

# Introduction to the program *FullProf*

## How to use the self-training examples

### Short Rietveld Course– May 2001 Atlanta

This brief introduction to *FullProf* is part of the future manual that will be available shortly. The detailed description of the input control file is described in the document: **App\_fp2k\_man.pdf** distributed with the program via the Internet as described below.

**The mathematical section is not complete**

#### *Availability of FullProf*

The old versions of *FullProf*, written in Fortran 77 and running in different platforms, are in the directory *pub/divers/fullp* of the anonymous **ftp-area** of the server *charybde.saclay.cea.fr*. Users interested in creating their own subroutines to link with the FULLP-library are asked to read the file **fpreadme** in the above-mentioned **ftp-area**. To access this area from the Internet, one has to type in the local host the following command:

- LocalPrompt> ftp charybde.saclay.cea.fr <cr>

Answer with the word: *anonymous*, to the Login request and password. Within the ftp prompt, do:

From the local host:

ftp>cd pub/divers/fullp	! Go to FullProf area (Multi-platform)
ftp>get fpreadme	! Obtain the document
ftp>bye	! Return to host

The most recent versions of *FullProf*, written in Fortran 90, are in one of the areas *pub/divers/fullprof.9x* of the same server. Or (example for getting the Windows version of FullProf.98)

ftp>cd pub/divers/fullprof.98	! Go to FullProf.98 (DOS, Windows95/NT)
ftp>cd windows	! Go to Windows95/NT directory
ftp>get README	! Get the installation guide
ftp>binary	! Switch to binary mode
ftp>get winfp98.zip	! Get programs and documentation

Experienced users of ftp can go directly to the subdirectories and get the files they want. The structure of subdirectories matches the different platforms in which *FullProf* can be run. Details are given in the file **fpreadme**. Recently the anonymous ftp-area can be accessed via the WEB through the URL of the LLB:

<http://www-llb.cea.fr/fullweb/powder.htm>

or

<ftp://charybde.saclay.cea.fr/pub/divers/>

Working with powder diffraction data cannot be properly achieved without visual tools. It is of capital importance to have a plot program in order to visualise the observed versus calculated powder pattern and their difference. Such a program is not included in the *FullProf* executable code. Different freeware, shareware, or commercial programs can be user for this task. On the PC-world a very useful program is *WinPLOT*R (written by Thierry Roisnel in collaboration with the author at the LLB). *WinPLOT*R can be obtained in the same **ftp-**

area as **FullProf** in the directory *pub/divers/winplotr*. The program **WinPLOT**R is also distributed with the Windows 9x/NT version of **FullProf**. So users working with Windows 9x/NT can get the complete kit in the single file:

<ftp://charybde.saclay.cea.fr/pub/divers/fullprof.98/windows/winfp98.zip>

Or for the latest version handling multiple patterns simultaneously:

<ftp://charybde.saclay.cea.fr/pub/divers/fullprof.2k/windows/winfp2k.zip>

The structure of the directories of the Web site or the name of some files may be changed and be eventually different than those described here. One can also access to the **FullProf** / **WinPLOT**R areas through the CCP14 (<http://www.ccp14.ac.uk>) site that acts as a mirror of the Saclay site.

To install correctly the program under Windows the user should read carefully the **README** file contained in the same area as **winfp2k.zip**, or use the **install** program included in the kit. In case of troubles the only important point that the user should know is that an environment variable, called FULLPROF, pointing to the directory where the executable program is placed, must be created. Another variable called WINPLOT

```
SET WINPLOT
```

The label of the disk (d:\) and the name of the directory should be selected by the user.

## Technical Support

The author does not provide technical support to the users of the program. If you have any questions regarding the use of **FullProf**, troubles with its installation or running the program try the following steps in the given order.

- Read the relevant manual sections carefully, paying particular attention to examples files.
- Ask to someone who is an experienced user of the program in the surroundings.
- Send an e-mail to one of the lists concerned with powder diffraction in the Internet.
- Send an e-mail to [juan@llb.saclay.cea.fr](mailto:juan@llb.saclay.cea.fr) (response depends on availability of the author, so do not expect to receive an answer immediately!)

# General Information on FullProf

## Purpose, reference and documentation

The program has been mainly developed for Rietveld analysis [H.M. Rietveld, *Acta Cryst.* **22**, 151 (1967); H.M. Rietveld, *J. Applied Cryst.* **2**, 65 (1969); A.W. Hewat, Harwell Report No. 73/239, ILL Report No. 74/H62S; G. Malmros & J.O. Thomas, *J. Applied Cryst.* **10**, 7 (1977); C.P. Khattak & D.E. Cox, *J. Applied Cryst.* **10**, 405 (1977)] (structure profile refinement) of neutron (nuclear and magnetic scattering) or X-ray powder diffraction data collected at constant or variable step in scattering angle  $2\theta$ . The program can be also used as a Profile Matching (or pattern decomposition) tool, without the knowledge of the structure. Single Crystal refinements can also be performed alone or in combination with powder data. Time-of-flight (TOF) neutron data analysis is also available. Energy dispersive X-ray data can also be treated but only for profile matching.

The first versions of the program **FullProf** were based on the code of the DBW program, which, in turn, is also a major modification of the original Rietveld-Hewat program. An early version is discussed in the Young and Wiles article published in [D.B. Wiles & R.A. Young, *J. Applied Cryst.* **14**, 149 (1981); D.B. Wiles & R.A. Young, *J. Applied Cryst.* **15**, 430 (1982)] and described in the user's guide distributed by R.A. Young. The program **FullProf** was developed starting with the code DBW3.2S (Versions 8711 and 8804), but it has been so much modified that only the name of some basic subroutines and variables keep their original names. However, the main control input file created for use with DBW (and DBWS) program can be used by **FullProf** with minor modifications. This file is accepted by **FullProf**, that reads it in "interpreted free format". The file generated, at the end of a run, by **FullProf** cannot be read by DBWS. If the first position of a line in the file contains the symbol ! the whole line is considered as a comment. The comments are useful for remembering the name of variables and flags and facilitate the use of the program.

Two versions of the source code exist at present. The first corresponds to a source written in standard FORTRAN 77 (F77) language, and is organised as to be easily adapted to different computers. This version is that running in multiple platforms. The second version of the source code (**FullProf.9x/2k**) has been developed from the previous one, and it has been totally re-written in a subset (ELF90) of the new standard Fortran 95 (F95). It uses the new syntax and features of Fortran 95. This last version has many more options than the F77 version, which is no more developed (Version 3.5d - Oct98). The current version works with some *allocatable* arrays, in which the user can directly control the dimensions of important arrays at run time. The future development of **FullProf** will be continued only within the F95 version of the source code.

## Features of FullProf.9x/2k

Some of the most important features of **FullProf** are summarised below:

- X-ray diffraction data: laboratory and synchrotron sources.
- Neutron diffraction data: Constant Wavelength (CW) and Time of Flight (TOF).
- One or two wavelengths (eventually with different profile parameters).
- The scattering variable may be  $2\theta$  in degrees, TOF in microseconds and Energy in KeV.
- Background: fixed, refinable, adaptable, or with Fourier filtering.
- Choice of peak shape for each phase: Gaussian, Lorentzian, modified Lorentzians, pseudo-Voigt, Pearson-VII, Thompson-Cox-Hastings (TCH) pseudo-Voigt, numerical, split pseudo-Voigt, convolution of a double exponential with a TCH pseudo-Voigt for TOF.
- Multi-phase (up to 16 phases).
- Preferred orientation: two functions available.
- Absorption correction for a different geometries. Micro-absorption correction for Bragg-Brentano set-up.
- Choice between three weighting schemes: standard least squares, maximum likelihood and unit weights.
- Choice between automatic generation of *hkl* and/or symmetry operators and file given by user.
- Magnetic structure refinement (crystallographic and spherical representation of the magnetic moments). Two methods: describing the magnetic structure in the magnetic unit cell or making use of the propagation

vectors using the crystallographic cell. This second method is necessary for incommensurate magnetic structures.

- Automatic generation of reflections for an incommensurate structure with up to 24 propagation vectors. Refinement of propagation vectors in reciprocal lattice units.
- *hkl*-dependence of FWHM for strain and size effects.
- *hkl*-dependence of the position shifts of Bragg reflections for special kind of defects.
- Profile Matching. The full profile can be adjusted without prior knowledge of the structure (needs only good starting cell and profile parameters).
- Quantitative analysis without need of structure factor calculations.
- Chemical (distances and angles) and magnetic (magnetic moments) slack constraints. They can be generated automatically by the program.
- The instrumental resolution function (Voigt function) may be supplied in a file. A microstructural analysis is then performed.
- Form factor refinement of complex objects (plastic crystals).
- Structural or magnetic model could be supplied by an external subroutine for special purposes (rigid body TLS is the default, polymers, small angle scattering of amphiphilic crystals, description of incommensurate structures in real direct space, etc).
- Single crystal data or integrated intensities can be used as observations (alone or in combination with a powder profile).
- Neutron (or X-rays) powder patterns can be mixed with integrated intensities of X-rays (or neutron) from single crystal or powder data.
- Full Multi-pattern capabilities. The user may mix several powder diffraction patterns (eventually heterogeneous: X-rays, TOF neutrons, etc.) with total control of the weighting scheme.
- Montecarlo/Simulated Annealing algorithms have been introduced to search the starting parameters of a structural problem using integrated intensity data.

