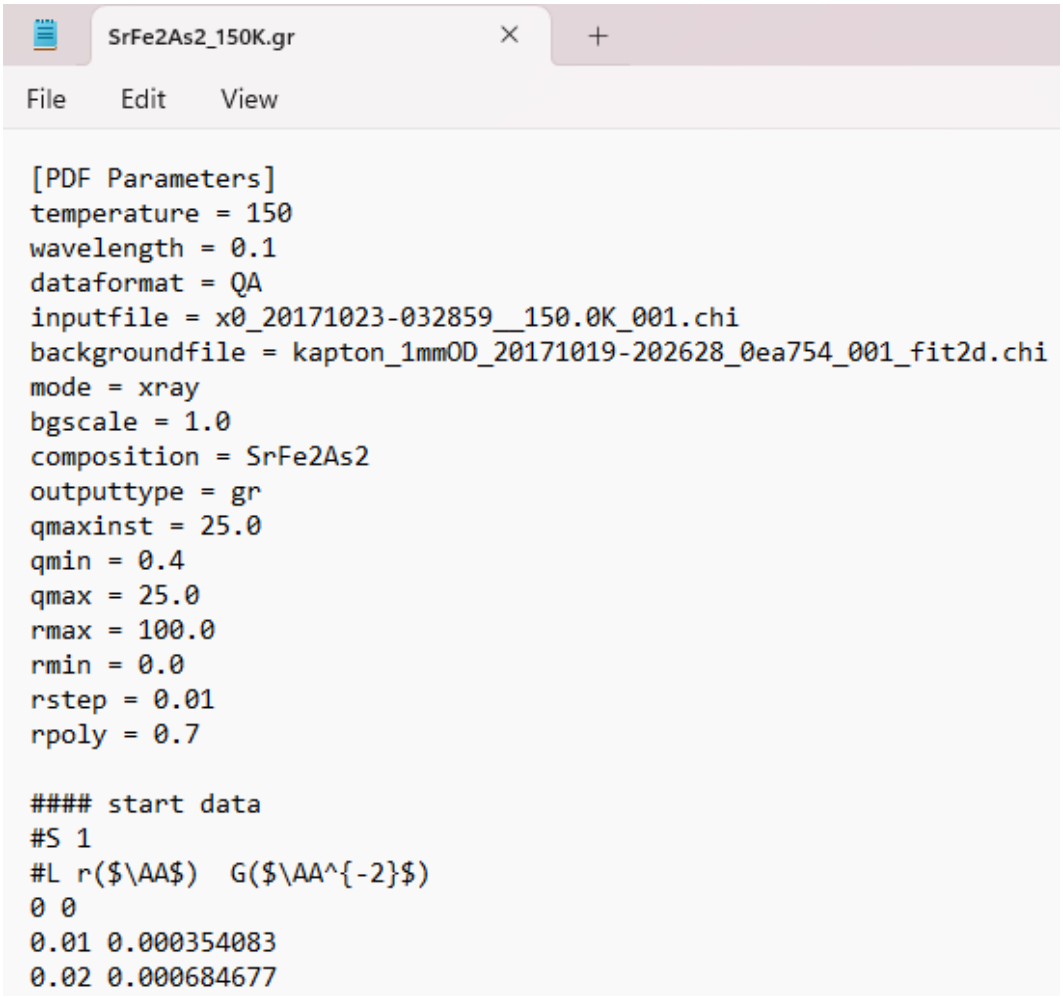

Figure 3.7: Parameters located at the top of the `SrFe2As2_150K.gr` file. Some parameters included are temperature, wavelength, and composition.

4. By running

   ```
   pdfmorph SrFe2As2_150K.gr . --multiple --sort-by=temperature
   ```

   we can sort the plotted R_W values by the temperature parameter included within each file.

   - When the value of the parameter given to `--sort-by` is numerical, a line plot is generated.
   - Otherwise, a bar chart with the parameter values sorted in ASCII sort order is

generated. You can observe this behavior by using `inputfile` as the `--sort-by` parameter instead of `temperature`.

