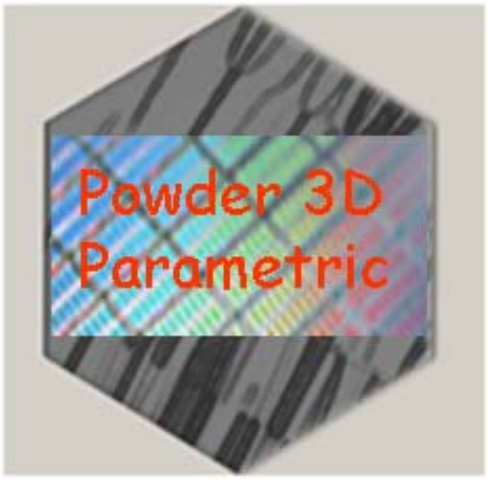
Powder 3D Parametric

A tutorial



Max Planck Institute for Solid State Research,
Heisenbergstrasse 1, Stuttgart

1. Introduction

Powder 3D Parametric (Rajiv *et al*., 2010) is a program for fast and automated sequential and parametric Rietveld refinements (WPPR) (Stinton and Evans, 2007) of large number of data collected in 2D powder diffraction experiments. The program interacts with the launch mode kernel of total pattern analysis software Topas (Coelho, 2007) to perform the refinements. Powder 3D Parametric provides many graphical user interfaces, using them the Topas input files (*.*inp*) files required for the sequential and parametric refinements can be easily prepared and/or executed.

This program requires the Topas user's license dongle (version 3 or higher) for its successful execution.

2. Installation

Unpack the file "Powder3DParametric.zip" and extract all its contents to a desired folder (e.g. D:\Extracted). Now select the context menu (right click, Figure 0) of the shortcut "Powder3DParametric.exe".

Change the Target path to "D:\Extracted\bin\bin.x86\idlrt.exe".

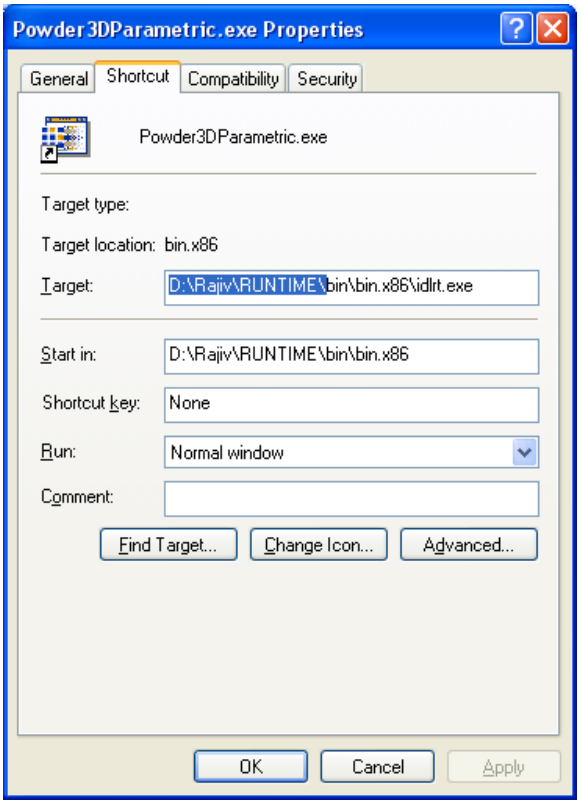


Figure 0 The context menu (right click) of the shortcut **'Powder 3D Parametric.exe'**.

Now execute the program by double clicking on 'Powder3DParametric.exe'  . Click on 'Click to continue'. The welcome screen ('**Powder 3D Parametric Graphics'** interface: Figure 1)[1] of the program will popup.

3. Loading data

3.1 Select files

To load the data, click on the '**File'** menu (in the '**Powder 3D Parametric Graphics'** interface: Figure 1), select the '**Import'** option and choose the corresponding experimental file format.

The program supports the following data file formats:

CHI, XY, XYE (DASH' files), GDA (GSAS files), DAT (Fullprof files), UXD (Bruker) and CPI (Sietronics).

The multiple data files to be refined must be selected and loaded simultaneously.

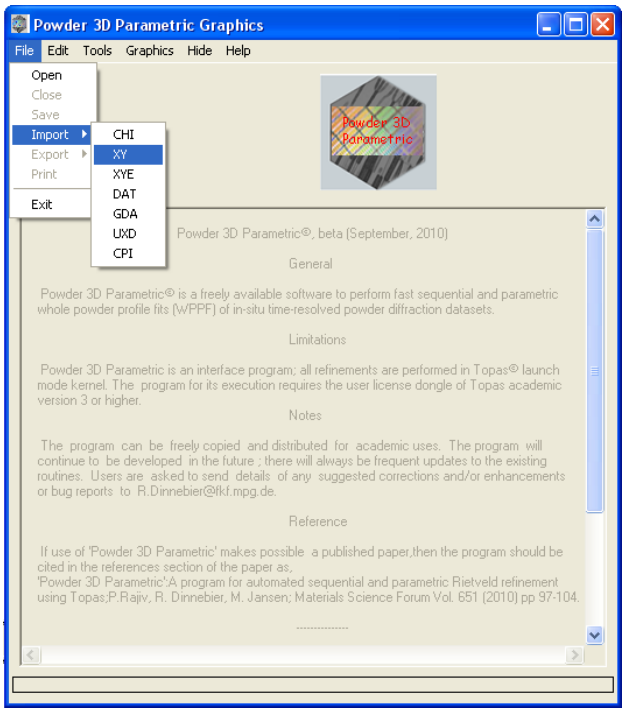
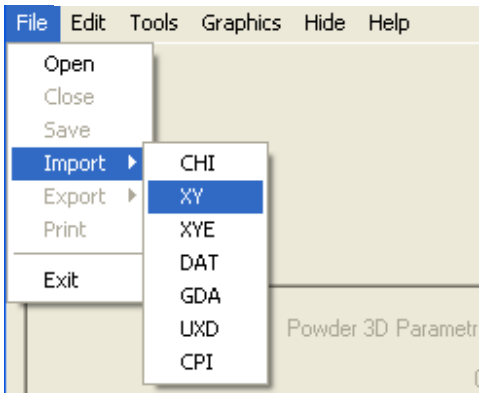


Figure 1 The **'Powder 3D Parametric Graphics'** interface and the **'File'** menu used for loading the data files.



[1] Some conventions used in this manual: The words typed in bold letters (e.g., **'Powder 3D Parametric Graphics**' GUI, '**File**' menu '**Load**' button etc.) represent the widget tools used in the GUIs of the program. The words typed in italics (e.g., **'***iters*', '*ctrl*', '*convolution_step*', '*scale'*) are the keywords associated with Topas macro language.

3.2 View loaded files

1 After loading the data files, the some information regarding the number of files, their location etc., will be displayed in the '**Database information'** GUI (Figure 2). Click '**OK'** to proceed further.

2 The 2θ vs I plot of the first file (in the 2D dataset) will appear on the '**Graphics'** tab of the '**Powder 3D Parametric Graphics'** interface (plotted in Figure 2).

3 The data loaded in Figure 2 is a high temperature sample data (Sample_data.zip) provided along with "Powder3DParametric.zip". The experimental details are in the file 'Expt.txt' in the folder 'Sample_data' in Sample_data.zip file.

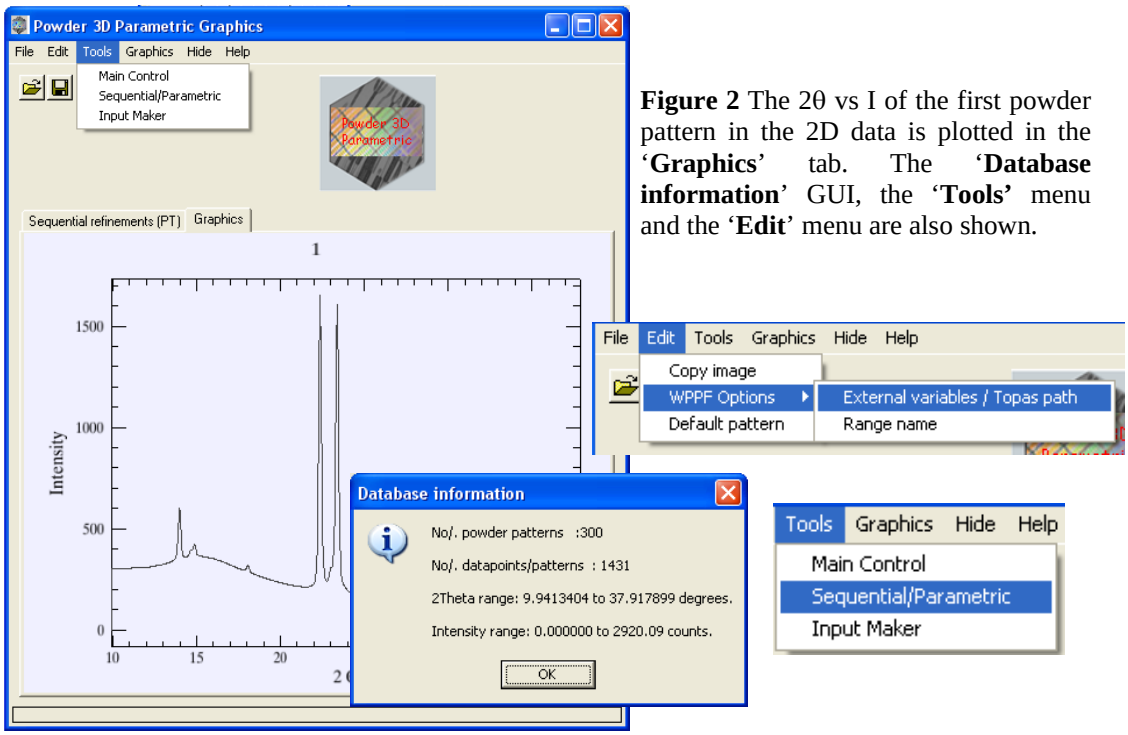


Figure 2 The 2θ vs I of the first powder pattern in the 2D data is plotted in the '**Graphics**' tab. The '**Database information**' GUI, the '**Tools'** menu and the '**Edit**' menu are also shown.