## Running the program

The program **FullProf** exists in two forms under the operating system Window9x/2k/NT: the console mode program **fp2k.exe** or the Windows application **wfp2k.exe**. Both programs are identical but the Windows application can be run just by clicking on an *alias* put on the desktop or run from a graphic interface. In other operating systems only the console mode is available.

To run the program in a DOS/Unix shell the user has to invoke the name of the executable file or an appropriate *alias*, for instance:

```
FULLPROF <cr>, or FullProf <cr>, or fp2k <cr>...etc
```

Of course, the executable file must be placed in an accessible path. **FullProf** can also be run from a command file. After invoking the execution of the program the following dialog appears in the current window:

```
*****
** PROGRAM FULLPROF.2k (Version 1.9a - Mar2001-LLB JRC) **
*****
      M U L T I  -- P A T T E R N
      Rietveld, Profile Matching & Integrated Intensity
      Refinement of X-ray and/or Neutron Data
      (Multi_Pattern: DOS-version)

==> Give the code of the files (xx for xx.pcr):
```

After entering a value for **xx**, hereafter assumed to be **CODFIL**, the program prompts the following question:

```

==> Give the name of data file      (yy for yy.dat      )
                                     (      or yy.uxd      )
                                     (<cr> = CODFIL      ):

```

If the user answer with <cr> the name of the data file is **CODFIL.dat** (or **CODFIL.uxd**). We assume, in the following, the user has attributed the value **FILE** to the item **yy**. The file **CODFIL.pcr** must be created (from the scratch or by modifying an existing one) with the help of an ASCII editor. This file contains the diffraction conditions and crystallographic information needed by the program. The optional file **FILE.dat** (or **FILE.uxd**) contains the profile intensity of the powder diffraction pattern.

The program and input files can also be invoked directly in a single line as:

```
LocalPrompt>FullProf  CODFIL  FILEDAT
```

If **FILEDAT** is absent, the code of the data file is assumed to be the same as that of the **CODFIL.pcr** file. For using the Windows 9x/NT version you may create a shortcut pointing to the program in the desktop and then double click on it, invoke the program from a DOS window, or run it from within **WinPLOT**.

For doing sequential refinements the user can run the program using a command file, or answer **CYC** to the prompt asking for the code of the files, or use **CYC** as **CODFIL** name when using direct invoking.

In the last two cases another dialog opens:

```
==> Code of the starting *.pcr file (xx for xx.pcr):
```

Here the user should answer with the xx explicit name (let us assume that the code is **CODFIL**).

```

==> Give the code of data files (yy for yynnn.dat,uxd,acq)
                                     (<cr> =CODFIL  ):

```

As suggested by the question the name of the data files should have a part that constitutes the code followed by an ordinal number. The user should give here the code. Let us assume that the code is **FILEDAT**.

```

==> Number of the starting *.dat file: 46
==> Number of the      last *.dat file: 133

```

The user should give the ordinal numbers of the first and last files to be processed. The program runs by using the file **CODFIL.pcr** to process the file **FILEDAT46.dat**. The updated **CODFIL.pcr** is then used to process the file **FILEDAT47.dat**, etc. At the end, when the file **FILEDAT133.dat** has been processed, the control file contains the parameters adequate to the last treated file. The results of the whole set of treatments are stored in file **CODFIL.rpa**.

The user may create his (her) own scripts (or bat-files to be executed in a DOS shell) invoking the program to adapt the execution of the program in different contexts. For using scripts only the console version of the program should be used. It may be advantageous to take into account that the console version admits three arguments on the command line. For example:

```
Fp2k  pbso4a    pbso4    pblog
```

Means that the input control file *pbso4a.pcr* is read, as well as the data file *pbso4.dat*. The normal screen output is directed to the file *pblog.log*, so that one can run **FullProf** sequentially several times with different input files in batch mode.

## Input files

In the following, references to some variables are done without explicit explanations; this means that they are explained in the appendix, where it is described in detail the contain of the input files.

### **CODFIL.pcr**

Input control file. It will be called sometimes **PCR-file**. It must be in the current directory to run the program. This file contains the title and crystallographic data and must be prepared by the user with the help of a file editor. There are two different formats for this file: the first one is free format and closely related to that of the DBWS program. The second is based on keywords and commands<sup>1</sup>. Within the free format type of the file there are two slightly different ways of writing the PCR-file: the classical way adapted to treat only a single pattern, and the new way suitable to treat multiple pattern refinements.

This file is normally updated, or written to **CODFIL.new**, every time you run the program. In the first stages of a refinement, it is wise to use the option generating a **new** file. The complete description of the file **CODFIL.pcr** is given in the appendix of this document. The text file **FULLPROF.INS** or in the HTML file **fp\_frame.htm** to be used locally with a WEB browser correspond to older versions of *FullProf*. The current version of the program is totally compatible with older versions except in some particular points that are described in the text file **fp2k-v1\_7.inf**.

The following files are **optional** :

### **FILE.dat**

Intensity data file, its format depends on instrument. This corresponds to the profile intensity of a powder diffraction pattern. If you do not specify the name FILE, the program takes FILE=CODFIL. It is not necessary for pattern calculation modes. In the current version of the program the extension may be different from "dat". The program recognises automatically (the extension is not given) the following extensions: **dat**, **uxd**, **acq**. The user may specify his own extension giving the complete name of the file. If multiple patterns are treated simultaneously a file FILE.dat exists with a different name for each pattern.

### **FILE.bac (or CODFIL.bac)**

Background file. The program uses this file to calculate the background at each value of the scattering variable. There are two types of formats for this file:

1. The first format is the same as that of FILE.dat for INSTRM=0:
  - First line: 2 $\theta$ /TOF/Energy (initial) step 2 $\theta$ /TOF/Energy (final), any comment
  - Rest of lines: list of intensities in free format.
2. The second format is the is adapted to the case were there is no fixed step in the scattering variable. The first line is a comment and the rest of lines are pairs of values, scattering variable – intensity, in free format.

The program may generate this file, from refined polynomial or interpolated data, if the user asks for it.

### **CODFILn.hkl or hkl<sub>n</sub>.hkl**

Set of files with the reflections corresponding to phase  $n$  ( $n$  is the ordinal number of a phase). These files are optional and depend on the value of the parameter  $IRF(n)$ . The program reads the list of reflections instead of generating them.

---

<sup>1</sup> This last format is not available at present

## MYRESOL.irf

File describing the instrumental resolution function. Any legal filename can be used and its content depends on the value of the parameter IRESO.

## global.shp or CODFIL.shp

File providing a numerical table for calculating the peak shape and its derivative.

## CODFIL.cor

User-defined intensity corrections. Two types of corrections may be applied.

- In the first case the corrections are applied to the integrated intensities as a multiplier constant. The file **CODFIL.cor** starts with a comment and in the rest of the file one pair Scattering Variable – Correction is given per line.
- In the second case the correction is applied to the profile intensities. The format depends on a number of variables. See appendix for details.

# Output files

Except for **CODFIL.out** and **CODFIL.sum**, the creation of output files depends on the value of a flag that is quoted in parenthesis. All possible values of the flags are given in the appendix.

## CODFIL.out

This is the main output file that contains all control variables and refined parameters. Its content depends on the values of flags set by the user.

## CODFIL.prf or CODFIL\_p.prf (IPL2 different from zero)

Observed and calculated profile: to be fed into visualisation programs. This file is used automatically by **WinPLOT**. In case of multiple pattern refinements a file CODFIL\_p.prf is created for each pattern, where p is the ordinal number of the diffraction pattern.

## CODFIL.rpa (JCIL=1)

Summary of refined parameters. Short version of CODFIL.sum . This file has the “append” attribute, so if it exists the new output is appended. It is useful when running **FullProf** in cyclic modes. An auxiliary program can extract values of particular parameters as a function of temperature, number, etc.

## CODFIL.sym (IPL1=JSY=1)

List of symmetry operators

## CODFIL.sum

Parameter list after last cycle: summary of the last parameters, their standard deviations and reliability factors. An analysis of the goodness of the refinement is included at the end if IANALY=1.

## CODFIL.fou

(JFOU=1)

$h, k, l$ , Structure Factors in Cambridge (CCSL) format to be fed into FOURTK (FOURPL) to produce Fourier maps. It corresponds to the file usually called HKLFF.DAT but you must prepare the second file CRYST.cry.

(JFOU=2)

List of 'observed' structure factors in SHELXS format  $h, k, l, F_{obs}, S(F_{obs})$  (3I4,2F8.2)

(JFOU=-1 or -2)

As above but the structure factors are calculated in another way. The  $F_{calc}$  in JFOU>0 may depend on the peak shape and the integration interval, because they are obtained by integration of the calculated profile in the same way as the  $F_{obs}$  are obtained from  $I_{obs}$ . If JFOU is negative,  $F_{calc}$  are really the structure factors of the conventional cell in absolute units.