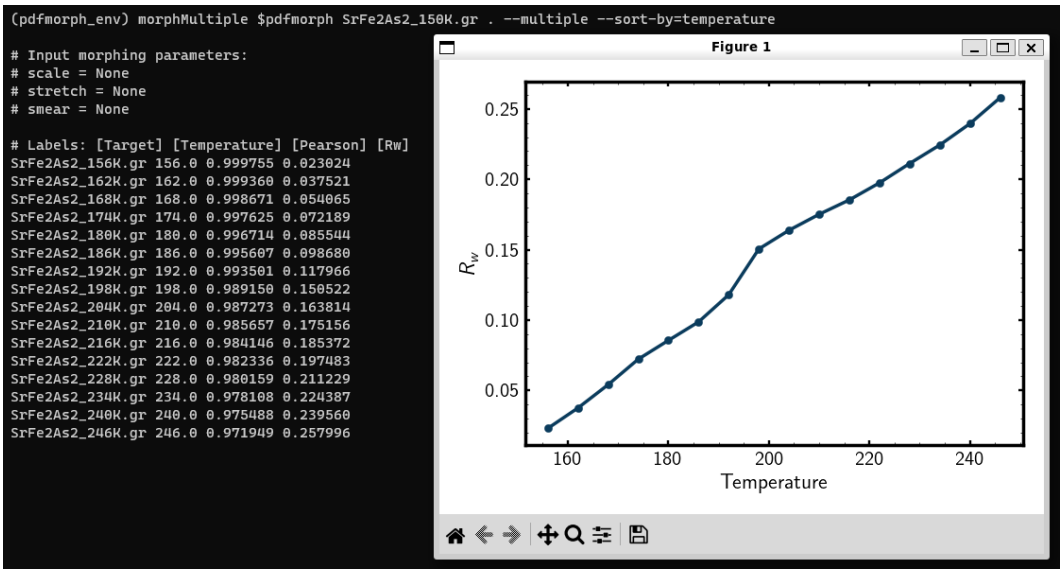

Figure 3.8: The R_W plotted against the temperature the target PDF was measured at.

5. Between $192K$ and $198K$, the R_W has a sharp change, which may be indicative of a phase change. To be more certain, let us apply morphs to take into account isotropic expansion and differences in incident flux (stretching and scaling).

   ```
   pdfmorph SrFe2As2_150K.gr . --scale=1 --stretch=0 --multiple \
   --sort-by=temperature
   ```

   The change in R_W has become more pronounced.

   - Note that we are not applying the smear morph in this example as it takes a long time to refine and does not significantly change the R_W values in this example.

6. We can also change what is being plotted in the ordinate using `--plot-parameter`. In our case, it is useful to look at the `stretch` factor

   ```
   pdfmorph SrFe2As2_150K.gr . --scale=1 --stretch=0 --multiple \
   --sort-by=temperature --plot-parameter=stretch
   ```

   We can see that the stretch factor generally increases, but from $192K$ to $198K$, there is no increase. This means `PDFmorph` found that isotropic lattice expansion due to latent thermal effects (indicated by an increase in the stretch factor) does not account for the change in R_W in this region. This is highly suggestive of an alternative source of the dissimilarity such as a phase transition. In fact, there is a structural phase transition from orthorhombic below $192K$ to tetragonal above $198K$! More sophisticated analysis can be done with PDFgui (`https://www.diffpy.org/products/pdfgui`).

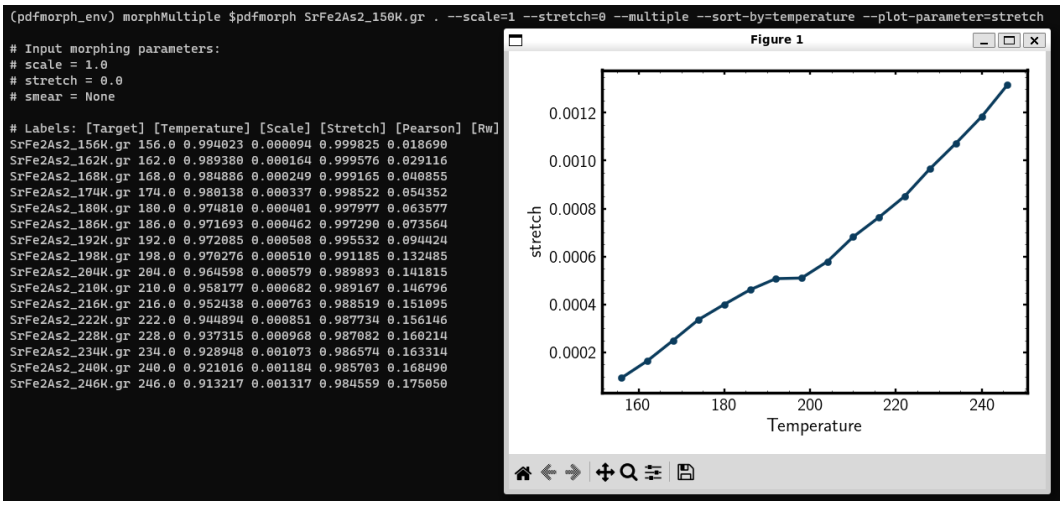

Figure 3.9: The refined stretch factor for each target PDF. There is very little change between $192K$ and $198K$.