4 Now, click on the '**Tools**' menu (Figure 2) and select '**Sequential/Parametric**'. A new interface with the title '**Basic information**' will appear (Figure 3).

4. Basic experimental information

1 In the '**Basic information**' window (Figure 3), use the drop-down menu '**External variable**' to choose the external variable (e.g., Temperature)**.** The default external

variable name is '**Pattern No**'. (This will be used as the abscissa title in all the graphics).

2 Enter the range of external variable in the text boxes '**External variable Range**', e.g., Temperature '**From'** 300 K '**To'** 400 K. (Make sure that the units are correct).

3 If the measurements are made at variable (uneven) steps (as in case of some high pressure experiments) then select '**variable interval**' checkbox, to expand the '**Basic information**' window (Figure 3).

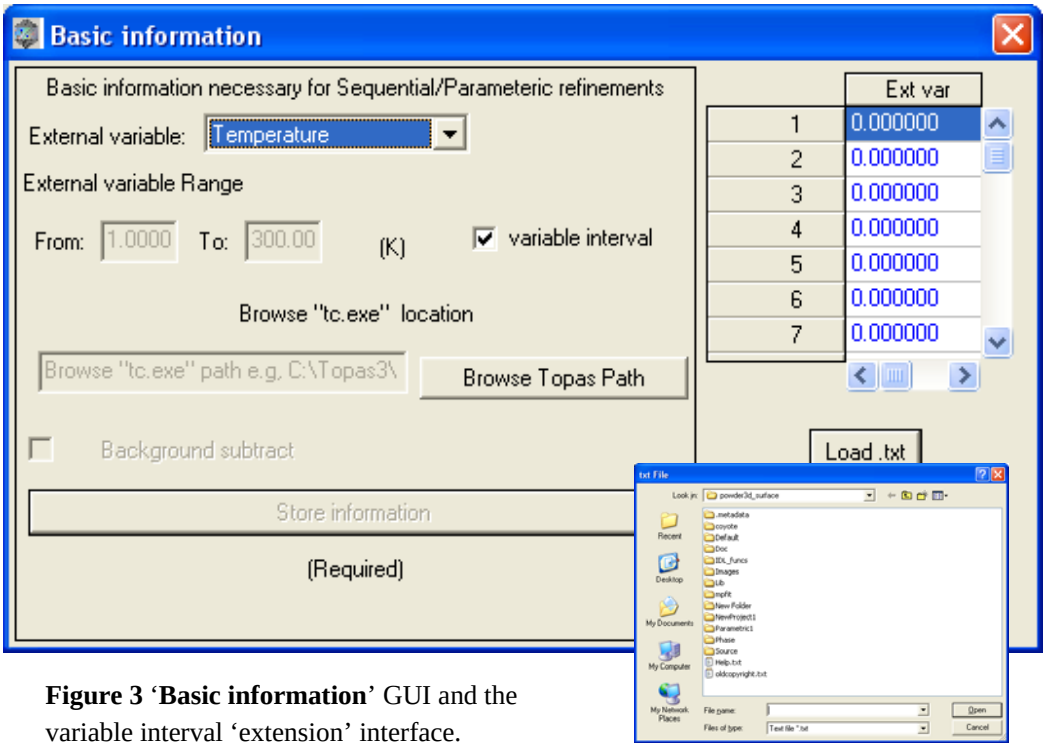


Figure 3 '**Basic information**' GUI and the variable interval 'extension' interface.

4 The variable measurement steps can either be entered manually in a table ('**Ext var**' in Figure 3) or can be loaded as an ASCII file ('*.txt') using the '**Load. txt**' button (Figure 3).(Note that the prepared *.txt file must contain the external variables entered in succeeding lines).

5 Now, locate the Topas launch mode executable ('tc.exe') using '**Browse Topas Path**' button (Figure 3). By default the 'tc.exe' will be located in the Topas working directory (e.g., D:\Topas 4-1 or C:\ Topas 3). It is obligatory to give the correct path of the 'tc.exe' file for successful sequential/parametric refinements.

6 Make sure that all the information is correctly entered and then click on the '**Store information**' button (Figure 3) to save them. (Note that the '**Store information**' button will become active only after locating the 'tc.exe' path).

7 Prompting the '**Store information**' button opens the '**Topas input file builder**' interface (Figure 4, right). This GUI contains several tabs where the Topas input files can be built with the help of various widget tools provided.

8 The stored information can be modified at anytime in the '**Edit'** menu of the '**Powder 3D Parametric Graphics**' interface. **Edit**>**WPPF Options**> **External variables / Topas path** ('**Edit**' menu in Figure 2).

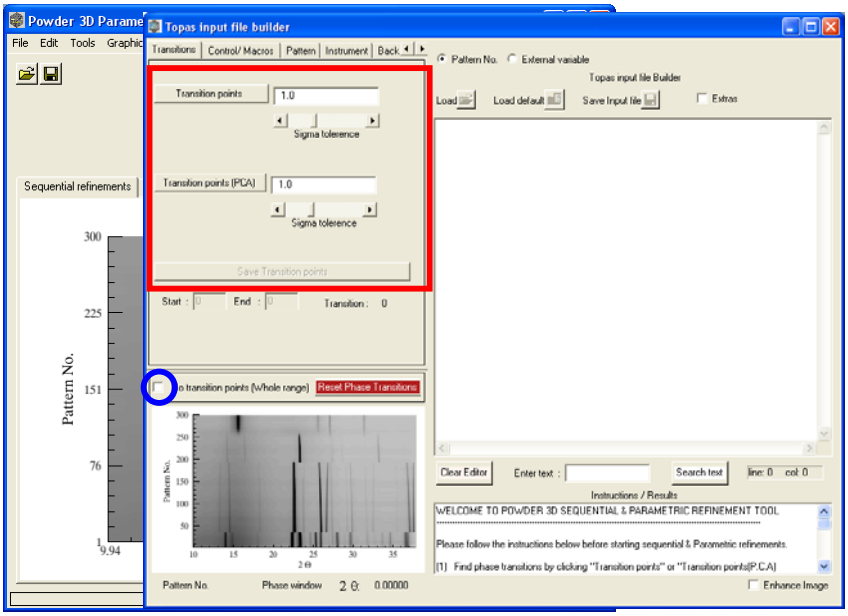


Figure 4 The '**Topas input file builder'** (right) pops up after storing the experimental information. '**Transition points**' (red framed) and '**No transition points (Whole range)**' checkbox (circled in blue) are highlighted.

5. Phase transition points

Before proceeding with the sequential/Parametric refinements, the user must provide the information whether the compound undergoes a reconstructive type of phase transition[2] (Chung *et al.*, 1993) or not. The purpose of this step is to cluster the powder patterns that belong to the like phases and also those patterns that correspond to the mixed phases

[2] Reconstructive phase transitions involve extensive rearrangements of atoms in the unit cell with the change in time/temperature. The transitions are reflected clearly in the 2D dataset as boundaries (e.g. the high temperature phase transitions of Ag_3VO_4 (The sample dataset used in the first example of this tutorial Figures 1 to 11).

(e.g., the overlapped part in Figure 6) and prepare the two groups for independent sequential or parametric refinements.

If the compound shows reconstructive type of phase transitions then the following steps (Option 1) must be followed.

Option 1

Determine the phase transition points (Rajiv *et al*., 2008) using one of the transition points buttons (Figure 5), or manually add the phase transition points using the context menu (by right clicking) in the '**Powder 3D Parametric Graphics**' interface (Figure 5).

The determined (or assigned) phase transition points will be displayed as lines (colored blue) in the '**Sequential refinements**' tab of the '**Powder 3D Parametric Graphics**' interface (Figure 5). These lines can be moved manually to fine tune or redefine the phase transition points' positions.

Once the user is sure about the number of phase transitions, the '**Save Transition points**' button must be clicked (Figure 5). This will mask the overlapped regions between the phases with hatches (Figure 6) and cluster the powder patterns that correspond to pure phases and those that belong to the overlapped parts (mixtures).

(Later, these two groups can be independently subjected to sequential/parametric refinements).

Option 2

In case of non-reconstructive transitions[3] the '**No transition points (whole range)'** option (Figures 4 & 5) must be chosen. Selecting this option will automatically assign and the number of reconstructive transitions to zero (No hatches will be displayed unlike the previous case).

[3] Contains no 'visible' phase transitions (occurs in some time-dependent experiments).