(JFOU=3)

Format suitable for the program FOURIER

$h, k, l, A, B, F_{obs}, \sin \theta / l$

$A$  and  $B$  are the real and imaginary parts of the calculated structure factors. The observed  $F_{obs}$  and calculated structure factors of the conventional cell are in absolute units.

(JFOU=4)

Format suitable for the program GFOURIER

$h, k, l, F_{obs}, F_{calc}, Phase$

$Phase$  is the phase in degrees. The observed ( $F_{obs}$ ) and calculated ( $F_{calc}$ ) structure factors of the conventional cell are in absolute units.

**CODFILn.ins** (JFOU=2)

Template of the input control file for the program SHELXS.

**CODFILn.inp** (JFOU=3, 4)

Template of (G)FOURIER \*.inp file.

**CODFILn.hkl**

Files that can be input or output files. The content depends on the value of  $IRF(n)$

**CODFIL.int**

Single integrated intensity file when the program is used for refining with ICRYG=1, 2, 3 and IRF=4.

**CODFIL.hkl**

Complete list of reflections of each phase. This file can be used as a CODFILn.HKL files for new runs.



- JLKH=1
  - If JOBTYP < 2  
Code,  $h, k, l$ ,  $mult, d_{hkl}, 2q$ , FWHM,  $I_{obs}$ ,  $I_{calc}$ ,  $I_{obs} - I_{calc}$
  - If JOBTYP > 1  
 $h, k, l$ ,  $mult, I_{calc}, 2q, d_{hkl}$
- JLKH=2  
Output for EXPO  
 $h, k, l$ ,  $mult, \sin q / l, 2q, FWHM, F^2, S(F^2)$
- JLKH=3,-3  
Output of real and imaginary part of structure factors (only for crystal structures)  
 $h, k, l$ ,  $mult, F_{real}, F_{imag}, 2q, Intensity$

If JLKH is negative the structure factors are given for the conventional cell, otherwise the structure factor corresponds to the non-centrosymmetric part of the primitive cell.

- JLKH=4  
Output of:  $h, k, l, F^2, S(F^2)$ .  
Where  $F^2$  is the *observed* structure factor squared. The file may be used as an input for a *pseudo-single crystal* integrated intensity file using ICRYG=1 and IRF=4.
- JLKH=5  
Output of:  $h, k, l, mult, F_{calc}, T_{hkl}, d_{hkl}, Q_{hkl}$   
Where  $F_{calc}$  is the module of the calculated structure factor. This file can be used as an input for JBT=-3 IRF=2 in order to perform quantitative analysis without re-calculating the structure factors for each cycle. The  $F_{calc}$  values are given in absolute units for the conventional unit cell.

**CODFIL.sav** (JCIL=2 only outputs if an interval in the scattering variable is given)

List of reflections between two selected angles

$h, k, l$ ,  $mult, I_{obs}, 2q, d_{hkl}$

**CODFILn.dis** (JDIS=3)

List of distances and angles (eventually bond valence calculations) for phase **n**.

**CODFIL.mic** (IRESO ≠ 0)

File containing microstructural information. The use of a resolution file, as well as the profile function NPROF=7, is imperative.

**CODFIL.sim** (abs(JOBTYP) > 1 and IDUM =1)

File containing a simulated diffraction pattern. A Poissonian noise is added to the deterministic calculated pattern. The statistics is controlled by the value of the scale factor. This file may be renamed as a DAT-file and used for refinement in simulation work. The use of the same model as that used for generating the diffraction pattern should give a reduced chi-squared nearly equal to 1.

**CODFILn.sub** (IOT=2, 3)

Files containing the calculated profile corresponding to the phase **n**.

**CODFILn.atm** (IMORE=1 and JDIST= $\pm 1$ , 2)

For JDIST= $\pm 1$  it is supposed that a magnetic phase is concerned (JBT= $\pm 1$ , 5, 10).

If JDIST=+1 the files contain the list of magnetic atom positions within a primitive unit cell corresponding to the phase **n**.

If JDIST=-1 the file is suitable as input to the program **MOMENT** that calculates everything concerned with magnetic structures from the Fourier components and phases of magnetic moments.

If JDIST= 2 a nuclear phase is concerned (JBT=0, 4) and the files contain the list of atom positions within a conventional unit cell.

**CODFILn.sch** (IMORE=1 and JVIEW=1, 2)

Files suitable as input for the programs **SCHAKAL** (JVIEW=1) and **STRUPLO** (JVIEW=2) corresponding to the phase **n**.

**CODFILn.int** (JBT=2, IMORE=1 and JVIEW=11)

Files suitable as input for integrated intensity refinements. The generated file contains a list of overlapped reflections obtained adding integrated intensities from profile matching refinement (JBT=2) when they belong to a cluster. More details about this file is given in the appendix. The file corresponds to the phase **n**.

# The Rietveld Method in Practice

In this chapter some simple rules for starting a Rietveld refinement are stated. After discussing some of these rules and comment about the problems the user can experience in running the program, a detailed description of the examples given in the file *pcr\_dat.zip* is given. The examples treated in this chapter are quite simple. Experienced users may apply other procedures and more sophisticated sets of parameters of peak shapes that will not be discussed here. Recently, the Commission on Powder Diffraction of the International Union of Crystallography has published some guidelines for Rietveld refinement [L.B.McCusker *et al.*, *J. Appl. Cryst.* **32**, 36-50 (1999)] that can be used to complete the short notes provided in this paragraph. Rietveld **refinement** has nothing to do with **structure determination**. To start refining a structure an initial model (even if incomplete) is necessary. This model is supposed to be obtained from a **crystal structure solver** program or by any other mean. Special tutorial documents on Rietveld refinement will be included in the distribution of *FullProf*.

## Rietveld refinement

Although the principles behind the Rietveld profile refinement method are rather simple (see next chapter), the use of the technique requires some expertise. This results merely from the fact that Rietveld refinement uses a least-squares minimisation technique which, as any local search technique, gets easily stuck in false minima. Besides, correlation between model parameters, or a bad starting point, may easily cause divergence in early stages of the refinement. All these difficulties can actually be readily overcome by following a few simple prescriptions:

- Use the best possible starting model: this can be easily done for background parameters and lattice constants. In some cases, in particular when the structural model is very crude, it is advisable to analyse first the pattern with the *profile matching* method in order to determine accurately the profile shape function, background and cell parameters before running the Rietveld method.
- Do not start by refining all structural parameters at the same time. Some of them affect strongly the residuals (they must be refined first) while others produce only little improvement and should be held fixed till the latest stages of the analysis.
- Before you start, collect all the information available both on your sample (approximate cell parameters and atomic positions) and on the diffractometer and experimental conditions of the data measurement: zero-shift and resolution function of the instrument, for instance. Then a sensible sequence of refinement of a crystal structure is the following:
  1. Scale factor.
  2. Scale factor, zero point of detector, 1st background parameter and lattice constants. In case of very sloppy background, it may be wise to actually refine at least two background parameters, or better fix the background using linear interpolation between a set of fixed points provided by user.
  3. Add the refinement of atomic positions and (eventually) an overall Debye-Waller factor, especially for high temperature data.
  4. Add the peak shape and asymmetry parameters.
  5. Add atom occupancies (if required).
  6. Turn the overall temperature factor into individual isotropic thermal parameters.
  7. Include additional background parameters (if background is refined).
  8. Refine the individual anisotropic thermal parameters if the quality of the data is good enough.
  9. In case of constant wavelength data, the parameters SYSCOS and/or SYSIN to correct for instrumental or physical  $2\theta$  aberrations with a COS or SIN angular dependence.
  10. Microstructural parameters: size and strain effects.

In all cases, it is essential to plot frequently the observed and experimental patterns. The examination of the difference pattern is a quick and efficient method to detect blunders in the model or in the input file controlling the refinement process. I may also provide useful hints on the best sequence to refine the whole set of model parameters for each particular case.

When large and unrealistic fluctuations of certain parameters occur from one cycle to the next, examine the correlation matrix: if large values (say larger than 50%) are observed, refine separately the corresponding parameters, at least in the early stages of the refinement.

Finally it must be remembered that there is a limit to the amount of information that can be retrieved from a powder diffraction pattern. Indeed structures with up to a hundred or more structural parameters can be

refined from neutron powder data but such refinements must be performed with great care; for refinements involving a large number of variables the physical significance of certain parameters must be carefully examined. For instance thermal and profile parameters can become poorly defined and act as a dumping ground for systematic errors; then it is preferable to fix their values to a physically reasonable number and exclude them from the refinement.

When the uncertainty concerns the atomic parameters, it may help to provide some external information to the program. This can be achieved for instance by using strict constraints. For instance the displacement (thermal) parameters of chemically similar but crystallographically distinct atoms may be constrained to be identical, or the occupancy of two distinct and partly occupied sites of a structure may be compelled by the chemical analysis of the material. For complex structures it may be necessary to use slack (soft) constraints on distances and angles, or even rigid body constraints.