7. Finally, let us save all the morphed PDFs into a directory named `savedMorphs`.

```
pdfmorph SrFe2As2_150K.gr . --scale=1 --stretch=0 --multiple \
--sort-by=temperature --plot-parameter=stretch \
--save=savedMorphs
```

Entering the directory with `cd` and viewing its contents with `ls`, we see a file named `Morph_Reference_Table.txt` with data about the input morph parameters and refined output parameters and a directory named `Morphs` containing all the morphed PDFs. See the `--save-names-file` option to see how you can set the names for these saved morphs!

3.2.2 Nanoparticle shape effect

A nanoparticle's finite size and shape can affect the shape of its PDF. We can use PDFmorph to morph a bulk material PDF to simulate these shape effects. Currently, the supported nanoparticle shapes include only spheres and spheroids.

- Within the `additionalData` directory, `cd` into the `morphShape` subdirectory. Inside, you will find a sample Nickel bulk material PDF `Ni_bulk.gr`. This PDF comes from *Atomic Pair Distribution Function Analysis: A Primer* (`https://github.com/Billingegroup/pdfttp_data`). Also in the directory are multiple `.cgr`, which are calculated Nickel nanoparticle PDFs.

```
(pdfmorph_env) PDFmorph_tutorial_directory $cd additionalData/
(pdfmorph_env) additionalData $ls
additionalData
(pdfmorph_env) additionalData $cd additionalData/
(pdfmorph_env) additionalData $ls
morphMultiple  morphShape
(pdfmorph_env) additionalData $cd morphShape
(pdfmorph_env) morphShape $ls
Ni_bulk.gr  Ni_nano_sphere.cgr  Ni_nano_spheroid.cgr
(pdfmorph_env) morphShape $
```

Figure 3.10: The files within the `morphShape` directory.

- In the following sections, we apply shape effect morphs on the bulk material to reproduce the calculated PDFs.

Note that there are also support for morphing a nanoparticle PDF into bulk. For more information see Chapter 5 [PDFmorph options], page 19. When applying these inverse morphs it is recommended to set `--rmax=psize` where `psize` is the longest diameter of the nanoparticle as data beyond `psize` is noise.

3.2.2.1 Spherical shape

1. The `Ni_nano_sphere.cgr` file contains a generated spherical nanoparticle with unknown radius. First, let us plot `Ni_bulk.gr` against `Ni_nano_sphere.cgr`.

   ```
   pdfmorph Ni_bulk.gr Ni_nano_sphere.cgr
   ```

2. Despite the two being the same material, the R_W is quite large. To reduce the Rw, we will apply spherical shape effects onto the PDF. However, in order to do so, we first need the radius of the spherical nanoparticle.

3. A nanoparticle cannot have bonds longer than the particle size. Thus, for r beyond the spherical diameter, we expect the PDF amplitude to be close to zero. On our plot, the nanoparticle PDF amplitude falls to zero around $r = 22$. Thus, the nanoparticle radius should be about half of that, or around 11.

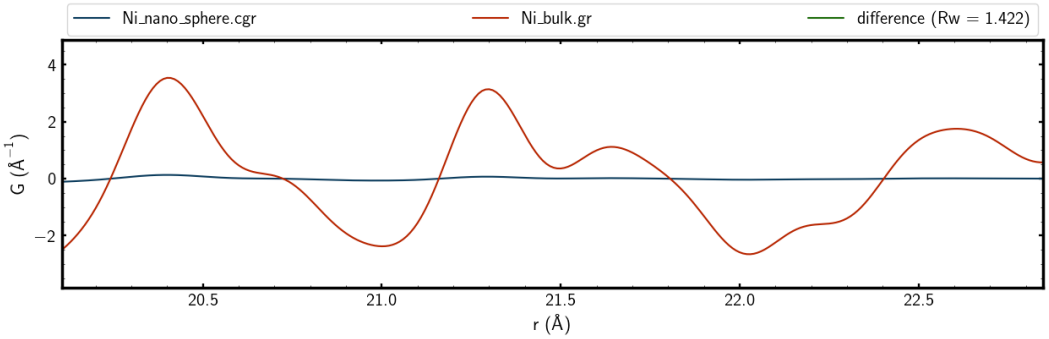

Figure 3.11: The PDF amplitude goes to zero between $r = 21$ and $r = 22$.

4. We are ready to perform a morph applying spherical shape effects. To do so, we use the `--radius` parameter

   ```
   pdfmorph Ni_bulk.gr Ni_nano_sphere.cgr --radius=11 -a
   ```

   We can see that the R_W value has decreased significantly from $R_W = 1.422$ to $R_W = 0.085232$.

5. Finally, let us refine by running the same command without the `-a` tag. The refined radius should be `radius=12.183129`.