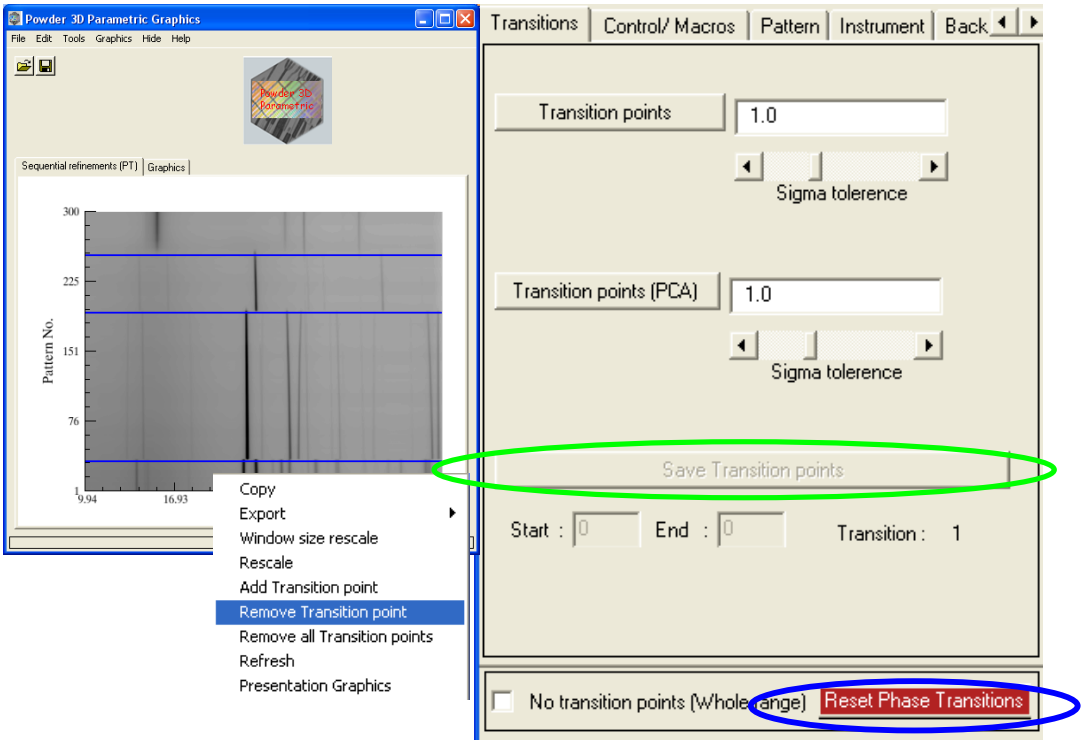


Figure 5 The transition points found are shown as lines (colored blue).The context menu (right click) options to modify the number of transition points is shown. The '**Transition points**' buttons are shown. '**Save Transition points**' button is highlighted in green. The '**Reset Phase Transitions**' button is highlighted with a circle colored blue.

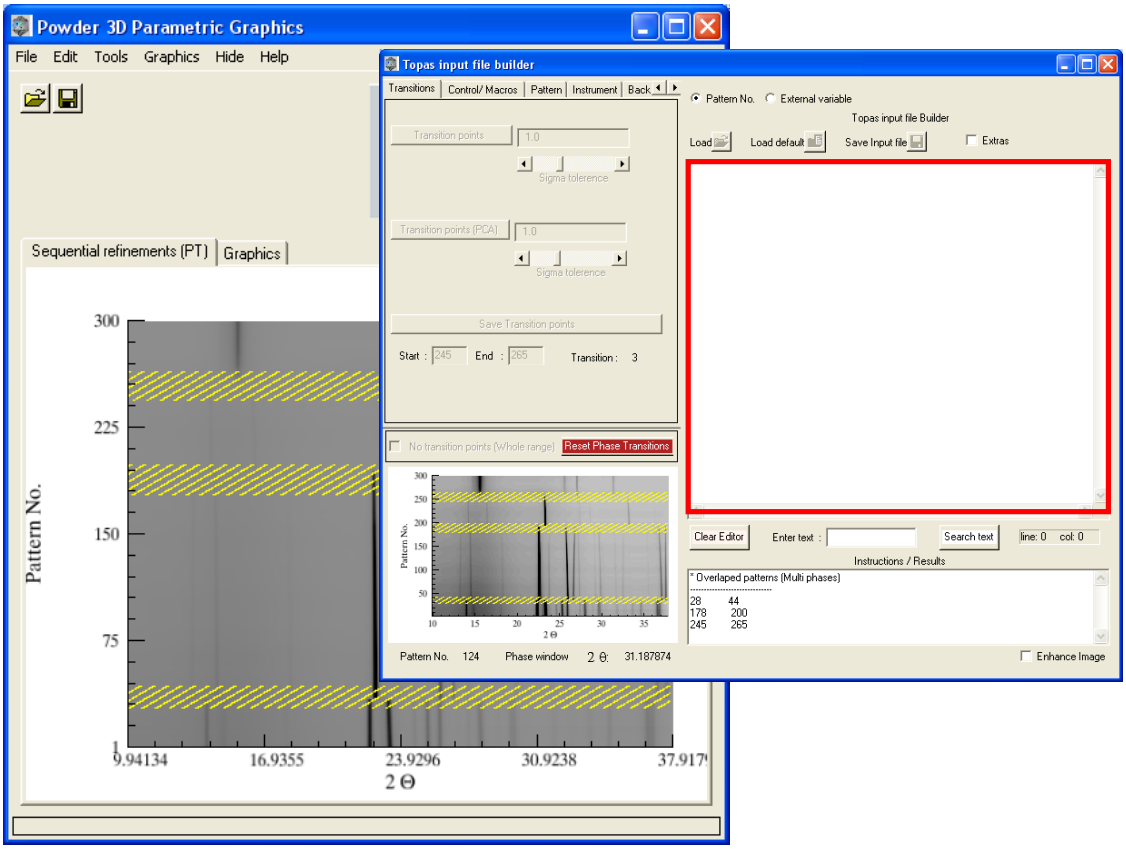


Figure 6 The hatches (yellow) that defines the overlapped region between two phases. The input file editor is at the right (red framed).

The number of phase transition points determined can be reset and modified at any time (for both Option 1 and Option 2) by prompting the '**Reset Phase Transitions**' button (Figure 5).

6. Test refinement

Before starting sequential/parametric refinements the program demands that the user performs a refinement of a single powder pattern[4] (will be called 'test' refinement). The input file used in the test refinement will be used by the program for making the input files required for sequential and parametric refinements.

In section 7.1 and 7.2, the preparation of 'test' input file and the procedure for performing 'test' WPPR (LeBail fit) of the sample dataset (α-Ag_3VO_4) (Dinnebier *et al.*, 2007) will be demonstrated.

[4] Usually the pattern measured at lowest time step or the pattern from where the user wishes to begin the refinements.

7.1. Input file preparation

Before reading the following instructions, read the syntax restrictions presented in section 11.

1 The input file required for the test refinement can be built using some widget tools provided in the tabs (with the titles: **Control/Macros**, **pattern**, **background**, **Range1, Overlapped1,** etc) in the '**Topas input file builder'** (Figure 7). These widget tools tag some 'Topas' keywords and macros that constitutes the input file statements required for the refinement.

2 In Figure 7, one of the tabs (the '**Control/Macros**' tab) is highlighted. The variables associated with the '**Control/Macros**' information (e.g. *'iter'*, *'convolution_step'*, *'r_wp'*, etc.) can be loaded to the text editor by prompting the respective gadgets.

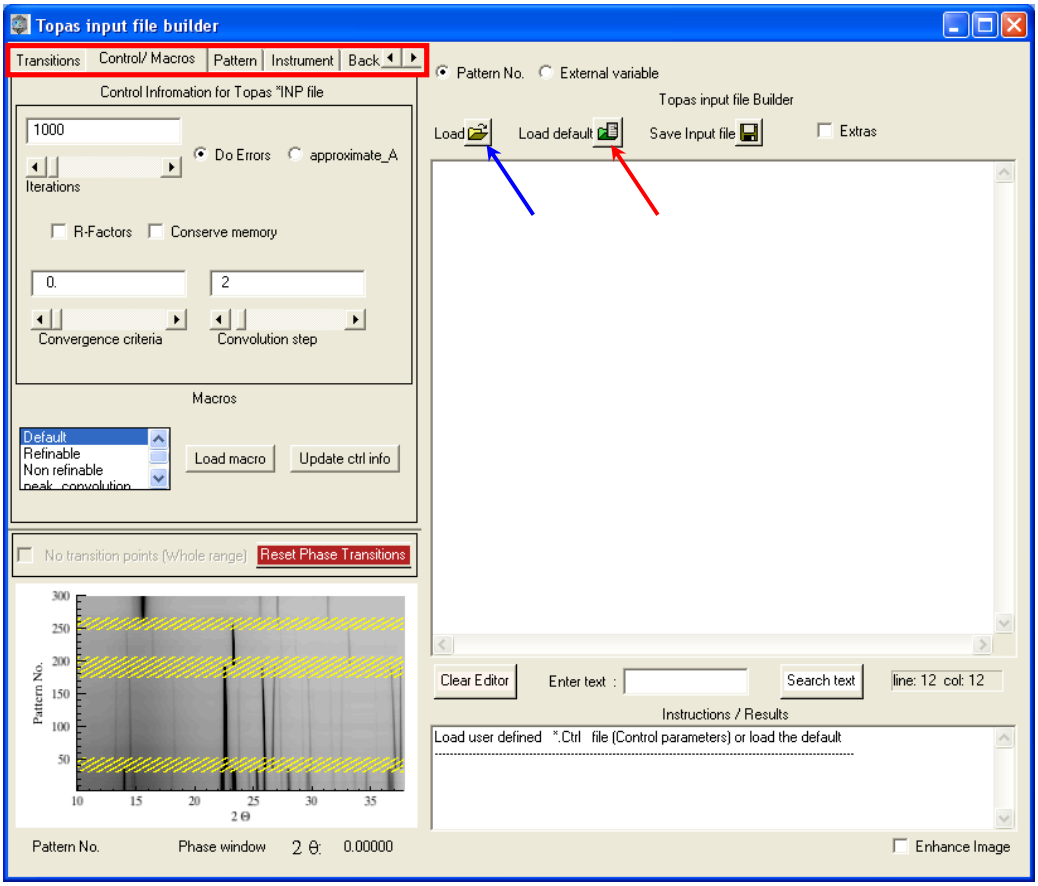


Figure 7 The **Control/Macro** tab is highlighted. The control parameters (e.g. '*iter*', '*convolution_step*', '*r_wp*', etc.) can be loaded to the text editor. The '**load'** and '**load Default**' buttons are pointed by arrows. The **Control/Macros**, **pattern**, **background** tabs are highlighted by a red rectangle.

3 In a similar way the Topas keywords associated with the powder pattern (e.g., '*start_X*', '*finish_X*' etc.) and the keywords related to a structural phase (e.g., '*al*', '*weight_percent*') can be loaded from the **Pattern** and **Range**[5] tabs (Figure 8) of '**Topas input file builder'** respectively.

4 The approximate starting values for some variables (e.g., *convolution_step*, *start_X*, *finish_X*, *bkg* etc.) can be automatically determined using the '**Load default**' button (Figure 7).