## Whole-pattern decomposition (Profile Matching)

This procedure, that is also known as Le Bail fitting [A. LeBail, H. Duroy and J.L. Fourquet, *Mat. Res. Bull.* **23**, 447 (1988)], does not require any structural information except approximate unit cell and resolution parameters. A similar method developed by Pawley uses traditional least squares with constraints [G.S. Pawley, *J. Applied Cryst.* **14**, 357 (1981)]. A discussion about the profile matching algorithm involved in this kind of refinement may be found in [J. Rodríguez-Carvajal, *Physica B* **192**, 55 (1993)]. This method makes the data input much simpler and enlarges considerably the field of application of powder pattern profile refinement. However the constraints applied to the refinement are far less severe than for Rietveld refinement and profile matching is thereby more prone to instabilities if profile shape parameters or microstructural parameters are refined. In **FullProf** this refinement mode can be used in two ways:

1. *Profile Matching with constant scale factor* (JBT=2). In this mode the scale factor is not allowed to vary and integrated intensities are refined individually using iteratively the Rietveld formula for obtaining the integrated *observed* intensity. The recommended procedure is as follows:
  - For the first refinement, set *IRF*(*n*) of the phase *n* undergoing profile matching to 0 and the number of refined parameters (*MAXS* on line 9) to zero. Set to 0 the flag controlling the automatic assignment of refinement codes. Run **FullProf** for a few cycles (say 10). This will set up the *hkl* 's and intensity file **CODFILn.hkl**.
  - If the result of the above step is satisfactory (see plot!), rename the file **CODFIL.new** to **CODFIL.pcr**, or use directly **CODFIL.pcr** if it was automatically updated. Edit the new **CODFIL.pcr** file to select the parameters to refine. The progression of the refinement is very similar to that used for Rietveld refinement: zero point of detector, background parameters and lattice constants.

In this mode of refinement **FullProf** cannot calculate *theoretical* line intensities and all *hkl* values permitted by the space group are considered and included in the refinement, which sometimes means a lot of reflections! Using this type of refinement one has to bear in mind that the starting cell parameters and resolution function determines to a large extent the obtained intensity parameters. One cannot expect to refine properly the cell parameters of a compound with a severe overlap of reflections if the starting parameters are of poor reliability. It is wise to start with low angle reflections (without refining the FWHM parameters) and progressively increase the angular domain.

2. *Profile Matching with constant relative intensities* (JBT=3). In this mode the intensities are held fixed and only the scale factor is varied. Since profile matching does not require the calculation of the structure factors it runs faster than Rietveld refinement.

## Hints and tricks

Even if you follow carefully the recommendations mentioned above, you might experience difficulties to refine your data; most of them can be avoided by following a few simple rules:

- If the number of measured reflections is limited, select carefully the refined parameters and keep the others fixed to physically reasonable values or introduce suitable constraints.

- Exclude regions where the background is strongly distorted if any (e.g. background from sample environment may show odd variations) for the background functions used in the program may not be able to cope with it.
- It is important to know beforehand the best peak shape function adapted to your particular diffraction pattern. In general, for constant wavelength and energy dispersive data, the pseudo-Voigt function is well adapted for X-ray and neutron diffraction, with predominant Lorentzian character for the former and Gaussian for the latter. Remember to increase the range of the calculated profile (WDT on line 4) to large values (20-30 or more) for Lorentzian peaks; failure to increase properly this parameter will lead to discontinuities in the edges of the calculated profile [7]. For T.O.F. data the profile function NPROF=8 (convolution of pseudo-Voigt with back-to-back exponentials) is normally well suited.
- The FWHM parameters are sometimes difficult to refine especially for data spanning only a limited angular range or samples giving broad diffraction lines. The best method is, of course, to refine first a standard pattern with no sample broadening in order to determine the FWHM parameters of your instrument, create a resolution file and fit only the FWHM parameters (size and strain) characteristics of your sample.

## ***Trouble shooting***

- If you experience difficulties from the very beginning (for instance a singular matrix at the first refinement), start refining the scale factors only and examine the difference pattern with a plotting program. These will most of the time reveal a glaring blunder in the input data (zero-shift, step size, angular limits etc).
- Owing to the complexity of the control file, error messages from the program are not always easy to decipher in that they do not necessarily point to the initial error but merely to one of its consequences. Most of the time, run time errors result from an error in the sequence of lines in the control file, e.g., an inadequacy between the number of atoms stated on line 11.2 and the number of atomic positions given on lines 11.4.

## ***Examples. Content of pcr\_dat.zip***

To test the installation of the program, or for training purposes, a list of complete examples are provided together with **FullProf**. The file *pcr\_dat.zip* can be obtained by anonymous ftp to the server *charybde.saclay cea.fr* in the same area as the program. Set the mode to *binary* in order to get properly the file. The file *pcr\_dat.zip* must be *unzipped* using PKWARE pkzip/pkunzip or WinZip. The resulting ASCII files are in PC/DOS format. They have to be converted to UNIX, Mac, or VMS using one of the appropriate utilities (dos2unix, editor...).

The files contained in *pcr\_dat.zip* can be used for testing **FullProf**. They have been selected in order to illustrate the use of **FullProf** in a variety of situations. In no way the proposed models pretend to be the most adequate to the data. In some cases there is a clear disagreement between the data and the model. The user may try to improve the models including new parameters that have a clear physical relevance. Increasing the number of parameters just for getting more nice fits may result in non sense values.

For a quit test under DOS or under UNIX the user can type at the command line:

```
fullprof_alias < tempo.inp
```

**FullProf** will be executed for all files existing in file: *tempo.inp* where the answers expected by the program are collected.

In a Windows 9x/2k/NT environment the user may execute the file **test\_fp.bat** to run all examples. Verify first that the DOS-like version **fp2k.exe** is in a proper directory within the PATH environment variable. The Windows version of **FullProf** is not adequate for rapid tests because the program needs the intervention of the user to answer questions about the continuation or stop the refinement process.

The number and nature of files within file *pcr\_dat.zip* may change with different releases.

At present the files are:

PCR Code	Purpose	Data File
Ce1	refinement of a CeO <sub>2</sub> standard	ceo2.dat
Ce2	"	"
Rutana	Conventional X-ray diffraction pattern: Rutile+Anatase	Rutana.dat
Tbbaco	Conventional X-ray diffraction pattern: Tb <sub>2</sub> BaCoO <sub>5</sub>	Tbbaco.dat
Tbba	Conventional X-ray diffraction pattern: Profile Matching	"
Tb	Search for Tb,Ba and Co by Montecarlo with prev. output	Tb.int
PbSOx	Crystal structure refinement of PbSO <sub>4</sub> with X-rays	Pbsox.dat
PbSO	Profile matching to obtain an overlapped intensity file	"
PbSOM	Search for Pb by Montecarlo using previous output	pbsom.int
Pb	Profile matching test of PbSO <sub>4</sub> neutron data	Pbso4.dat
PbSO4	Crystal structure refinement of PbSO <sub>4</sub>	"
PbSO4a	Crystal structure refinement of PbSO <sub>4</sub> (anisotropic b's)	"
Pb_ho	Artificial multipattern refinement	Pbso4.dat,Pbsox.dat, Hobk.dat
Pb_sing	Example of new format of PCR file adapted for multipattern refinements	Pbso4.dat
Pb_san	Example of Simulated Annealing: solves the structure of PbSO <sub>4</sub>	Pb_san.int
C60s	Compares C <sub>60</sub> x-tal data to form-factor SPHS sin(Qr)/Qr	C60.int
C60	Refinement of C <sub>60</sub> x-tal data using symmetry adapted cubic harmonics. Form-factor type SASH.	C60.int
Dy	Four different ways of refining the crystal	Dy.dat
Dya	and magnetic structure of DyMn <sub>6</sub> Ge <sub>6</sub>	"
Dyb		"
Dyc		"
Hocu	Refinement of the magnetic structure of Ho <sub>2</sub> Cu <sub>2</sub> O <sub>5</sub> (D1B data)	Hocu.dat
Hobb	Refinement with integrated intensities (Nuc+mag)	Hobb.int
Hob	Montecarlo search for mag. moments in Ho <sub>2</sub> BaNiO <sub>5</sub>	"
Hobk1	Three different ways of refining the crystal	Hobk.dat
Hobk2	and magnetic structure of Ho <sub>2</sub> BaNiO <sub>5</sub>	"
Hobk3		"
Cuf1k	Refinement of crystal & magnetic structure of CuF <sub>2</sub> . Microstructural effects (D1A data)	Cuf1k.dat
Pb_san	Example of Simulated Annealing: solves the structure of PbSO <sub>4</sub>	Pb_san.int
La	Two ways for strain refinement in La <sub>2</sub> NiO <sub>4</sub> (D1B)	La.dat
Lab	with low resolution neutron powder data	"
Monte	Montecarlo test with single crystal data	Monte.int
Hmt	Rigid body-TLS refinement of published single X-tal data	Hmt.int
Urea	Test Rigid body with satellites (simulated data)	Urea.dat
Pyr	Test Rigid body with general TLS refinement (sim. data)	Pyr.dat
Ycbacu	YBaCuO with Ca. Data from D1A	Ycbacu.dat
Arg_si	Corrected TOF data of Si from SEPD at Argonne	Arg_si.dat
Cecoal	TOF data from POLARIS at ISIS	Cecoal
Cecual	TOF data from POLARIS at ISIS	Cecual.dat
Lamn_3t2	Constant wavelength neutron data from 3T2 (LLB) of LaMnO <sub>3</sub>	Lamn_3t2.dat
Lamn_pol	TOF data from POLARIS at ISIS on the RT phase of LaMnO <sub>3</sub>	Lamn_pol.dat
Si3n4r	Quantitative phase analysis. Two polymorphs of Si <sub>3</sub> N <sub>4</sub> . (Studvik)	Si3n4r.dat
Sin_3t2	As above but data taken at 3T2 (LLB)	Sin_3t2
Pb_san	Example of Simulated Annealing: solves the structure of PbSO <sub>4</sub>	Pb_san.int
Maghem	Refinement of Fe <sub>2</sub> O <sub>3</sub> -Fe <sub>3</sub> O <sub>4</sub> at RT (D1A data)	Maghem

In general, the user must first run the program to verify that the provided *pcr*-files behave correctly. After that, the user should make a copy of the control files for saving them before running his(her) own options. The best way is to modify the given values for different sets of parameters and run the program. The beginner must make

extensive use of editor-plot cycles. The plot of the file **CODFIL.prf** is of absolutely necessity for knowing the behaviour of the program under bad (or inaccurate) input parameters.

