1. PDF Introduction - Basic Refinements

Introduction

Here we aim to introduce you to all of the basic TOPAS and PDF functions that you will use in most, if not all, refinements that you will conduct. The machines in use at this workshop have already been configured in such a way that they should work for all tutorials going forward. You will have to setup your own machine following the instructions provided by John Evans' jEdit setup and Philip Chater's PDF github (download) and tut orials.

Macros

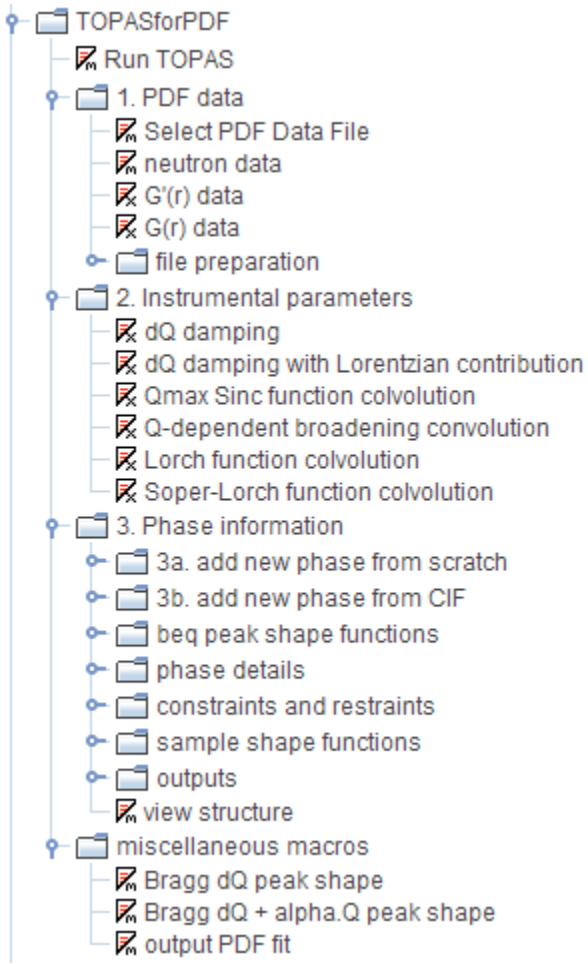
Topas utilises a simple system of built-in keywords (such as `pdf_data`, `space_group`, `r_wp`) and defined macros (such as `dQ_damping`, `beq_spherical`, `Damp`, and `pdf_for_pairs`). The user is able to define their own macros, and the `#include` statement is the method by which macros are utilised without them needing to be defined within the current input file.

By default, Topas comes with a selection of macros which are held within the file topas.inc in the main topas directory. By default, this file is consulted when you run a refinement; so if you request the macro Out_Ycalc it is found within topas.inc and implemented.

jEdit PDF Menu

The XInsert menu on the left hand side has already been configured on your training machines, and provides you with a helpful way of accessing various macros defined within the .inc files. You will see the word XInsert vertically on the very left side, click this button to toggle whether the menu appears. We typically make our topas input files using these menus, although of course you can just type the commands straight into the editor window if you know them! Double clicking on an item will insert the relevant lines wherever the cursor is. The location of keywords and macro calls can be important - make sure you have your cursor in the correct position before selecting your desired options.

Here is an image of the menu expanded down a couple of levels. We're going to build our first PDF refinement using these menus now, looking at a nanosolution of CeO_2 in a solvent.