5 Note: The user need not use the '*Out*' macro and the '*out*' statement in the 'test' input file. The program will automatically read the refinable variables from the input file and outputs them, the '*Out*' and '*out*' commands (if present in the input file) will be overwritten.

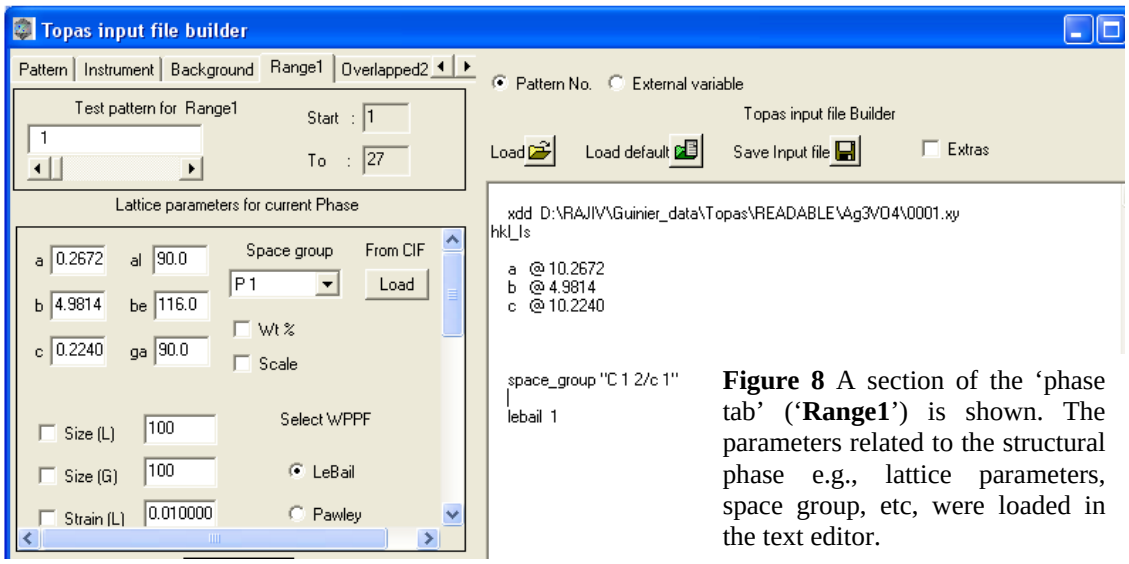


Figure 8 A section of the 'phase tab' ('**Range1**') is shown. The parameters related to the structural phase e.g., lattice parameters, space group, etc, were loaded in the text editor.

6 Alternatively, all the required keywords can also be typed manually by the user in the text editor!

7 The required keywords can also be placed into the text editor by loading an ASCII file (with extensions *'*.ctrl' '*.patt'* or *'*.phs*', etc.) containing the set of required keywords. These files have to be manually prepared beforehand by the user and must be loaded to the text editor by using '**Load**' button (Figure 7).

[5] The names of the '**Range'** tabs (e.g, Range1, Overlapped1, etc) can be modified (e.g., to Room_ Temperature, High_Temperature_1 etc.) in the **Edit** menu **WPPF Options**, **Range name.**

8 Note that the steps 1 to 4 are meant for the users who want to build the input file from the beginning. If the input file required for the test refinement is already at hand, then the '**Range**' part (or phase part) of the prepared file must be copied to the editor of the corresponding '**Range**' tab (e.g., **'Range1'** in Figure 8). All the other 'common' keywords (e.g., *start_X* or *LP_factor*) must be copied to the text editor that corresponds to one of the other tabs (e.g., the '**Background'** tab).

9 The '**Range**'(or phase) part of the file starts from the keyword '*str*' or the macro '*STR'* or '*hkl_Is*' to the *end of file*. (The phase and the common part of the Topas input file are distinguished in Figure 9). (Use the file "alpha_Ag3VO4.INP" for the present demonstration).

```
xdd D:\Guinier_data\Topas\READABLE\Ag3VO4\0001.xy
r_wp 2.441967259 r_wp_dash 3.453943704

iters 1000
        bkg @ 0 0 0 0 0 0 0 0 0 0 0 0
        Zero_Error(@, 0)
        LP_Factor( 90)
        convolution_step 2
        Rp 100
        Rs 100
        lam
                ymin_on_ymax  0.0001
                la  1 lo  1.12058 lh  0.00001
        x_calculation_step 0.02

        hkl_Is
                lebail  1
                CS_G(@, 200)
                Strain_L(@, 0.1)
                r_bragg  0
                phase_name hkl_Phase
                space_group C2/c
                MVW( 0, 0, 0)
                be @ 116
                a @ 10.2672
                b @ 4.9814
                c @ 10.224
```

Figure 9 The input file for performing LeBail fit for α-Ag_3VO_4 data is presented. The 'common' part of the input file is typed in red. The Range or phase part of the file is typed in blue.

7.2. Starting the 'Test' refinement

1 Once the input file is ready, the user must select the name of the phase to be refined from the list provided in the '**Sequential Refinement**' tab (Figures 10 & 11). (For the present demonstration, the user must select the phase '**Range1'** which correspond to α-Ag_3VO_4).

2 The final Topas '*inp*' file (all the comments gathered from various tabs) will be loaded in the text editor in the '**Sequential Refinement**' tab as soon as the phase to be refined (**Range1**) is selected. This will also make the '**Sequential Refine**' button active (depending upon the whether the test refinement of the selected phase is successfully finished).

3 The loaded input file cannot be edited in this tab; the keywords/commands can be edited only in their editors in their respective common tabs and can be reloaded in the editor in the '**Sequential Refinement**' tab.

4 The test refinement can be started by prompting the **'Test Refine'** button (Figure 10).

5 After performing the test refinement, its results will be displayed in a separate window (inner picture in Figure 10). This GUI will contain the convergence plot, the Topas *.*out* file, the list of refined variables of the selected ('**Range1')** phase.

6 The user can decide whether to keep the results of the test refinement ('**Keep Parameters**' button) or to redo the test refinement ('**Ignore**' button) based on the convergence of the refinement.

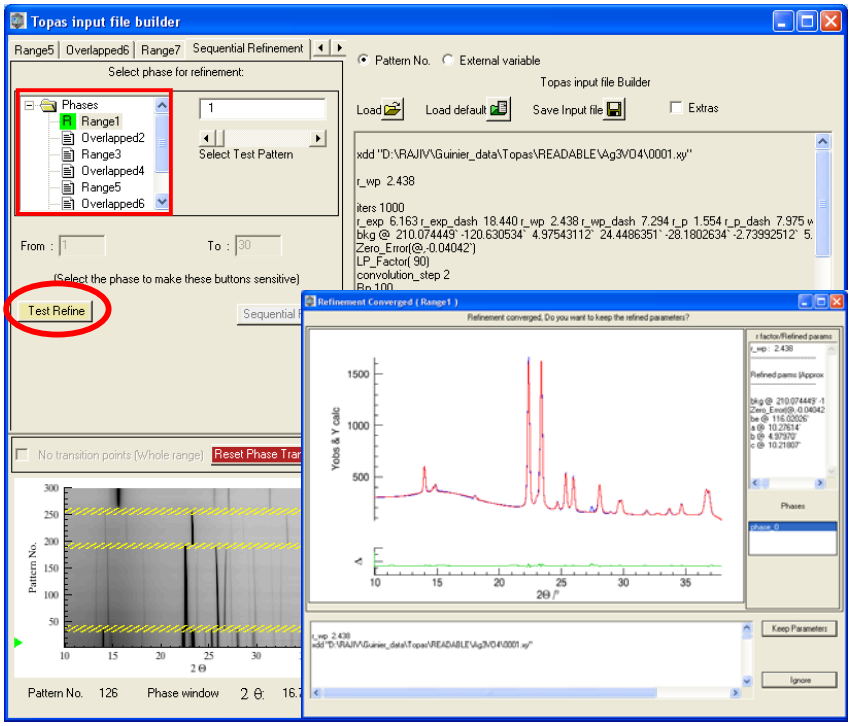


Figure 10 '**Sequential Refinement**' tab is shown. The list of phases and the overlapped parts of Ag_3VO_4 with their default names ('**Range1'**, '**Overlapped2'**, etc) are highlighted with the red frame. The '**Test refine**' button is circled in red.

7 Prompting the '**Keep Parameters**' button will change the label of the refined phase to the letter 'R' and its color to green (Figures 10 and 11) indicating that the 'test.*out*' file of test refinement has been stored and the stored file will be used for the sequential refinements.

Note: If the user wants to redo a test refinement at this point with modified variables (after clicking '**Keep Parameters**', after the letter 'R' changes to green) then the corresponding phase must be reset by right clicking and resetting the phase (Figure 11); Only after resetting, the variables can be modified in their respective editors.

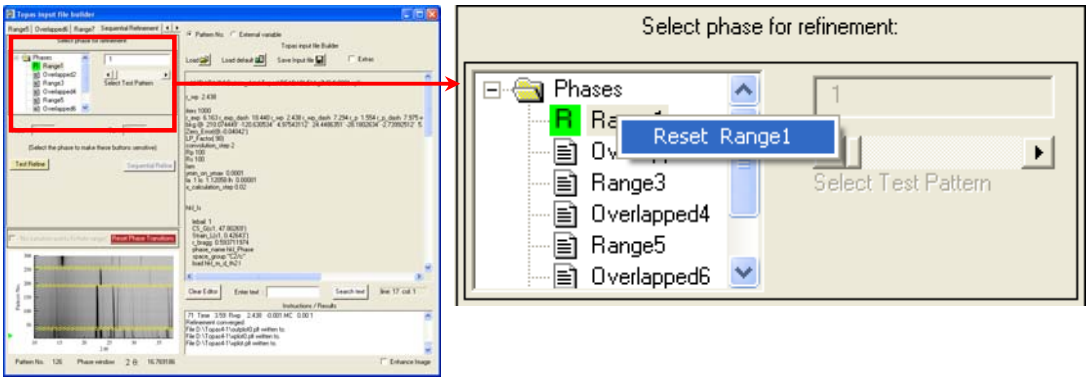


Figure 11 Resetting a '**Range'** (a refined phase) for a new test refinement.