To use the above files for training, the inexperienced user must start with the simplest cases, that is **ce1.pcr** and **ce2.pcr** used to process the file **ceo2.dat**. This file corresponds to a data collection on cerium oxide with a laboratory X-ray powder diffractometer, using  $\text{CuK}\alpha$  doublets. Other simple examples with conventional X-rays are: the Rutile-Anatase mixture, that allow a quantitative analysis of the relative fraction of each component, and the diffraction pattern of  $\text{Tb}_2\text{BaCoO}_5$  presenting micro-absorption effects that produce some negative temperature factors. The user can modify the input file in order to input the micro-absorption correction and look for the changes in the results. The next files to be processed are those of  $\text{PbSO}_4$ . The data file correspond to a laboratory X-ray diffraction pattern (**pbsox.dat**) and to a neutron powder diffraction pattern (**pbso4.dat**) obtained on D1A (ILL) that was used in a Round Robin on Rietveld refinement (*R.J. Hill (1992), J.Appl.Cryst* **25**, 589). For a person working mainly with crystal structures the next files to be studied are: **ycbacu**, **hmt** and **urea** for powder diffraction.

Some files to be used with single crystal data are also given: **c60**. The first one uses a simplistic model (just a spherical shell) for describing the  $\text{C}_{60}$  molecule that gives relatively good results. The user can try this file as an example of special form factor refinement. The free parameter is the radius of the  $\text{C}_{60}$  molecule. The file **monte** corresponds to an artificial use of the Montecarlo technique for searching a starting set of initial parameters. Data are from neutron diffraction on a single crystal of oxidised  $\text{Pr}_2\text{NiO}_{4+\delta}$ .

People interested in magnetic structures may use the rest of the files in the following order.

- **la, lab**: refinement of the low temperature phase crystal and magnetic structure of  $\text{La}_2\text{NiO}_4$ . The data are from a medium-low resolution neutron powder diffractometer (D1B at ILL). This phase present a microstrain that is refined using two equivalent methods in the two files. The magnetic structure is very simple. A peak from an impurity phase is near the first magnetic peak.
- The files **hobb**, **hob**, **hobk1**, **hobk2**, **hobk3** concern the refinement of the crystal and magnetic structures of  $\text{Ho}_2\text{BaNiO}_5$  at 1.5K, using different methods and conditions. The user can verify that **hob.pcr** can *solve* the magnetic structure of  $\text{Ho}_2\text{BaNiO}_5$  just testing random configurations. This is a very favourable case and this method cannot be applied for general magnetic structure determination. The data are from D1B at ILL.
- **Hocu**: refinement of the magnetic structure of  $\text{Ho}_2\text{Cu}_2\text{O}_5$ . The data have been taken on D1B diffractometer at the ILL. Magnetic scattering dominates nuclear scattering. The crystal structure cannot be refined with these data.
- **Cuf1k**: refinement of the magnetic structure of  $\text{CuF}_2$ . The data have been taken on D1A diffractometer when it was installed provisionally at the LLB. Nuclear scattering dominates magnetic scattering. The diffraction pattern cannot be refined properly without taking into account microstructural effects.
- The files **dy**, **dya**, **dyb**, **dyc** use different methods to refine the incommensurate magnetic structure of  $\text{DyMn}_6\text{Ge}_6$ . This is a conical structure that can be refined using a *real space approach* as in **dy** and **dya** or using Fourier components of the magnetic moments, which is the general formalism of **FullProf** for handling magnetic structures. This is the case of files **dyb** and **dyc**.

An example of simulated annealing application is given. The file is **Pb\_san.pcr**, where the user finds a particular case of how to prepare a PCR file adapted for simulated annealing. The user may play with the different parameters (starting temperature, number of Montecarlo cycles per temperature, type of algorithm, number of reflections to be used, etc) to experience when the method is able to solve the  $\text{PbSO}_4$  structure.

The only right way to learn about crystal and magnetic structure refinements is practising with real data as these are. However, it is better when the user try with its own data on a problem of interest to him(her).

# Mathematical Information

In this chapter the basic equations concerned with the calculations performed in **FullProf** are discussed. It is supposed that the reader has a good knowledge of powder diffraction, so the explanations given below concern mainly the *description* of the terms involved in the formulae. The references to numbered lines (e.g. LINE 3, Line 11-4-1) correspond to lines of the main input control file **CODFIL.pcr** described in **FULLPROF.INS**.

## The Rietveld Method. Calculated profile

A powder diffraction pattern can be recorded in numerical form for a discrete set of scattering angles, times of flight or energies. We will refer to this scattering variable as  $T$ . Then, the experimental powder diffraction pattern is usually given as two arrays  $\{T_i, y_i\}_{i=1, \dots, n}$ . The profile can be modelled using the calculated counts  $y_{ci}$  at the  $i$ th step by summing the contribution from neighbouring Bragg reflections plus the background:

$$y_{ci} = \sum_f s_f \sum_h I_{f,h} \Omega(T_i - T_{f,h}) + y_{bi} \quad (3.1)$$

The vector  $\mathbf{h}(= \mathbf{H}, \text{or} = \mathbf{H} + \mathbf{k})$  labels the Bragg reflections, the subscript  $\mathbf{f}$  labels the *phase* and vary from 1 up to the number of *phases* existing in the model. In **FullProf** the term *phase* is synonymous of a *same procedure for calculating the integrated intensities*  $I_{f,h}$ .

$$I_{f,h} = \{LF^2 ATPC\}_{f,h} \quad (3.2)$$

For simplicity we will drop the  $\mathbf{f}$ -index. Sometimes we will refer to the whole arrays  $\{y_i\}$  and  $\{y_{ci}\}$  as  $y_{obs}$  and  $y_{calc}$  respectively. The meaning of the different terms appearing in (3.1) and (3.2) is the following:

- $S_f$  is the scale factor of the phase  $\mathbf{f}$
- $L_h$  contains the Lorentz, polarisation and multiplicity factors
- $F_h$  is the structure factor<sup>1</sup>
- $A_h$  is the asymmetry function
- $T_h$  is the transmission factor
- $P_h$  is the preferred orientation function
- $\Omega$  is the reflection profile function that models both instrumental and sample effects
- $y_{bi}$  is the background intensity

In the following sections we discussed the different terms in more detail. The Rietveld Method consist of refining a crystal (and/or magnetic) structure by minimising the weighted squared difference between the observed  $\{y_i\}_{i=1, \dots, n}$  and the calculated (3.1) pattern  $\{y_{ci}(\mathbf{a})\}_{i=1, \dots, n}$  against the parameter vector  $\mathbf{a}$ . The function minimised in the Rietveld Method is:

$$\mathbf{c}^2 = \sum_{i=1}^n w_i \{y_i - y_{ci}(\mathbf{a})\}^2 \quad (3.3)$$

---

<sup>1</sup> The ratio of the intensities for the two wavelengths handled by the program is absorbed in the calculation of  $F_h^2$ , so that only a single scale factor is required.



with  $w_i = \frac{1}{s_i^2}$ , being  $s_i^2$  the variance of the "observation"  $y_i$ .

## Background

The background intensity  $y_{bi}$  at the  $i$ th step is obtained (see LINE 2 and LINE 6\* or Line 10-2\*) either from an user-supplied table of background intensities, or from a refinable background function:

$$y_{bi} = \sum_{m=0}^{11} B_m \left( \frac{T_i}{BKPOS} - 1 \right)^m + \sum_{j=1}^6 B_{Cj} \frac{\sin(Q_i d_j)}{Q_i d_j} \quad (3.4)$$

The origin of the background polynomial is given by a selectable input parameter BKPOS (LINE 4) and should be supplied by the user. The second sum is used only if NBCKGR=-1. The parameters to be refined are:  $B_0, B_1, \dots, B_{11}, B_{C1}, \dots, B_{C6}, d_1, \dots, d_6$ .  $Q_i$  is the modulus of the scattering vector ( $\mathbf{Q} = 2\mathbf{p}\mathbf{h}$ ) at the  $i$ th step, and is given by:

$$Q_i = 4p \frac{\sin \mathbf{q}}{l} \quad (3.5)$$

where  $l = l_1$  **FullProf**. The parameters  $d_j$  are distances in Å.