3.2.2.2 Spheroidal shape

1. The `Ni_nano_spheroid.cgr` file contains a calculated spheroidal Niickel nanoparticle. Again, we can begin by plotting the bulk material against our nanoparticle.

   ```
   pdfmorph Ni_bulk.gr Ni_nano_spheroid.cgr
   ```

2. The nanoparticle shape of the calculated PDF is an oblate spheroid with equitorial radius of about 12 and polar radius of about 6 (this information is contained within the `Ni_nano_spheroid.cgr` file). To apply the spheroidal shape effects onto the bulk, run

   ```
   pdfmorph Ni_bulk.gr Ni_nano_spheroid.cgr --radius=12 --pradius=6 -a
   ```

   - The `--radius` option corresponds to the equitorial radius.
   - The `--pradius` option corresponds to the polar radius.

3. Run the same command without `-a` to refine. Refining should give `radius=12.183129` and `pradius=6.279252`.

4 Available morphs

In this section, we detail the available morphs and the theory behind when they can (and should) be applied. For specifics on how to use these options in `PDFmorph`, check out Chapter 5 [PDFmorph options], page 19, and the Chapter 3 [Tutorials], page 4.

4.1 Temperature-related morphs

Comparing two PDFs of the same material measured at different temperatures can produce large R_Ws and signals in the difference curves. Though this can be due to a structural phase transition across the measurements, structure-preserving changes such as isotropic expansion/compression and thermal peak broadening/thinning can produce just as large R_W values.

The following morph options are related to our discussion:

- `--stretch=STRETCH` - Stretch the abscissa by a factor $1 +$ `STRETCH`.
- `--scale=SCALE` - Scale the plotted function by a factor `SCALE`.
- `--smear=SMEAR` - Broaden the PDF peaks with a Gaussian smear of width (standard deviation) `SMEAR`.

4.1.1 Isotropic expansion

The effects of isotropic expansion/compression can be accounted for by scaling and stretching the PDF. To prove this, we will make use of the (total) radial distribution function (RDF), denoted $R(r)$. This function is related to the PDF through

$$G(r) = \frac{R(r)}{r} - 4\pi r\rho_0\gamma_0(r),$$

where ρ_0 is the atomic number density the and $\gamma_0(r)$ is the nanoparticle form factor (see Section 4.2 [Shape-related morphs], page 17). A partial RDF $R_i(r)$ is defined such that $R_i(r)dr$ gives the number of atoms in the spherical shell bounded by r and $r + dr$ centered at atom i. The total RDF for an atomic system is the average of the partial RDFs of each atom in the system[1].

$$R(r) = \frac{1}{\#\ atoms}\sum_{i}^{atoms} R_i(r).$$

Therefore, the integral of the RDF from a to b gives the number of atomic pairs per atom with a separation distance between a and b.

When a material expands isotropically by a factor α, all distances between pairs of atoms increase by a factor of α (expansion by a factor of $0 < \alpha < 1$ is considered compression). Therefore, the number of atomic pairs with separation distance between a and b before the expansion should equal the number of atomic pairs with separation distance between αa and αb after. Defining $R(r)$ to be the RDF pre-expansion and $R'(r)$ to be that post-expansion,

$$\int_a^b R(r)dr = \int_{\alpha a}^{\alpha b} R'(r)dr.$$

[1] Farrow, C.L. and Billinge, S.J.L. (2009), Relationship between the atomic pair distribution function and small-angle scattering: implications for modeling of nanoparticles. Acta Cryst. A, 65: 232-239. `https://doi.org/10.1107/S0108767309009714`

A change of variables tells us

$$\int_a^b R(r)dr = \int_{\alpha a}^{\alpha b} \frac{R(r/\alpha)}{\alpha} dr,$$

and since these relations hold for all choices of $a \leq b$,

$$R'(r) = \frac{R(r/\alpha)}{\alpha}.$$

The corresponding PDFs are

$$G(r) = \frac{R(r)}{r} - 4\pi r \rho_0 \gamma_0(r)$$

pre-expansion, and

$$G'(r) = \frac{R'(r)}{r} - 4\pi r \rho_0' \gamma_0'(r) = \frac{R(r/\alpha)}{\alpha r} - 4\pi r \rho_0' \gamma_0'(r)$$

post-expansion. Due to the expansion, the volume of the material has increased by α^3, while the total number of atoms remains the same, meaning

$$\rho_0' = \frac{1}{\alpha^3}\rho_0,$$

and the nanoparticle form function is scaled

$$\gamma_0'(r) = \gamma_0(r/\alpha)$$

(see the bottom of Section 4.2 [Shape-related morphs], page 17).