8. Sequential refinements

1 For the current demonstration, 30 powder patterns that belong to α-Ag_3VO_4 ('**Range1'**) can be sequentially refined. (The number of patterns that belong to **Range1** can be changed by modifying the widths of the 'hatches' -Figure 6).

2 The sequential refinements can be started by prompting the button '**Sequential Refine**' (Figure 10). Before proceeding further, the names of the sequentially refinable variables must be checked in a 'confirmation' GUI (Figure 12). This interface also provides an option to save the refined variables and errors as an ASCII file with the name <phase_name>_err.txt"), where 'phase_name' is the name of the first phase of the compound read from the input file (default filename is phase0_err.txt).

3 The other options include the direction of sequential refinements (forward or reverse '**F'** and '**R'** in Figure 12).

4 Once the variable names are confirmed, the refinements can be started by prompting '**Start refinement**' button (Figure 12). The program will prepare and execute all the input files needed for the following refinements. After the completion of all the refinements, a spread sheet containing the refined parameters and several options for parameterization will appear (Figure 13).

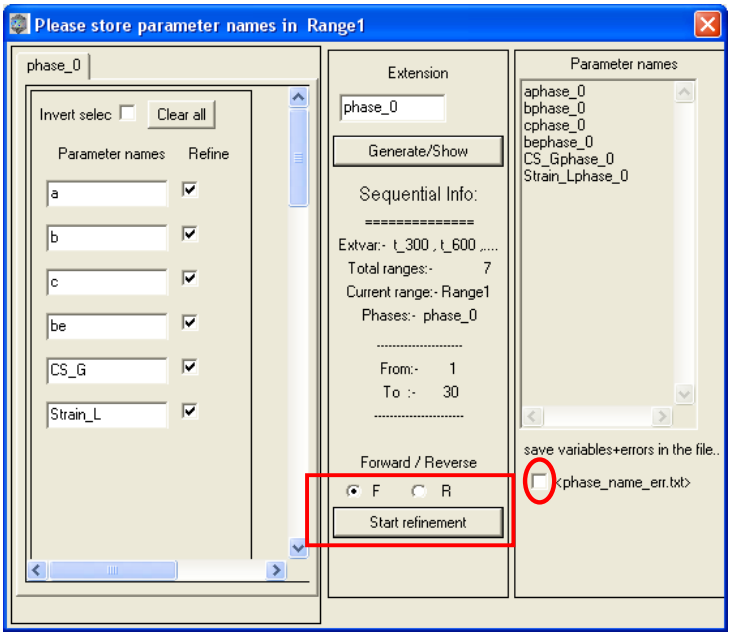


Figure 12 Sequential refinement confirmation window with options to store refined variables.

5 The refined parameters will be sorted according to their **Ranges** and phases in the '**Parameters spread sheet**' (Figure 13). Their variation with respect to external variables can be studied, by plotting them simultaneously against the external variable. The program provides options to plot the weighted residuals obtained in the sequential refinements.

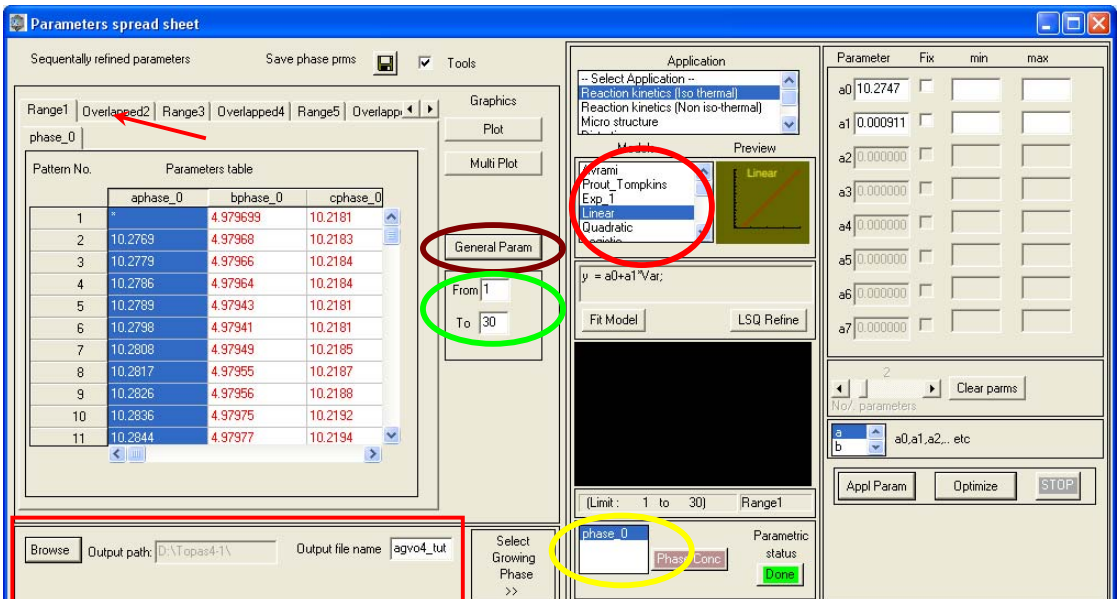


Figure 13 The '**parameters spread sheet'** that pops up after the sequential refinements is shown. The refined parameters are sorted according to their phases and ranges (pointed by arrow).

6 The ‘least squares’ tool supplied by the program can also be used to initialize the model parameters of the variables to be used in parametric refinement.

9. Parametric refinements

The program provides the facility to parameterize all the variables associated with the test input file (Figure 13) with a model listed by the program or by any other user defined equation.

Depending on the requirement of the user, the variables can be parameterized in two modes; the ‘General parameterization’ and the ‘Application parameterization’ (‘**General Param**’ and ‘**Appl Param**’ buttons in Figure 13). In both modes, the user can either,

- Prepare the input file associated with the parameterization of a variable (or)
- Prepare the input file and also carryout the refinements in the Topas launch mode.

9.1. General parameterization

The ‘General parameterization’ mode depends completely on the need of the user. Any variable associated with the test input file can be parameterized with any model.

The procedure to prepare the input file associated with the parameterization of lattice parameter ‘a’ of α- Ag_3VO_4 with a linear model is as described below:

1 The phase (range) to be refined must be selected first (Figure 11). (For this demonstration it is ‘**Range1**’:α- Ag_3VO_4).

2 The ‘**.out*’ file that will be generated in the parametric refinement must be given a valid name (without extension). Its location must be selected using the ‘**Browse**’ button (Figure 13, red framed). (Recommended *.out file location is the Topas working directory).

3 The correct range of patterns that will be used in parametric refinement must be entered in their respective fields. (‘**From’** and ‘**To’** circled in green in Figure 13). (For this demonstration it is 1 to 30 for ‘**Range1**’or α- Ag_3VO_4).

4 An appropriate model of the variable must be selected from the list ‘**Models**’ (Figure 13 circled red). (Linear model is chosen for the present case).

5 To start the refinements, the ‘**General Param**’ (Figure 13) button must be prompted first, which will result in the ‘**Parameterization Option**’ window. (Figure 14).

6 The '**Parameterization Option**' GUI provides the following options which must be set before proceeding further:

1 The variable to be parameterized (Figure 14 circled red) must be selected. (Lattice parameter 'a' of α- Ag_3VO_4 must be chosen for the present case).

2 The refinement flag status ('**Fix**') of all the other variables must be set.

3 The scope (**Local/Global**) of all the other refinable variables must be set (default is '**Local**').

4 The name of the text file in which the parametrically refined variables will be stored must be given. The default names are 'Pars0, Pars1,..,etc. (where 0,1,...,etc are the indices of the phases- Figure 14).

5 One of the options '**Make file**' or '**Refine**' must be selected. '**Make file**': For only creating the parametric input file (no refinements) **'Refine**': For creating the file and also for doing refinements (with graphics)

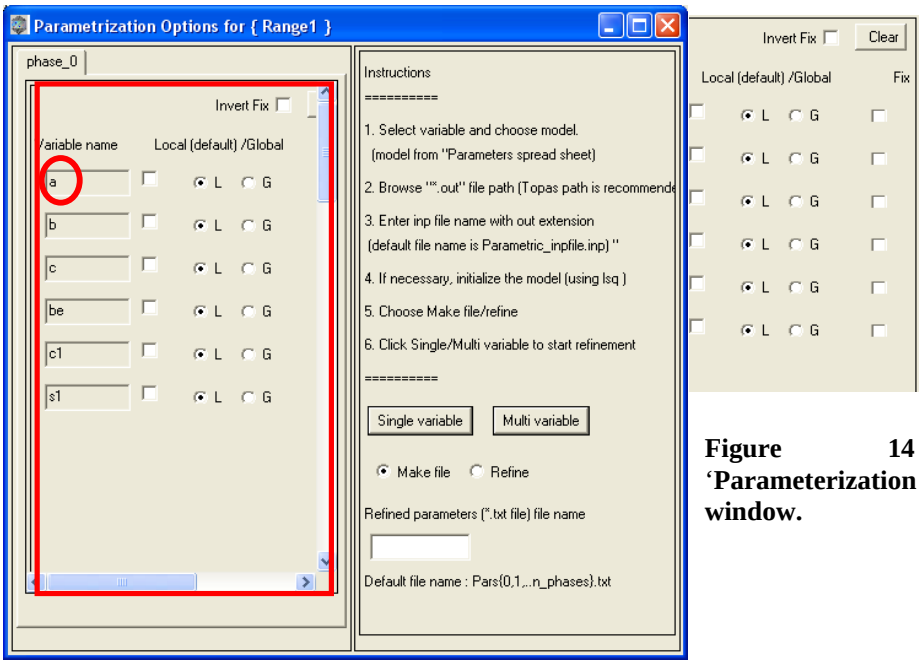


Figure 14 The 'Parameterization Options' window.