The background can be also read from a supplied file **FILE.bac**. The actual background is calculated from the read background applying the following formula:

$$Background(2q) = b_1 Backg \left[ (1 + b_3) 2q + b_4 \right] + b_2 \quad (3.6)$$

The background parameters and codes, given in Line 10-2\*, correspond to the coefficients of the above formula in the order:  $b_1, b_2, b_3, b_4$ . If  $b_1$  is given as zero, the program puts  $b_1 = 1$ . Limits against divergence are fixed by program. The parameter  $b_3$  is allowed to vary up to a maximum value  $ABS(b_3) = 0.1$  and  $ABS(b_4)$  is kept below 3 degrees ( $2q$ ). The user can check the excursion of those parameters out of the allowed range when they are strictly zero and their standard deviation is (fixed arbitrarily to) 0.99999. This option is useful when complicated background shapes are present due to sample environment.

A new procedure for treating the background has also been introduced. The background is adjusted iteratively at each cycle by using a Fourier filtering technique. The starting background is read from the file FILE.bac. At cycle  $n$  the new background is calculated from the old one, cycle  $n-1$ , with the formula:

$$Background(n) = Background(n-1) + Filtered[y_{obs} - y_{cal}](n-1) \quad (3.7)$$

where  $Filtered[y_{obs} - y_{cal}]$  is a strongly smoothed version of  $y_{obs} - y_{cal}$ . The parameter controlling the smoothing is FWINDOW that is equivalent to PST used in subroutine SMOOFT described in Numerical Recipes (see [Numerical Recipes, by W.H. Press, B.P. Flanery, S.A. Teukolsky and W.T. Vetterling (Fortran version), Cambridge University Press, 1990]). The implementation of SMOOFT in **FullProf** is not the same as in Numerical Recipes. When using this method it is wise to draw the final background to see if it is really a smooth curve. This option is only justifiable in cases of a very wavy background. The starting background should be close to the real one.

## Peak-shape functions for constant wavelength data

The profile function  $\Omega$  is selected by the control variable NPROF (LINE 2). All the profile functions used in **FullProf** are normalised to 1 (i.e.  $\int_{-\infty}^{+\infty} \Omega(x) dx = 1$ ). The variable is  $x = T - T_h$ , the FWHM will be called  $H$ . The definition of  $\Omega_i(x)$ , where the subscript  $i$  stands for the value of the variable NPROF, is the following:

$$GAUSSIAN : \Omega_0(x) = G(x) = a_G \exp(-b_G x^2) \quad (3.8)$$

where:

$$a_G = \frac{2}{H} \sqrt{\frac{\ln 2}{p}} \quad b_G = \frac{4 \ln 2}{H^2}$$

The integral breadth of the Gaussian function is:  $b_G = \frac{1}{a_G} = \frac{H}{2} \sqrt{\frac{p}{\ln 2}}$

$$LORENTZIAN : \Omega_1(x) = L(x) = \frac{a_{ML}}{1 + b_L x^2} \quad (3.9)$$

where:

$$a_L = \frac{2}{pH} \quad b_L = \frac{4}{H^2}$$

The integral breadth of the Lorentzian function is:  $b_L = \frac{1}{a_L} = \frac{pH}{2}$

$$MODIFIED LORENTZIAN : \Omega_2(x) = ML(x) = \frac{a_{ML}}{(1 + b_{ML} x^2)^2} \quad (3.10)$$

where:

$$a_{ML} = \frac{4\sqrt{\sqrt{2}-1}}{pH} \quad b_{ML} = \frac{4(\sqrt{2}-1)}{H^2}$$

$$INTERMEDIATE LORENTZIAN : \Omega_3(x) = IL(x) = \frac{a_{IL}}{(1 + b_{IL} x^2)^{3/2}} \quad (3.11)$$

where:

$$a_{IL} = \frac{\sqrt{2^{2/3}-1}}{H} \quad b_{IL} = \frac{4(2^{2/3}-1)}{H^2}$$

$$PEARSONVII : \Omega_6(x) = PVII(x) = \frac{a_{VII}}{(1 + b_{VII} x^2)^m} \quad (3.12)$$

where:

$$a_{VII} = \frac{\Gamma(m)}{\Gamma(m)-1/2} \frac{2\sqrt{2^{1/m}-1}}{\sqrt{p}H} \quad b_{VII} = \frac{4(2^{1/m}-1)}{H^2}$$

$$PSEUDO-VOIGT : \Omega_5(x) = pV(x) = hL'(x) + (1-h)G'(x) \quad 0 \leq h \leq 1 \quad (3.13)$$

The  $pV(x)$  function is a linear combination of a Lorentzian ( $L'$ ) and a Gaussian ( $G'$ ) of the same FWHM ( $H$ )<sup>2</sup>, so there are two parameters characterising the peak shape:  $pV(x) = pV(x, h, H)$ . If  $L'(x)$  and  $G'(x)$  are normalised,  $pV(x)$  is also normalised. The integral breadth of a normalised pseudo-Voigt function is just

<sup>2</sup> The reader can verify that the FWHM of  $pV(x)$  is also  $H$ .

the inverse of the maximum value. The FWHM is the same for  $L(x)$ ,  $G(x)$  and  $pV(x)$ . If the function is multiplied by a constant (integrated intensity) the integral breadth doesn't change:

$$\mathbf{b}_{pV} = \frac{\int \Omega(x) dx}{\Omega(0)} = \frac{\int I_i pV(x) dx}{I_i pV(0)} = \frac{1}{pV(0)}$$

$$\mathbf{b}_{pV} = \frac{pH/2}{\mathbf{h} + (1-\mathbf{h})\sqrt{p \ln 2}}$$

The profile function NPROF=4,  $\Omega_4(x)$ , is a superposition of three  $pV(x)$  functions (see below (3.22)). The difference between NPROF=5 and NPROF=7 is that in the later the parameter  $\mathbf{h}$  is not directly refinable but calculated from  $H_L$  and  $H_G$ . Let us precise this point. The pseudo-Voigt function is an approximation to the Voigt function defined as the convolution of a Lorentzian and a Gaussian:

$$VOIGT: V(x) = L(x) \otimes G(x) = \int_{-\infty}^{+\infty} L(x-u)G(u) du \quad (3.14)$$

where  $L(x)$  and  $G(x)$  have different FWHM ( $H_L$  and  $H_G$ , respectively). The  $pV(x)$  function is an approximation that substitutes the two shape parameters  $H_L$  and  $H_G$  by the pair  $(\mathbf{h}, H)$ .

$$V(x) = L(x) \otimes G(x) = \int_{-\infty}^{+\infty} L(x-u)G(u) du$$

The Voigt function can be written in a closed form in terms of the complex error function and the integral breaths of the Lorentzian ( $\mathbf{b}_L$ ) and Gaussian components ( $\mathbf{b}_G$ ):

$$V(x) = V(x, H_L, H_G) = V(x, \mathbf{b}_L, \mathbf{b}_G)$$

$$V(x) = \mathbf{b}_G^{-1} \operatorname{Re} \left[ \operatorname{erf} \left( \frac{\sqrt{p}}{\mathbf{b}_G} |x| + i \frac{\mathbf{b}_L}{\mathbf{b}_G \sqrt{p}} \right) \right] \quad (3.15)$$

where:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{p}} \int_0^x e^{-t^2} dt$$

Numerically is more easy to calculate the pseudo-Voigt approximation (3.13) using a numerical approximation relating the pairs  $(H_L, H_G)$  and  $(\mathbf{h}, H)$  (see [24]):

$$(H, \mathbf{h}) = F(H_G, H_L)$$

$$H = (H_G^5 + 2.69269H_G^4H_L + 2.42843H_G^3H_L^2 + 4.47163H_G^2H_L^3 + 0.07842H_GH_L^4 + H_L^5) \quad (3.16)$$

$$\mathbf{h} = 1.36603 \frac{H_L}{H} - 0.47719 \left( \frac{H_L}{H} \right)^2 + 0.11116 \left( \frac{H_L}{H} \right)^3 \quad (3.17)$$

The inversion of the above two expressions leads to the relations:

$$(H_G, H_L) = F^{-1}(H, \mathbf{h})$$

$$\frac{H_L}{H} = 0.72928\mathbf{h} + 0.19289\mathbf{h}^2 + 0.07783\mathbf{h}^3 \quad (3.18)$$

$$\frac{H_G}{H} = (1 - 0.74417\mathbf{h} - 0.24781\mathbf{h}^2 - 0.00810\mathbf{h}^3)^{1/2} \quad (3.19)$$

The difference between the  $pV$ -functions calculated with NPROF=5 or NPROF=7 concerns only the parameterisation of  $\mathbf{h}$  and  $H$  which can be simply related to significant physical parameters using NPROF=7.