Finally, we can conclude that the PDF after expansion follows

$$G'(r) = \frac{R(r/\alpha)}{\alpha r} - 4\pi r \frac{\rho_0}{\alpha^3}\gamma_0(r/\alpha) = \frac{G(\alpha r)}{\alpha^2},$$

which is the original PDF scaled by a factor $1/\alpha^2$ and stretched by α.

4.1.2 Thermal broadening

Peaks in the radial distribution functions (see Section 4.1.1 [Isotropic expansion], page 15) obtained from measured PDFs have approximately Gaussian shapes due to Debye-Waller effects. The variance of each peak is the mean square atomic displacement factor (ADP), denoted $\bar{u^2}$ which can depend on dynamic (temperature-dependent) and static factors. Models, such as the Debye model, generally separate the two: $\bar{u^2} = A(T) + A_{static}$, where $A(T)$ increases with temperature.

When a material consists of atoms with similar masses, the $A(T)$ at each peak is approximately the same at a fixed temperature (motivated below). Therefore, an increase in temperature only serves to increase the ADP (and thus the variance of each Gaussian peak) by some fixed constant ζ^2. `PDFmorph` simulates this effect by converting the morphed PDF into an RDF, convolving the RDF with a Gaussian of variance ζ^2 centered at $r = 0$, and

converting back to a PDF. The convolution step increases the variance of each peak by ζ^2 exactly[2].

Using the Debye model[3], we can motivate the statement that $A(T)$ is similar for a material composed of similar-mass atoms. The model shows

$$A(T) = \frac{3h^2T^2}{4\pi^2 M k_B \theta_D^3} \int_0^{\theta_D/T} \frac{x}{e^x - 1} dx + \frac{3h^2}{16\pi^2 M k_B \theta_D},$$

where M is the mass of the oscillating atom, θ_D is the Debye temperature of the (crystal) material, and h and k_B are Planck's constant and Boltzmann's constant respectively. Thus, when the M for each atom is similar, $A(T)$ is also similar. Note also that $A(T)$ is monotonically increasing as a function of temperature.

4.2 Shape-related morphs

The shape and size of a nanoparticle can affect its electronic and optical properties[4]. `PDFmorph` contains tools to help a researcher identify the shape and size of a nanoparticle PDF given a PDF of a bulk sample. The researcher should select a shape-related morph (listed below) associated with a particular shape and provide the bulk sample PDF as the morphed PDF and nanoparticle PDF as the target. `PDFmorph` will then multiply the nanoparticle form factor $\gamma(r)$ for that particular shape to the bulk PDF and refine the parameters (e.g. the radius for a spherical shape) to best match the target. Significant difference curve signals or R_Ws indicate large deviations from the desired shape, while small signals allow the user to extract size parameters (e.g. the radius of the sphere) from the fit.

This approach has been used to estimate diameters of spherical CdSe nanoparticles consistent with those obtained from transmission electron microscopy, ultraviolet-visible spectroscopy, and photoluminescense measurements[5].

The available shape morphs are listed below:

- `--radius=RADIUS` - Multiply the PDF by the nanoparticle form factor for a sphere of radius `RADIUS`. If used with `--pradius`, multiply the PDF by the nanoparticle form factor for a spheroid of equitorial radius `RADIUS` and polar radius `PRADIUS`.
    - The sphere form factor was computed by Kodama et al.[6].

[2] Bromiley, P. (2003). Products and Convolutions of Gaussian Distributions.

[3] Dinnebier, R.E. and Billinge, S.J.L. (2018). Overview and principles of powder diffraction. In International Tables for Crystallography (eds C.P. Brock, T. Hahn, H. Wondratschek, U. Müller, U. Shmueli, E. Prince, A. Authier, V. Kopský, D.B. Litvin, E. Arnold, D.M. Himmel, M.G. Rossmann, S.R. Hall, B. McMahon, M.I. Aroyo, C.J. Gilmore, J.A. Kaduk, H. Schenk, C.J. Gilmore, J.A. Kaduk and H. Schenk). `https://doi.org/10.1107/97809553602060000935`

[4] Singh, M., Goyal, M., & Devlal, K. (2018). Size and shape effects on the band gap of semiconductor compound nanomaterials. Journal of Taibah University for Science, 12(4), 470–475. `https://doi.org/10.1080/16583655.2018.1473946`

[5] Masadeh, A. S., Božin, E. S., Farrow, C. L., Paglia, G., Juhas, P., Billinge, S. J. L., Karkamkar, A., & Kanatzidis, M. G. (2007). Quantitative size-dependent structure and strain determination of CdSe nanoparticles using atomic pair distribution function analysis. Phys. Rev. B, 76(11), 115413. `https://doi.org/10.1103/PhysRevB.76.115413`