7 After setting all the options in the '**Parameterization Option**' GUI, the parametric refinements can be started by clicking the '**Single variable'** button. The results of the refinement (if '**Refine**' option was selected) will appear as in Figure 15.

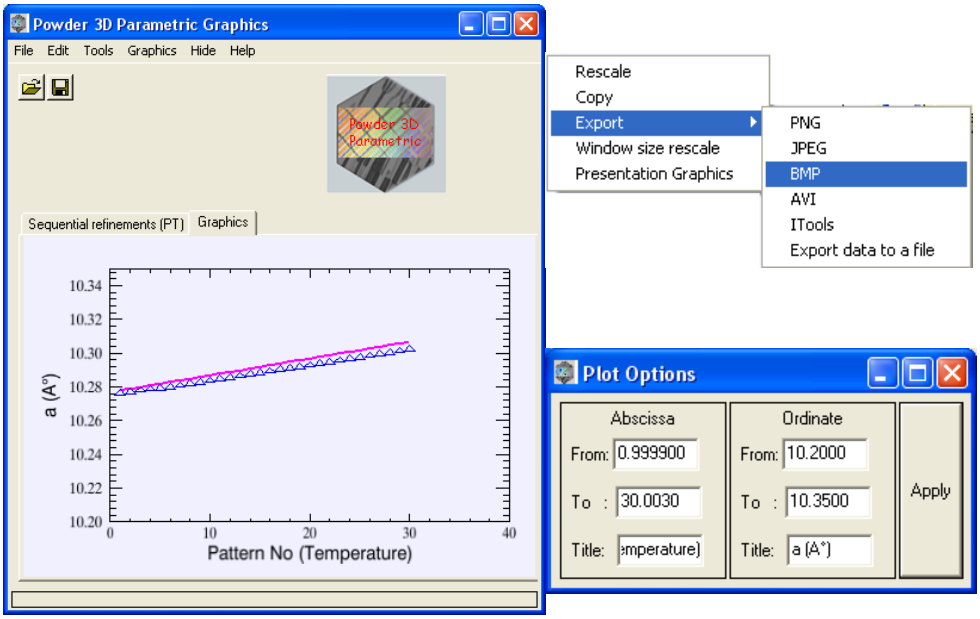


Figure 15 Parametric refinement results (pink line) are compared with sequential refinements (blue triangles). Lattice parameter 'a' of α-Ag_3VO_4 is parameterized by a linear model. The options for plotting graphics and storing the graphics are also shown.

9.2 Application parameterization

The program provides some predefined application modules such as, '**Reaction kinetics**', '**Zero error**' etc. These modules have unique scientific purposes. The procedure to execute one of these applications (reaction kinetics) is demonstrated in this section.

9.2.1. Reaction kinetics

The goal of the '**Reaction kinetics**' module of the program is to use the parametric Rietveld refinement method to extract the reaction rate constant and reaction order[6] of a specimen from its time dependent XRPD datasets. For the demonstration purpose, the time dependent dataset of one of the mixtures of Copper Phthalocyanine (CuPC) (Müller, *et al*, 2009) will be used in this tutorial.

To get the Avrami parameters ('n' and 'k') from parametric Rietveld refinement, the following instructions must be followed. (The set of instructions provided in sections 4-5 loading data, entering basic experimental information etc, must be done before following the instructions given below).

[6] The Avrami parameters 'n' and 'k' (Müller 2009)

1 The sample dataset used in this illustration are measured isothermally (in dependence on time). Therefore, they belong to 'No transition points Option 2' explained in section 6. The '**No transition points (Whole range)'** checkbox in '**Topas input file builder'** must be selected (Figure 4)**.**

2 The input instructions to do one test refinement must be placed into the editor in one of the following ways explained in 7.1 (depending upon the availability of the input file).

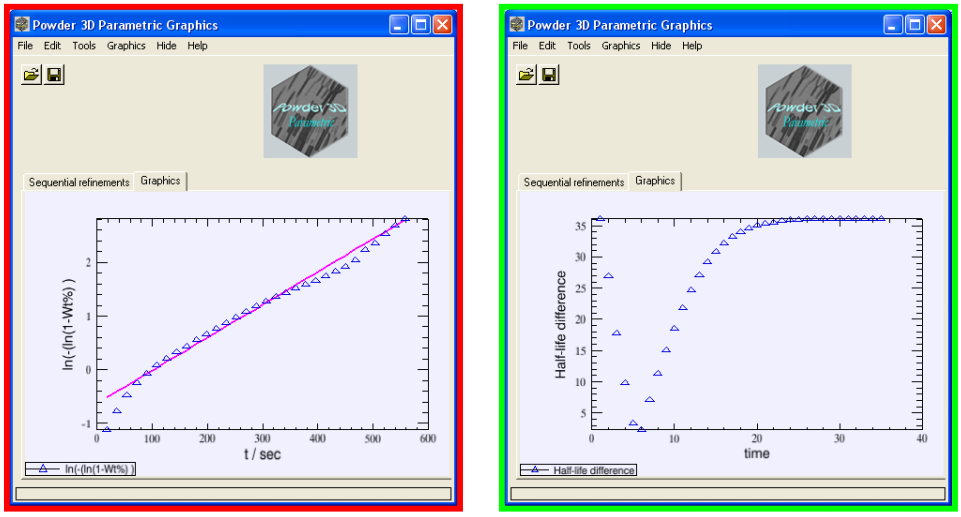
3 The test refinement must be performed as explained in the section 7.2. (For the kinetic analysis it is necessary to set the parameters '*scale*' and/or '*weight_percent*' refinable in the test refinement).

4 After the test refinement, the sequential refinements can be started as explained in the section 8.

9.2.2. Parametric refinements

1 To proceed with parametric refinement (in Reaction kinetics), the name of the application must be selected first ('**Application**' in Figure 13). Depending upon the experiment type, '**Reaction kinetics (Iso thermal)**' (the present case), or '**Reaction kinetics (Non iso-thermal)**' must be selected.

2 When an application is selected, the corresponding '**Options window**' will popup (Figure 16). The '**Reaction kinetics options window**' has two widget buttons '**lsq**' and '**Half-life**'. These two buttons are used for initializing the Avrami parameter 'k'. The '**lsq**' performs a general least squares fit to the linearized weight fraction of the growing phase. The '**Half-life**' button finds the half life of decay of the vanishing phase, from which the approximate value of 'k' is estimated. Before using these two options, make sure that the growing phase is correctly selected ('**Select Growing Phase'** in Figure 13). Note: Physically meaningful starting value of the parameter 'k' is required for the successful parametric refinement of datasets with higher fluctuation.



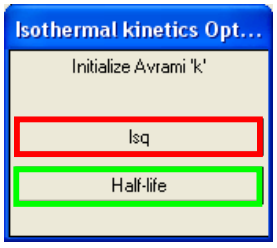


Figure 16 The starting values for the Avrami parameter 'k' is found by making a linear fit to the growing weight fraction (framed in red). The parameter 'k' is also initialized with the half-life of the decaying phase. The standard deviations of the half-life of decay of the decaying phase are plotted in this figure (framed green).

3 The growing phase must be selected now (Figure 13 circled in yellow). The default and best model for this analysis is '**Avrami**', which must be selected for the list (Figure 13).

4 '**Appl Param**' button shown in Figure 13 (NOT '**General Param**'!) must be prompted for starting the parametric refinement. A confirmation window as in Figure 14 will popup. The variable to be parameterized in this application is the 'scale factor'. This variable will be already selected in the confirmation window!

5 Now follow the instructions in step 6 in section 9.1 (from steps 1 to 6) to start the refinement or to make the input file.

6 The results of parametric refinement (weight fractions of the growing phase) are shown in Figure 17. The Avrami parameters obtained from parametric refinement are shown in Figure 17.

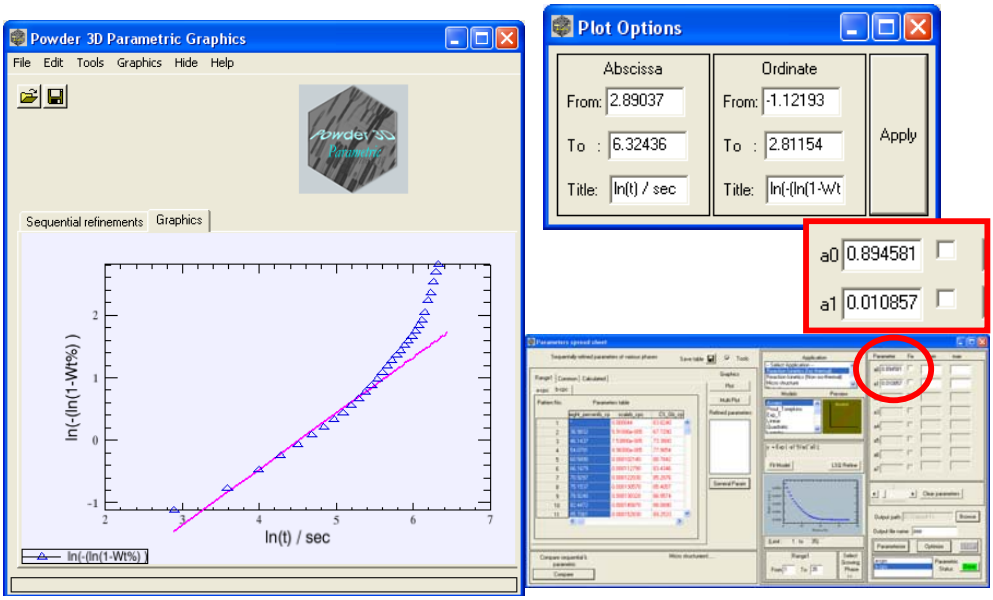


Figure 17 The results of parametric refinements (pink) associated with the '**Reaction kinetics**' module are shown. The refined parameters are highlighted with the red circle. The default coefficient names are a0, a1,., etc The coefficients a0 and a1 represent the Avrami parameters 'n' and 'k' respectively.