The dependence of the peak-shape parameters (including FWHM) on the scattering variable ( $2\mathbf{q}$  for constant wavelength diffractometers) is parameterised mainly through the variables defined in Line 11-6-1:  $U, V, W, X, Y, I_G, S_Z$ .

For profiles NPROF=0 to NPROF=6 :

$$H^2 = (U + D_{ST}^2) \tan^2 \mathbf{q} + V \tan \mathbf{q} + W + \frac{I_G}{\cos^2 \mathbf{q}} \quad (3.20)$$

For NPROF=5 (pseudo-Voigt) the  $\mathbf{h}$  parameter can be dependent on  $X$  through the formula:

$$\mathbf{h} = \mathbf{h}_0 + X \, 2\mathbf{q} \quad (3.21)$$

For NPROF=4 (tripled pseudo-Voigt), the three components are assumed to have the same  $\mathbf{h}_0$  and FWHM, so the effective total width depends on the additional shape parameter  $S_{h1}$  (see Line 11-8-3). The profile function is given by the formula:

$$\Omega_4(x) = p_4(x) = XpV(x-D) + (1-X-Y)pV(x) + YpV(x+D) \quad (3.22)$$

where:

$$D = \frac{S_{h1}}{d \cos \mathbf{q}}$$

So, apart from the FWHM that is calculated from  $U, V, W, D_{ST}$  and  $I_G$  parameters for a single component, the profile function has FOUR shape parameters  $\mathbf{h}_0, X, Y$  and  $S_{h1}$ . This function is adapted for medium resolution neutron powder diffractometers having defects on the monochromator and/or the spectral distribution that gives rise to a non-Gaussian distribution of wavelengths.

For NPROF = 6 (Pearson-VII) the  $m$  parameter can be dependent on  $X$  and  $Y$  through the formula:

$$m = m_0 + 100 \frac{X}{2\mathbf{q}} + 10000 \frac{Y}{(2\mathbf{q})^2} \quad (3.23)$$

For NPROF=7, the FWHM of the Gaussian ( $H_G$ ) and Lorentzian ( $H_L$ ) components is calculated as:

$$H_G^2 = (U + D_{ST}^2) \tan^2 \mathbf{q} + V \tan \mathbf{q} + W + \frac{I_G}{\cos^2 \mathbf{q}} \quad (3.24)$$

$$H_L = X \tan \mathbf{q} + \frac{[Y + F(S_Z)]}{\cos \mathbf{q}} \quad (3.25)$$

The units of parameters  $U, V, W, I_G$  are  $(degrees \, 2\mathbf{q})^2$ . The physical meaning of  $I_G, D_{ST}^2$  and  $F(S_Z)$  is briefly discussed in **FULLPROF.INS**.

---

For profiles 0 to 6 and 12:

$$FWHM^2 = (U + DST^2) \tan^2 \mathbf{q} + V \tan \mathbf{q} + W + IG / \cos^2 \mathbf{q}$$

For NPROF = 4 (tripled pseudo-Voigt), the three components are assumed to have the same  $\mathbf{h}_0$  and FWHM, so the effective total width depends on the additional shape parameter Shape1. The profile function is given by the formula:

$$p4(x) = XpV(x-D) + (1-X-Y)pV(x) + YpV(x+D)$$

where

$$D = Shape1 / d \cos \mathbf{q}$$

$$pV(x) = \mathbf{h}_0 L(x) + (1-\mathbf{h}_0)G(x)$$

So, apart from the FWHM that is calculated from U,V,W, DST and IG parameters for a single component, the profile function has FOUR shape parameters  $\mathbf{h}_0$ , X, Y and Shape1. This function is adapted for medium resolution neutron powder diffractometers having defects on the monochromator and/or the guide spatial spectral distribution giving rise to a non-Gaussian distribution of wavelengths.

For NPROF = 5 and 12 (pseudo-Voigt) the  $\mathbf{h}$  parameter can be dependent on X through the formula:

$$pV(x) = \mathbf{h}L(x) + (1-\mathbf{h})G(x)$$

$$\mathbf{h} = \mathbf{h}_0 + X \cdot 2\mathbf{q}$$

For NPROF = 11 (split pseudo-Voigt) DST and IG are common to the left and right parts of the profile. Moreover additional FWHM parameters are used as new shape parameters, so the expression of the left FWHM(L) for NPROF=11 is

$$FWHM^2(L) = (Ul + DST^2)\tan^2 \mathbf{q} + Vl \tan \mathbf{q} + Wl + IG/\cos^2 \mathbf{q} + Shape1/\tan^2 2\mathbf{q}$$

Shape1 is applied only for  $2\mathbf{q} \leq 90^\circ$ . The value of  $\mathbf{h}$  for the left part is given by:

$$\mathbf{h}_l = \mathbf{h}_0 + Xl \cdot 2\mathbf{q}$$

The expression of the right part is:

$$FWHM^2(R) = (Ur + DST^2)\tan^2 \mathbf{q} + Vr \tan \mathbf{q} + Wr + IG/\cos^2 \mathbf{q} + Shape2/\tan^2 2\mathbf{q}$$

Shape2 is applied only for  $2\mathbf{q} > 90^\circ$ . The value of  $\mathbf{h}$  for the right part is given by:

$$\mathbf{h}_r = \mathbf{h}_0 + X_r \cdot 2\mathbf{q}$$

The FWHM and shape parameters for the right part are read in next lines

For NPROF = 6 (Pearson-VII) the m parameter can be dependent on X and Y through the formula:

$$m = m_0 + 100X / 2\mathbf{q} + 1000Y / 2\mathbf{q}^2$$

For NPROF = 7, the FWHM of the two components is calculated as

$$FWHM^2(Gaussian) = (U + DST^2)\tan^2 \mathbf{q} + V \tan \mathbf{q} + W + IG/\cos^2 \mathbf{q}$$

$$FWHM^2(Lorentzian) = X \tan \mathbf{q} + [Y + F(SZ)]/\cos \mathbf{q}$$

To be completed and corrected (List of Formulae to include):

---

New formulae:

$$H_G^2 = (U + (1-x)^2 D_{ST}^2) \tan^2 \mathbf{q} + V \tan \mathbf{q} + W + \frac{I_G}{\cos^2 \mathbf{q}}$$

$$H_L = (X + x D_{ST}) \tan \mathbf{q} + \frac{[Y + F(S_Z)]}{\cos \mathbf{q}}$$

$$H_{GS}^2 = \frac{I_G}{\cos^2 \mathbf{q}}$$

$$H_{LS} = \frac{[Y + F(S_Z)]}{\cos \mathbf{q}}$$

For a particular reflection one can calculate, using the relations between the FWHM(=H) and  $\eta$  of the pseudo-Voigt function and the Gaussian and Lorentzian components of the Voigt function

$$(H_S, \mathbf{h}_S) = F(H_{GS}, H_{LS})$$

Substituting the expressions of the G and L size we obtain similar expressions where  $H_G$  and  $H_L$  are replaced by  $\text{SQRT}(IG)=I^{1/2}_G$  and  $Y_L=Y+F(S_Z)$  respectively. The term in  $1/\cos \mathbf{q}$  cancels out if we call  $H_S = Z_S/\cos \mathbf{q}$

$$Z_S^5 = I_G^{5/2} + 2.69269 I_G^2 Y_L + 2.42843 I_G^{3/2} Y_L^2 + 4.47163 I_G Y_L^3 + 0.07842 I_G^{1/2} Y_L^4 + Y_L^5$$

$$\mathbf{h}_S = 1.36603 \frac{Y_L}{Z_S} - 0.47719 \left( \frac{Y_L}{Z_S} \right)^2 + 0.11116 \left( \frac{Y_L}{Z_S} \right)^3$$

From  $Z_S$  and  $\mathbf{h}_S$  we obtain the integral breadth corresponding to size effects just from the corresponding expression for the normalised pseudo-Voigt function.

$$\mathbf{b}_{pV}(\text{Size}) = \frac{\mathbf{p} Z_S / 2}{\mathbf{h}_S + (1 - \mathbf{h}_S) \sqrt{\mathbf{p} \ln 2}} \times \left( \frac{\mathbf{p}}{180} \right) \times \frac{1}{\cos \mathbf{q}}$$

If  $Z_S$  and  $\eta_S$  have been calculated from the values of  $IG$  and  $Y$  as refined in FullProf, they are expressed in degrees, so we have to transform to radians, as written above, in order to get the apparent size in the correct units. From the Scherrer formula we have:

$$\mathbf{b}_h = \mathbf{b}_{pV}(\text{size}) = \frac{1}{D_h \cos \mathbf{q}} = \frac{\mathbf{p} Z_S / 2}{\mathbf{h}_S + (1 - \mathbf{h}_S) \sqrt{\mathbf{p} \ln 2}} \times \left( \frac{\mathbf{p}}{180} \right) \times \frac{1}{\cos \mathbf{q}}$$

So, the apparent size considering both Gaussian and Lorentzian components, and under the hypothesis of pseudo-Voigt (Voigt) peak shape, is given by the expression:

$$D_h = \frac{\mathbf{h}_S + (1 - \mathbf{h}_S) \sqrt{\mathbf{p} \ln 2}}{Z_S} \times \left( \frac{360 I}{\mathbf{p}^2} \right)$$

Where  $Z_S$  and  $\mathbf{h}_S$  have been obtained from  $I_G=IG$  and  $Y_L=Y+F(S_Z)$  using the TCH formula.