[6] Kodama, K., Iikubo, S., Taguchi, T., & Shamoto, S. (2006). Finite size effects of nanoparticles on the atomic pair distribution functions. Acta Crystallographica Section A, 62(6), 444–453. `https://doi.org/10.1107/S0108767306034635`

- `--pradius=PRADIUS` - Multiply the PDF by the nanoparticle form factor for a spheroid of equitorial radius `RADIUS` and polar radius `PRADIUS`.
  - The spheroid form factor was computed by Lei et al.[7].
- `--iradius=IRADIUS` - Divide the PDF by the nanoparticle form factor for a sphere of radius `IRADIUS`. If used with `--ipradius`, divide the PDF by the nanoparticle form factor for a spheroid of equitorial radius `IRADIUS` and polar radius `IPRADIUS`.
- `--ipradius=IPRADIUS` - Divide the PDF by the nanoparticle form factor for a spheroid of equitorial radius `IRADIUS` and polar radius `IPRADIUS`.

In the rest of this section, we will detail how the nanoparticle form factor is computed. We first need to define the shape function $s\vec{r}$, which is 1 within the nanoparticle and 0 outside. The unaveraged particle form factor $\gamma_0(\vec{r})$ is the autocorrelation of this shape function averaged over the nanoparticle volume V.

$$\gamma_0(\vec{r}) = \frac{1}{V}\int\int\int s(\vec{r}')s(\vec{r}'+\vec{r})d\vec{r}'.$$

In this manual, we refer to the angle-averaged version, denoted $\gamma_0(r)$, as the nanoparticle form factor.

$$\gamma_0(r) = \frac{\int d\phi \int d\theta \sin(\theta) r^2 \gamma_0(\vec{r})}{\int d\phi \int d\theta r^2 \sin(\theta)}.$$

This form factor is particularly useful as one can approximate the nanoparticle PDF $G_{nano}(r)$ by multiplying the proper nanoparticle form factor with the bulk PDF $G_{bulk}(r)$[8] as follows

$$G_{nano}(r) = \gamma(r)G_{bulk}(r).$$

Finally, an important property of the nanoparticle form factor $\gamma_0(r)$ that we use in Section 4.1.1 [Isotropic expansion], page 15, is that, after undergoing isotropic expansion by a factor α, the new form factor $\gamma'(r)$ is a stretched version of the original form factor: $\gamma_0'(r) = \gamma_0(r/\alpha)$. To show this, consider the new shape function $s'(\alpha\vec{r}) = s(\vec{r})$ as each point $\vec{r}$ pre-expansion is mapped to a point $\alpha\vec{r}$ post-expansion. The volume of the nanoparticle also increases by a factor α^3, so

$$\gamma_0'(\vec{r}) = \frac{1}{\alpha^3 V}\int\int\int s'(\vec{r}')s'(\vec{r}'+\vec{r})d\vec{r}'.$$

Aplying a change of variables $\vec{r}' \to \vec{r}'/\alpha$ gives

$$\gamma_0'(\vec{r}) = \frac{1}{\alpha^3 V}\int\int\int s'(\alpha\vec{r}')s'(\alpha\vec{r}'+\vec{r})\alpha^3 d\vec{r}',$$

and substituting $s'(\alpha\vec{r}) = s(\vec{r})$ gives

$$\gamma_0'(\vec{r}) = \frac{1}{V}\int\int\int s(\vec{r}')s(\vec{r}'+\vec{r}/\alpha)d\vec{r}' = \gamma_0(\vec{r}/\alpha).$$

Taking the angle-averaged integrals gives the desired relation $\gamma_0'(r) = \gamma_0'(r/\alpha)$.

[7] Lei, M., de Graff, A. M. R., Thorpe, M. F., Wells, S. A., & Sartbaeva, A. (2009). Uncovering the intrinsic geometry from the atomic pair distribution function of nanomaterials. Phys. Rev. B, 80(2), 024118. `https://doi.org/10.1103/PhysRevB.80.024118`

[8] Farrow, C.L. and Billinge, S.J.L. (2009), Relationship between the atomic pair distribution function and small-angle scattering: implications for modeling of nanoparticles. Acta Cryst. A, 65: 232-239. `https://doi.org/10.1107/S0108767309009714`

5 PDFmorph options

In this section we describe all the options available to use in the `PDFmorph` program. Without options, the `PDFmorph` program is run with the command

```
pdfmorph <morphed_PDF> <target_PDF>
```

where morphs will be applied to the morphed PDF in attempt to minimize the residual with the target PDF.

`-h, --help`
Show a brief summary of all the options listed in this section.

`-V, --version`
Show the program version (does not run the program).