7 The weighted residuals (*r_wp*) obtained in the parametric refinements (Figure 18) are plotted. Their shape can be used to select the optimal pattern range (time/temperature range) and redo parametric refinements.

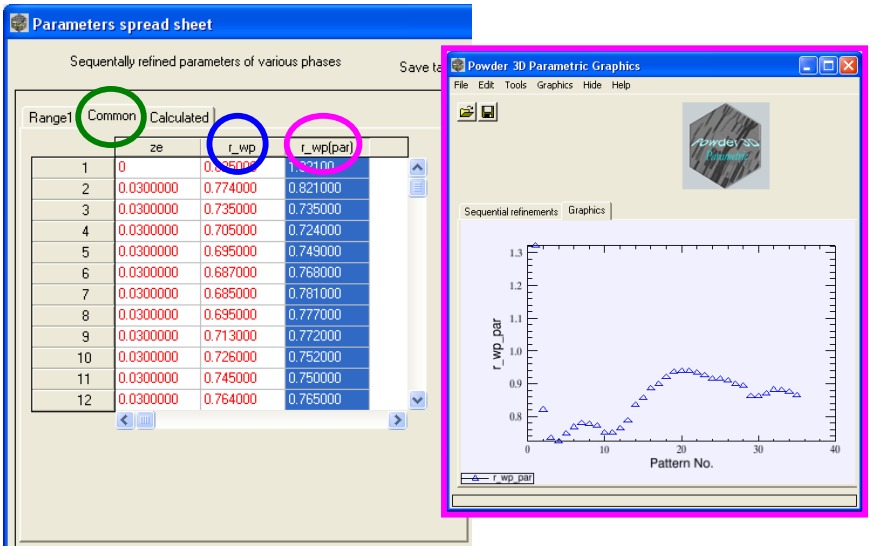


Figure 18 The weighted residuals (*r_wp*) of sequential/parametric refinements are shown in the **Common** parameters tab (circled in green). The weighted residuals of parametric refinement (pink circled) are plotted on the right picture.

10. More options

10.1. User defined equation

Instead of Avrami model, general user equations can also be used to parameterize the scale factor.

1 The model name '**User_defined**' has to be selected from the model list '**Models**' (in Figure 19).

2 The model equation must be entered in the field provided (Figure 19). (Note that the syntax for these equations must be identical to that of the Topas macro language).

3 Similar to Topas macro language, semicolon (;) must be used as the termination statement. The default model coefficient names are a0, a1, a2...etc and the default name to represent the independent variable 'x' is '**Var**'. (For example use the equation '**y = a0+a1*Var;**' to parameterize the scale factor with a linear model).

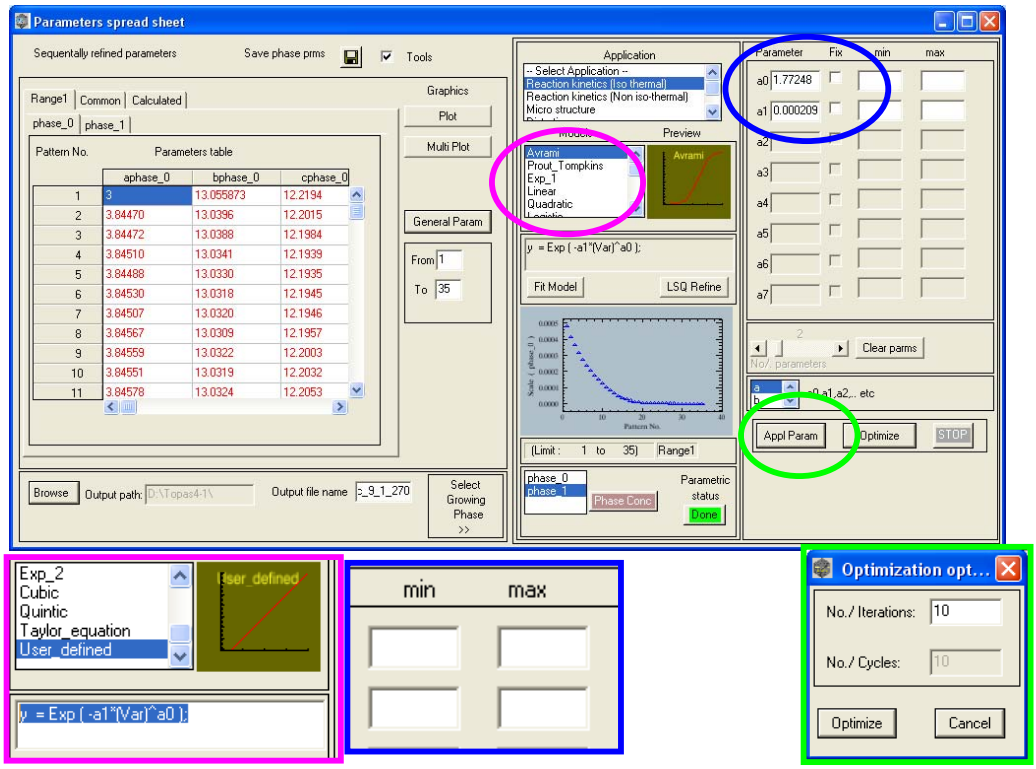


Figure 19 The option '**User_defined**' must be selected from the list first (circled in pink). The user defined equation of the parameter (in the present case scale factor) can be introduced in the text box (framed in pink). The '**min**' and '**max**' text boxes (blue framed) can be used to limit the refinable parameters. The options for global optimization are also shown ('**Optimization options**' GUI).

10.2. 'Min' –'Max' and global optimization of variables

The '**min**' and '**max**' fields (highlighted in blue frame' in Figure 19) are identical to the Topas keywords '*min*' and '*max*' used for constraining the variables. The '**Optimize**' button (high lighted in green frame in Figure 19) can be used to perform global optimization of the refined variables. To do global optimization of the variables follow the two steps below.

1 The '**Optimize**' button must be prompted. The number of iterations for global optimization must be entered in a new interface ('**Optimization options**') that appears (the default number of iterations is 10).

2 The '**Optimize**' button in the '**Optimization options**' window must be clicked to start the global optimization. Part of the input file used in global optimization is shown in Figure 20.

```
iters          10
out_prm_vals_on_convergence Filename.txt

continue_after_convergence
chi2_convergence_criteria 0.0001
prm  a1  0.00717` val_on_continue = Val + Rand ( 0.000000, 0.000000  );
```

Figure 20 A part of the Topas input file used in the 'Global optimization' is shown. The variable 'a1' is optimized here.

11. Some exceptions in the input file syntax

While creating the input file for the test refinement, the user must follow certain 'syntax' restrictions imposed by 'Powder 3D parametric'. These restrictions are made in order to improve the success of some of the 'keyword-search' algorithms used in the program. The syntax restrictions are listed and explained in detail in this section.

1 While entering the Topas statements 'str', 'hkl_Is' and the macro 'STR' in the text editor, make sure that all the statements are entered in succeeding lines and not in the same line (see Figure 21 for explanation).

```
str
CS_G(, 25.10749_0.33412)
Strain_L(, 2.62461_0.07480)
prm a0 =11.5;
.
.
.
str   CS_G(, 25.10749_0.33412)
Strain_L(, 2.62461_0.07480) prm a0 =11.5;
.
.
.
r_bragg  0.354
scale    0.2434
phase_name "Struct"
cell_mass  919

r_bragg  0.354  scale  0.2434 phase_name "Struct" cell_mass  919
```

Figure 21 The statements recognized by Topas launch mode and 'Powder 3D Parametric' are colored blue. The statements recognized only by Topas launch mode kernel are colored red.

Although Topas can read and execute multiple statements entered in a single line (shown in Figure 21 as red letters), 'Powder 3D parametric' can recognize only the statements that are entered in consecutive lines (letters colored blue in Figure 22).

2 In the test input file, the Topas-keyword parameters to be refined must be marked with the '@' symbol (as explained in the Topas technical reference, Coelho, 2007) and the variables associated with the macros to be refined must be given valid names and not '@'. For example, to refine the lattice parameter '*a*', the statement '*a @ 5.5*' must be used. To refine a macro variable e.g., the Gaussian component of crystallite size, the macro must be entered as *CS_G (cs1, 25.108)* (*cs1* is the variable name) and not as *CS_G (@ , 25.108).*

3 If the user wants to redo sequential refinements, the input files <datafilenames>.inp' etc (default file names) created by the program in the previous set of refinements must be deleted before restarting sequential refinements. (They will be saved in the location e.g., D:\Topas4-1 or C:\Topas3, depending upon where the 'tc.exe' file is located).

12. The input file maker

12.1 Introduction

The ‘input file maker’ tool is meant for doing quick sequential refinements and for the preparation of the input files required for parametric refinements. This module requires an already prepared ‘test’ input file is available at hand.

The procedures for doing sequential refinements and that for preparing the parametric input file using the ‘inp maker’ module are relatively simple as compared the previous sections (the test file creation, test refinement sequential refinements, etc). Using this module the input file associated with the parameterization of a variable can be prepared without performing any sequential refinements.

The ‘input file maker’ module doesn’t support any graphics and can be executed directly from the ‘**Tools**’ menu (Figure 22).