Basic PDF Refinement

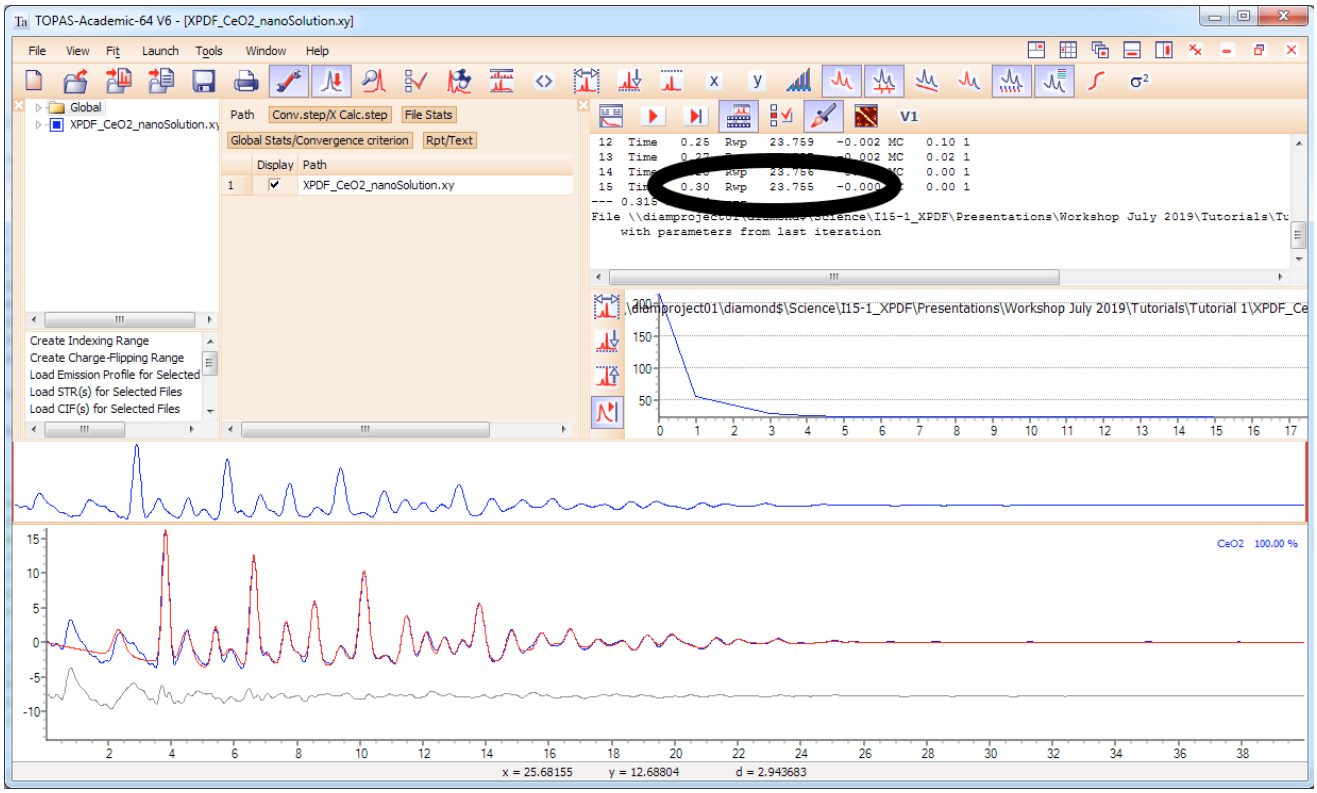
1. Start by running topas if it isn't already running, by double-clicking *Run TOPAS* at the top.
2. Double click *1. PDF data > Select PDF Data File* to browse for the file we are going to refine a model against; in this case you the pop up window to find the file "XPDF_CeO2_nanoSolution.xy" in the tutorial directory. Select this file and a new inp file will be created in a new buffer in jEdit. You should have something like this:

```
'=== FILE HEADER ===
r_wp 0 r_exp 0 r_p 0 r_wp_dash 0 r_p_dash 0 r_exp_dash 0 weighted_Durbin_Watson 0 gof 0
iters 100000 'Maximum number of iterations of refinement
chi2_convergence_criteria 0.001 'Stop criteria for refinement
'do_errors 'Reports errors for each refined value
'continue_after_convergence 'Continues refinement even after reaching the stop criteria

'=== PDF FILE ===
xdd XPDF_CeO2_nanoSolution.xy

'=== FILE PREPARATION ===
   pdf_data
   weighting 1
```