`-s NAME, --save=NAME`
Save the morphed PDF into a file named `NAME`. You can use `-` to save to `stdout` instead. When you have the `--multiple` tag enabled, multiple morphed PDFs are generated. Using this command with the `--multiple` tag will save all these morphs into a directory named `NAME` as well as a `.txt` file summary of refined morph parameters (if applicable) and R_W for each morph done. To specify names for each saved PDF file, use the `--save-names-file` option.

`-v, --verbose`
Increase the amount of information saved to the file(s) generated by `--save`. Without this option enabled, only the morphed PDF table will be saved (the table of r and corresponding $G(r)$ values). When enabled, information about the input and refined morph parameters (when applicable) and the R_W and Pearson correlation coefficient will be saved as parameters above the PDF table.

`--rmin=RMIN`
The minimum r-value to use for PDF comparisons and manipulations.

`--rmax=RMAX`
The maximum r-value to use for PDF comparisons and manipulations.

`--pearson`
When enabled, refinement will seek only to maximize the Pearson correlation coefficient between the two PDFs.

`--addpearson`
When enabled, refinement will seek to both maximize the Pearson correlation coefficient and minimize the residual between the two PDFs.

Morphs and manipulations

These options select the morphs and manipulations that are to be applied to the morphed PDF. The passed values will be refined unlesss specifically excluded with the `--apply` or `--exclude` options. If no morphs are selected, the morphed PDF will be unchanged and plotted against the target PDF.

`-a, --apply`
Apply the morph but do not refine the morph parameter value. When this is not enabled, `PDFmorph` will automatically refine the morph parameters for all morph options below.

`-x MORPH, --exclude=MORPH`
Do not refine the morph named `MORPH`. Note that the input `MORPH` must be lower case

and match exactly the name of the morph option that you wish not to refine. For example, running

```
pdfmorph <MORPHED> <TARGET> --scale=2 --smear=0.1 --exclude=scale
```

will scale the morphed PDF by exactly 2, but will refine the parameter 0.1 for smear. However, `--exclude=Scale` will not stop `--scale=2` from being refined.

`--scale=SCALE`

Apply a scale factor `SCALE` to the function being plotted. For instance, scaling $G(r)$ by `SCALE` will return `SCALE` $* G(r)$.

`--smear=SMEAR`

Smear the PDF peaks with a Gaussian of width (standard deviation) abs(`SMEAR`). This input can be negative, but will have the same effect as its absolute value. This function is designed only for use on PDFs. This operation assumes the RDF (see Chapter 4 [Available morphs], page 15) peaks are approximately Gaussian and works by converting the PDF to an RDF, convoluting a Gaussian of width `SMEAR` with the RDF, and converting back to the PDF. This conversion requires a parameter `BASELINESLOPE` which will be refined if not provided by the user in the `--slope` option.

`--stretch`

Stretch the function being plotted along the abscissa by a factor of 1 + `STRETCH`. For example, if the original function is $G(r)$, the stretch returns $g(r/(1 +$ `STRETCH`$))$. The returned PDF will always only be defined between `RMIN` and `RMAX` (see `--rmin` and `--rmax`). The `STRETCH` coefficient must be larger than -1 and values in the range $(-1, 0)$ will compress the function and set the remaining portion of the function up to `RMAX` as zero.

`--slope=BASELINESLOPE`

The slope of the baseline used when applying the smear factor. Unless excluded using `--apply` or `--exclude`, it will be refined whenever `--smear` is used, even if `--smear` is not being refined. The equation `BASELINESLOPE` $= 4\pi r\rho_0\gamma_0$) can be used if the atomic number density ρ_0 and the nanoparticle form factor γ_0 is known for the morphed PDF.

`--qdamp=QDAMP`

Dampen the PDF by a factor `QDAMP`. A limited Q-resolution of the diffractometer requires us to apply a Gaussian dampening envelope centered at $r = 0$ with width (standard deviation) `QDAMP`. This envelop is directly multiplied to the PDF.

`--radius=RADIUS`

Apply the nanoparticle form factor γ_0 for a sphere of radius `RADIUS`. If `PRADIUS` is also specified, instead apply the characteristic function of a spheroid with equitorial radius `RADIUS` and polar radius `PRADIUS`.

`--pradius=PRADIUS`

If `RADIUS` is also specified, see `--radius`. Otherwise, apply the characteristic function of a sphere with radius `PRADIUS`.

`--iradius=IRADIUS`

Apply the inverse characteristic function $1/\gamma_0$ of a sphere of radius `IRADIUS`/ If `IPRADIUS` is also specified, instead apply the characteristic function of a spheroid with equitorial radius `IRADIUS` and polar radius `PRADIUS`.