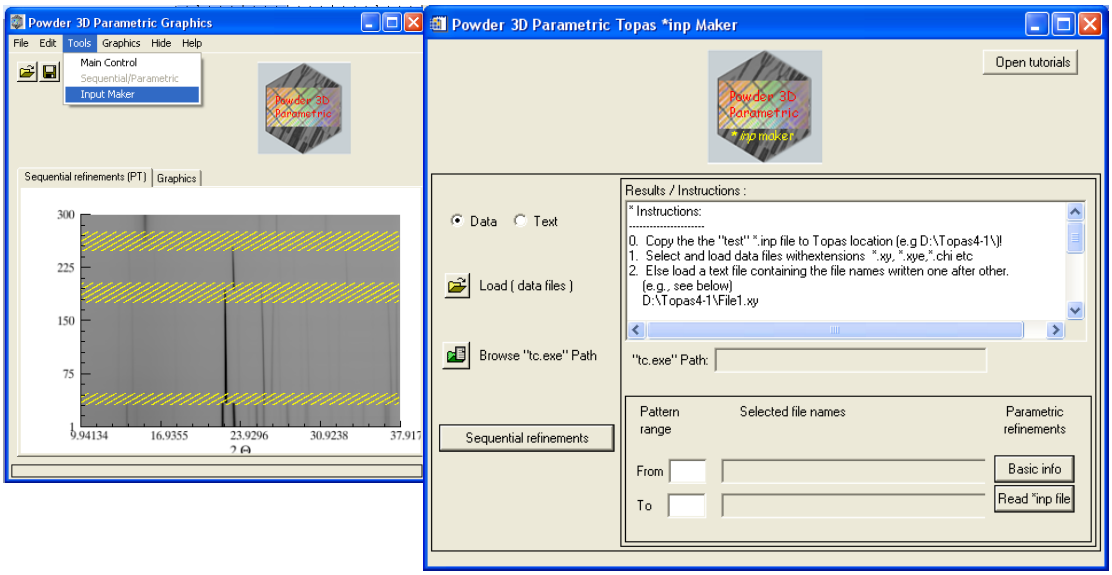


Figure 22 The execution of the ‘Inp maker’ module from the **‘Tools’** menu.

12.2 Sequential refinements

For starting the sequential refinements, the ‘**Load**’ and ‘**Browse Topas path**’ (Figure 23) buttons must be prompted to load the data location and the Topas path (similar to section 4.1). ‘**Sequential refinements**’ button (Figure 23) must be used to load the manually prepared ‘test’ input file.

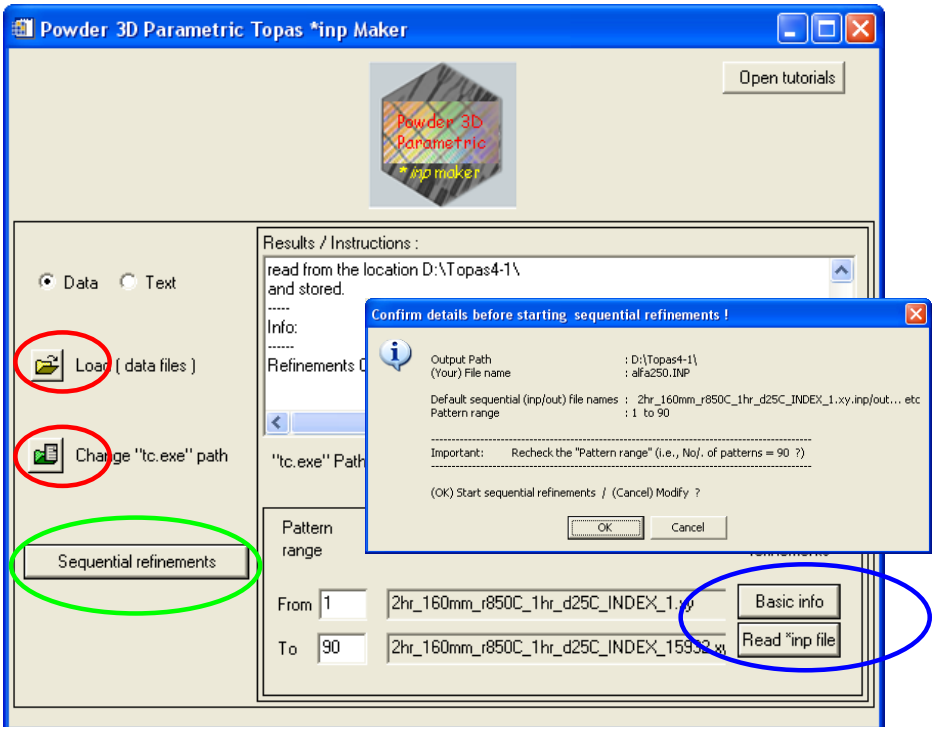


Figure 23 Sequential refinements using 'Inp maker' module

Some details about the filename location etc, will be displayed (Figure 23) before starting the refinements. After confirming these details, the program will start preparing the 'sequential' input files and execute them.

The '*OUT*' macro of Topas can be made use in the 'test' input file in such a way that the sequentially refined parameters are stored in the form of a table in a '*.txt' file. After executing all the input files sequentially, the refined parameters table can be easily analyzed using the commercial programs that offer facilities to analyze data using worksheets (Microsoft Excel, 2010).

12.2 Parametric refinements

The instructions to prepare the input file associated with parametric refinement are as given below.

1 '**Basic info**' button (Figure 23 circled blue) must be prompted to load the basic experimental information (similar to section 5).

2 '**Read *.inp file**' button (Figure 23 circled blue) must be prompted to load the prepared 'test' input file.

3 After selecting the ‘test’ input file an interface similar to the ‘**Parameterization Options**’ GUI (Figure 14) will pop up (Figure 24).

4 The parametric refinement options must be set as explained in section 9.1 step 6.

5 Input file required for the parametric refinement of any variable (associated with the loaded input file) can be created using the instructions provided in section 9.1.

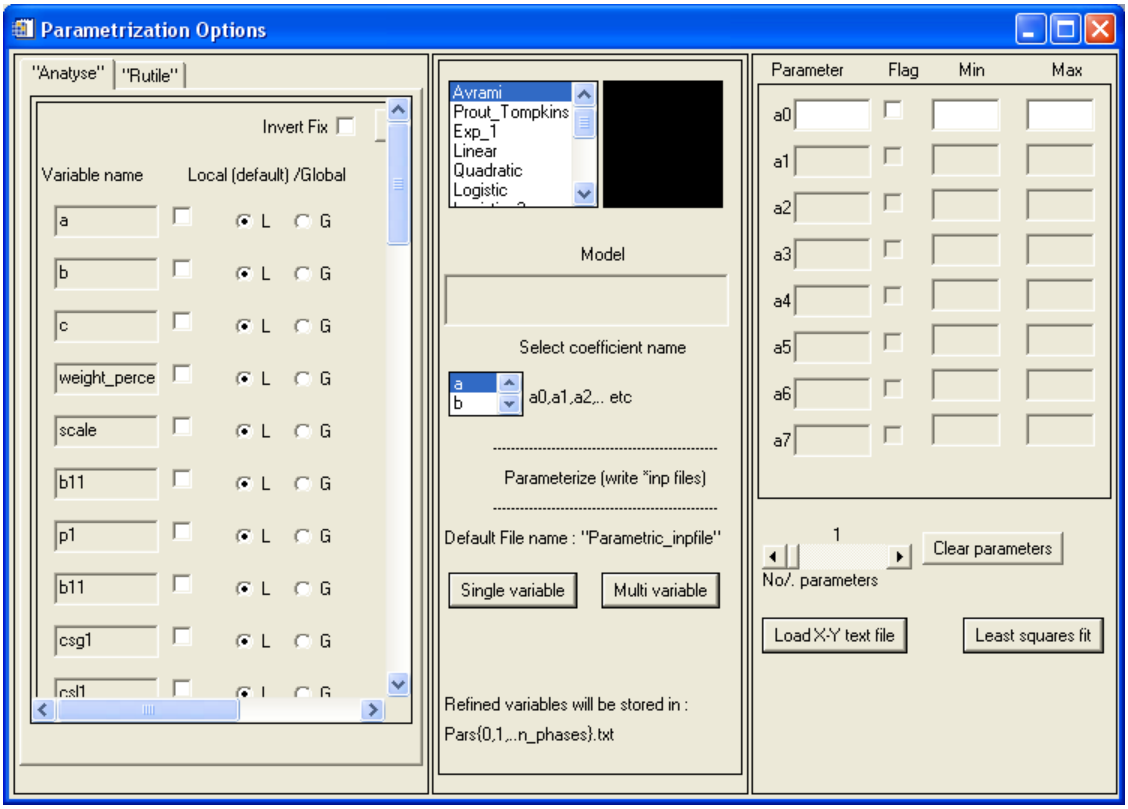


Figure 24 ‘Parametric refinement options’ window of the ‘inp maker’

6 The created parametric input file has a default name ‘Parametric_inpfile.inp’ will be stored in the Topas working directory. This file can be executed in the Topas GUI mode.

References

Chung, D. D. L., Haven, P.W.De., Arnold, H., and Ghosh, D., 1993, X-ray diffraction at elevated temperatures, Vch, New York.

Coelho, A. A., 2007, TOPAS, v4.0, *Bruker AXS*.

Dinnebier, R. E., Kowalevsky, A., Reichert, H., and Jansen, M., 2007, Polymorphism of Ag_3VO_4, *Z. Krist* **222:**420–426.

ITT, 2009, IDL version 6.4, in: *ITT visual information solutions*.

Müller, M., Dinnebier, R. E., Jansen, M., Wiedemann, S., and Plüg, C., 2009, Kinetic analysis of the phase transformation from α to β-copper phthalocyanine: A case study for sequential and parametric Rietveld refinements, *Powder Diffraction* **24**(3)**:**191-199.

Rajiv, P., Dinnebier, R. E., and Jansen, M., 2009, Automatic determination of phase transition points in in situ X-ray powder diffraction experiments, *Powder Diffraction* **24**(1)**:**8-16.

Rajiv, P., Dinnebier, R. E., and Jansen, M., 2010, "Powder 3D Parametric"- A program for automated sequential and parametric Rietveld refinement using Topas, *Materials Science Forum* **651:**97-104.

Stinton, G. W., and Evans, J. S. O., 2007, Parametric Rietveld refinement, *J. Appl. Crystallogr* **40:**87-95.

Written by Rajiv Paneerselvam 9.9.2011, Stuttgart.