3. where `r_wp` etc are various R-factors (you can check the technical reference for definitions of these; click the *TOPAS_v6_Menus > Help > Technical Reference* to open it); `iters` and `chi2_convergence_criteria` are described in the file with comments denoted by strings `'starting with a single quote`; `xdd` declares a new data file to be refined against, `pdf_data` says that it's PDF data, and `weighting 1` says that every point should be equally weighted in the residual equation (again, check the technical reference for more on this).
4. We also need to apply some `xdd` level convolutions such as an instrument convolution, so from select *2. Instrumental Parameters > dQ damping* and start with a value of 0.05.
5. Now we need to insert our starting model for the refinement. click *3. Phase Information > 3b Add new phase from CIF > Read a .CIF file* and locate the file ceo2.cif in the tutorial 1 directory.
   a. The name is read from the metadata in the cif file, and for some reason now includes question marks. Give this phase a better name.
   b. We want to refine the lattice parameters, but maintain the cubic symmetry of the phase that has been imported from the cif. Topas will allow you to break the metric symmetry by refining a, b, c, al, be, ga even if this is not in keeping with the space group, so we need to tell topas which lattice parameters to refine, click *3. Phase Information > 3b Add new phase from CIF > ii. constrain lattice parameters > convert to cubic* and an @ is added to the line defining the a lattice parameter, and the Get(a) command is used to declare that b and c are always equal to a, maintaining the cubic metric symmetry.
   c. add a scale factor by double clicking *3. Phase Information > 3b Add new phase from CIF > scale factor*
6. As with most materials we'll encounter, this sample also includes some local correlated motion. delete the beq 0.xxxxxx parts at the end of the two site lines (which would be fine for a Rietveld refinement), and replace with *3. Phase Information >beq peak shape functions > beq spherical with rmin and rlo cutoffs.* We need to allow these beq macros to refine some variables independetly, so change your site lines to like the below. We'll do a whole tutorial on this later so don't worry too much about it now.

```
site Ce1  x 0      y 0     z 0     occ Ce   1 beq_rcut_rlo_spherical(!rcut,3.0,beqcut,0.1,!rlo,5.0,beqloCe,0.2,beqhiCe,1,radius ,50)
site O1   x 0.25   y 0.25  z 0.25  occ O    1 beq_rcut_rlo_spherical(!rcut,3.0,beqcut,0.1,!rlo,5.0,beqloO, 0.2,beqhiO, 1,radius ,50)
```

7. This phase is actually a nanoparticle, so we need to define a real-space shape function to reflect this. Again, we'll do more tutorials on this later, but for the moment just double click *3. Phase Information > sample shape functions > spherical damping* and start with a radius of 50 Å. We want the *beq_spherical* radius and the `spherical_damping` radius to be equal, so replace the @ with the word `beqradius`, since this is what we called it within the `beq_rcut_rlo_spherical` macros above.
8. Save the file (which by default will have a name related to the file we started with) and click the button to send this input file to topas. Now go to topas and run the refinement, but do not update the input file when prompted *Always assume that in these tutorials, we don't want you to update the input file unless we explicitly say otherwise!*
9. The fit should be ok with an r_wp of ~23.7



10. There's some signal at low-r that is not accounted for by the model however, and since we know that this is a nanosolution we should try and take the solvent into account. Handily, we collected the PDF of just the solvent and we can add this in to topas and fit that as well.
11. The command for this is not in the PDF menus! we'll have to type it in by hand. Add `bkg_file(XPDF_Solvent.xy,@,1)` at the bottom of the file. This is a macro which imports a file (in this case, XPDF_Solvent.xy, also located in your tutorial 1 directory) and refines a scale factor for this "background" (which starts at 1 in this case).
12. Refine this model. It's made the fit better, but now we can't tell the difference between what signal comes from the CeO2 and which is the solvent component. Lets add a little more information about the solvent so we can ask topas to fit them separately.
13. Back to jEdit, and we'll add a section of code which we can turn on and off easily should we wish to. Copy the below lines into your input file, making sure to only include one bkg_file line.

```
#define proper_solvent_scaling

#ifdef proper_solvent_scaling
   dummy_str
    phase_name "Solvent"
    Cubic(4.9276)
    space_group "Fm3m"
      site O     x 0        y 0      z 0        occ O  1
      site H     x 0.25     y 0.25   z 0.25     occ H  1
        Phase_Density_g_on_cm3( 1.00010)
      scale = fit_obj_scale;
      user_y solvent XPDF_Solvent.xy
      fit_obj = fit_obj_scale solvent;
    prm fit_obj_scale  1
      fit_obj_phase 2
#else
    bkg_file(XPDF_Solvent.xy,@,1)
#endif
```

14. the `#define proper_solvent_scaling` line makes a definition that the `#ifdef` command depends on. If proper_solvent_scaling is defined, then the section of code in the `#ifdef` section is included in the refinement, otherwise it is not. Try running the refinement with and without the `proper_solvent_scaling` definition, either by changing the name on the `#define` line, or commenting it out entirely. Why is this better than just commenting out the section of code?
15. The rest of the commands in this block are an attempt to try and get a meaningful scale factor from the refinement, don't worry too much about them at this point.
16. With the definition included, run the refinement in topas. Highlight the phases in the top right of the fit window to show the contributions from the nanoparticle and the solvent.