`--ipradius=IPRADIUS`

If `IRADIUS` is also specified, see `--iradius`. Otherwise, apply the characteristic function of a sphere with radius `IPRADIUS`.

Plot options

These options control plotting. The morphed and target PDFs will be plotted against each other with a difference curve shown below. The following changes occur when `--multiple` is enabled. (1) The R_W for each morphed PDF compared to the target will be plotted unless another parameter is specified by `--plot-parameter`. (2) The plot will be a bar chart where the abscissa names are the file names of the target PDFs unless otherwise specified by `--sort-by`. If the parameter given to `--sort-by` has numerical value, the plot will be a line chart; otherwise, a bar chart the parameter values as the abscissa names will be plotted. By default, ASCII sort order is used for the bar chart abscissa names, but `--reverse` can be used to reverse the order.

`-n, --noplot`
Do not display a plot if enabled. Otherwise, show a plot.

`--mlabel=MLABEL`
Set the label for the morphed PDF to `MLABEL` on the plot. If not specified, the morphed file name will be used as the label.

`--tlabel=TLABEL`
Set the label for the target PDF to `TLABEL` on the plot. If not specified, the target file name will be used as the label.

`--pmin=PMIN`
The minimum r-value (abscissa) to plot. If not specified, `RMIN` will be used.

`--pmax=PMAX`
The maximum r-value (abscissa) to plot. If not specified, `RMAX` will be used.

`--maglim=MAGLIM`
The function $G(r)$ will be magnified by `MAG` for $r >$ `MAGLIM`. This may be useful as PDF amplitude can get very small for large r. No magnification will take place if `--maglim` is not specified.

`--mag=MAG`
See `--maglim`. No magnification will take place if `--maglim` is not specified.

`--lwidth=LWIDTH`
Set the line thickness of the plotted curves. If not specified, it will default to 1.5.

Multiple morphs

`PDFmorph` allows one to morph one PDF against multiple different targets when `--multiple` is enabled. See `-s` and the description under "Plot options" for how saving and plotting change when `--multiple` is enabled.

`--multiple`
Changes usage of `PDFmorph` to

```
pdfmorph <MORPHED_FILE> <TARGET_DIRECTORY>
```

where the PDF in the file `MORPHED_FILE` will be morphed with each (PDF) file in the directory `TARGET_DIRECTORY` as the target. Files in `TARGET_DIRECTORY` will be sorted in ASCII sort order order unless a sorting parameter is specified by `sort-by`.

`--sort-by=FIELD`
Used with `--multiple`. Sort the files in `TARGET_DIRECTORY` by some parameter named

`FIELD`. Parameters can be specified within each target PDF file by lines of the form `<PARAM_NAME> = <PARAM_VALUE>` in the header (anywhere above the r versus $G(r)$ data table). `PDFmorph` will attempt to find a parameter named `FIELD` using a case-insensitive search. Numerical `PARAM_VALUE` will be sorted in ascending order and non-numerical ones will be sorted in ASCII sort order.

`--reverse`

Used with `--multiple`. Sort the files in `TARGET_DIRECTORY` in reverse ASCII sort order. If a parameter is given by `--sort-by`, reverse the order given by `--sort-by`.

`--serial-file=SERIALFILE`

Used with `--multiple` and `--sort-by`. Look for `FIELD` in a serial file named `SERIALFILE` instead. Only serial file types supported by diffpy.utils (`https://www.diffpy.org/diffpy.utils`) such as `.json` are allowed.

`--save-names-file=NAMESFILE`

Used with `--multiple` and `-s`. Specify names for each manipulated PDF when saving using a serial file named `NAMESFILE`. The format of `NAMESFILE` should be as follows: (1) Each target PDF file name is an entry in `NAMESFILE`. (2) For each entry, there should be a key `save_morph_as` whose value specified the name to save the manipulated PDF as. An example `.json` `NAMESFILE` is below.

```
{
  "target1.gr": {
    "save_morph_as": "name.extension"
  },
  "target2.gr": {
    "save_morph_as": "another_name.extension",
    "other_information": "[optional]"
  },
  "target3.gr": {
    "SaVe_MoRph_As": "this_also_works.extension",
    "note": "Capitalization does NOT matter. This is case insensitive."
  }
}
```

Only serial file types supported by diffpy.utils (`https://www.diffpy.org/diffpy.utils`) such as `.json` are allowed.

`--plot-parameter=PLOTPARAM`

Used with `--multiple` and when plotting is enabled. Choose a parameter `PLOTPARAM` to plot for each morph. When not specified, the R_W values for each morphed PDF compared to the target PDFs will be plotted. This option is not case sensitive meaning `--plot-parameter=Scale` and `--plot-parameter=scale` will both cause the parameter `scale` to be plotted.

Index