Note: **Non-normalised pseudo-Voigt** appears in many papers.

Let us call the non-normalised function as:

$$\mathbf{p} V_n(x) = \mathbf{h}_n L_n(x) + (1 - \mathbf{h}_n) G_n(x) \quad \mathbf{p} V_n(0) = 1$$

where:

$$L_n(x) = \frac{1}{1 + b_L x^2} \quad L_n(0) = 1$$

$$G_n(x) = \exp(-b_G x^2) \quad G_n(0) = 1$$

The integral breadth of a non-normalised pseudo-Voigt function of peak intensity  $I_0$  and FWHM= $H$  is given by:

$$b_{pVn} = \frac{\int \Omega(x) dx}{\Omega(0)} = \frac{\int I_0 p V_n(x) dx}{I_0 p V_n(0)} = \frac{\int p V_n(x) dx}{1}$$

$$b_{pVn} = h_n \int L_n(x) dx + (1-h_n) \int G_n(x) dx$$

$$b_{pVn} = h_n \frac{1}{a_L} + (1-h_n) \frac{1}{a_G} = h_n \frac{pH}{2} + (1-h_n) \frac{H}{2} \sqrt{\frac{p}{\ln 2}}$$

The integral breadth of a particular peak is independent of the formulation of the pseudo-Voigt function. Both descriptions gives the same FWHM and the same integral breath, so the numerical relation between the  $\eta$  values is given by:

$$\text{Non-normalised: } b_{pVn} = h_n \frac{pH}{2} + (1-h_n) \frac{H}{2} \sqrt{\frac{p}{\ln 2}}$$

$$\text{Normalised: } b_{pV} = \frac{pH/2}{h + (1-h)\sqrt{p \ln 2}}$$

$$h_n \frac{pH}{2} + (1-h_n) \frac{pH}{2\sqrt{p \ln 2}} = \frac{pH/2}{h + (1-h)\sqrt{p \ln 2}}$$

$$h_n = \frac{h}{h + (1-h)\sqrt{p \ln 2}}$$

$$h = \frac{h_n}{h_n + (1-h_n) \frac{1}{\sqrt{p \ln 2}}}$$

## ***Peak-shape functions for Neutron Time of Flight***

The profile function  $\Omega$  for TOF corresponds to the control variable NPROF=9, 10. The peak shape is the same in both cases but the TOF position of the Bragg peaks are calculated in different ways for the two functions. (To be completed)

## ***Monochromator, Lorentz and geometrical corrections***

### ***Bragg-Brentano or Debye-Scherrer geometry***

Monochromator polarisation correction (CTHM and  $K$  in LINE 4); the Lorentz polarisation factor  $Lp$  is calculated as :

$$Lp = \frac{1 - K + K \cos^2 2q_{Monok} \cos^2 2q}{2 \sin^2 q \cos q} \quad (3.26)$$

with  $CTHM = \cos^2 2\mathbf{q}_{Monok}$ . For instance with a Graphite monochromator,  $CTHM = 0.8351$  and  $0.7998$  for  $CuK_b$  and  $CuK_a$  respectively.

Note that  $K$  is used only for synchrotron data (ILOR=3) and does not have to be input for other kind of data taken with the specified geometry.

- For neutrons,  $K=0.0$ .
- For characteristic x-ray radiation (non-polarised beam),  $K=0.5$ .
- For synchrotron radiation,  $K$  must be given ( $K \approx 0.1$ )

### ***Transmission flat-plate geometry***

In the transmission geometry, flat plate with the scattering vector within the plate (Stoe geometry for X-rays, ILOR=2) the Lorentz factor is:

$$L = \frac{1}{\sin 2\mathbf{q}} \quad (3.27)$$

and the polarisation correction is given by:

$$p = K \frac{1 + c_1 \cos^2 2\mathbf{q}}{1 + c_1} + (1 - K) \frac{1 + c_2 \cos^2 2\mathbf{q}}{1 + c_2} \quad (3.28)$$

where

$$\begin{aligned} c_1 &= |\cos 2\mathbf{q}_{Monok}| = \sqrt{CTHM} \\ c_2 &= \cos^2 2\mathbf{q}_{Monok} = CTMH \\ K &= \text{fraction of perfect crystal contribution} \end{aligned}$$

## ***Asymmetry***

The default asymmetry correction (see RLIM in LINE 4) follows the expression given in [12]. The asymmetry correction adopts the form:

$$A_h = 1 + \frac{P_1 F_a(z) + P_2 F_b(z)}{\tanh \mathbf{q}_h} + \frac{P_3 F_a(z) + P_4 F_b(z)}{\tanh 2\mathbf{q}_h} \quad (3.29)$$

where

$$z = \frac{2\mathbf{q}_i - 2\mathbf{q}_h - S_{shf}}{FWHM}$$

$S_{shf}$  includes the zero-point and other shifting terms,

$$F_a(z) = 2z \exp(-z^2), \quad F_b(z) = 2(2z^2 - 3)F_a(z)$$

The asymmetry correction (3.29) has four independent parameters, ( $P_1, P_2, P_3, P_4$ ) which are read in Line 11-8-1.

If NPHASE\$<\$0 the method proposed by C.J.Howard (see [11]), is used. The approximation of the convolution integral is performed using the Bode's rule (Simpson's formula for five points). The profile is calculated as a superposition of five profile functions  $g(x)$  (only pseudo-Voigt NPROF=5 and NPROF=7 are implemented for this correction) calculated at displaced points:



$$\Omega(x) = \frac{1}{90} \{ 7g(x) + 32g(x + c_2P) + 12g(x + c_3P) + 32g(x + c_4P) + 7g(x + c_5P) \} \quad (3.30)$$

where  $x = 2\mathbf{q}_i - 2\mathbf{q}_h - S_{shf}$ ,  $P$  is the asymmetry parameter and  $\cot 2\mathbf{q} = 16c_2 = 4c_3 = \frac{1}{9}c_4 = c_5$ .

The parameters  $P_1, P_2$  normally suffice for neutron diffraction. The starting value of these parameters should be significantly lower than unity (e.g. 0.01) to get good results. If the value of the asymmetry parameter used in the Rietveld's formula (see [2]) is kept fixed in **FullProf** the refinement may be very bad and may not converge.

## Preferred orientation

Only two preferred orientation functions are currently implemented in the program:

- NORI=0 → the exponential function:

$$P_h = G_2 + (1 - G_2) \exp(G_1 \mathbf{a}_h^2) \quad (3.31)$$

Where  $G_1$  and  $G_2$  are refinable parameters and  $\mathbf{a}_h$  is the acute angle between the scattering vector and the normal to the crystallites (platy habit). Note that setting  $G_1$  to any number >99.0 for a phase causes the program to generate, for that phase, only those reflections for which  $[hkl]^*$  is parallel to the preferred orientation vector PREF specified in Line 11-2.

- NORI=1 → Modified March's function:

$$P_h = G_2 + (1 - G_2) \left( (G_1 \cos \mathbf{a}_h)^2 + \frac{\sin^2 \mathbf{a}_h}{G_1} \right)^{-3/2} \quad (3.32)$$

Where  $G_1$  and  $G_2$  are refinable parameters. This expression is adapted to both fibber and platy habits:

- $G_1 < 1$ : Platy habit ( $\mathbf{a}$  is the acute angle between the scattering vector and the normal to the crystallites)
- $G_1 = 1$ : No preferred orientation
- $G_1 > 1$ : Needle-like habit ( $\mathbf{a}$  is the acute angle between the scattering vector and the fibber axis direction)

Note that these values of  $G_1$  correspond to the Bragg-Brentano geometry of usual X-ray powder diffractometers; for the Debye-Scherrer geometry of most neutron powder diffractometers the opposite holds.

The parameter  $G_2$  represents the fraction of the sample that is not textured. The program put its value between 0 and 1 in case of divergence.

## Absorption and micro-absorption

For Debye-Scherrer data, intensities may be corrected for the effects of sample absorption by applying the following transmission factor:

$$T_h = \exp \left\{ -(1.7133 - 0.0368 \sin^2 \mathbf{q}_h) \mathbf{mR} + (0.0927 + 0.375 \sin^2 \mathbf{q}_h) (\mathbf{mR})^2 \right\} \quad (3.33)$$

Where  $\mu$  is the linear absorption coefficient and  $R$  the radius of the cylindrical sample.

For a flat plate in transmission geometry for X-rays (ILOR=2) the absorption correction is implemented as:

$$T_h = \frac{\exp(-\mu t / \cos q_h)}{\cos q_h} \quad (3.34)$$

Where  $t$  is the "effective" thickness of the sample. The product  $\mu t$  is given in the same place as  $\mu R$ .

For Bragg-Brentano geometry in X-rays an angular-dependent microabsorption correction has been introduced following [25] and [26]. The factor  $T_h$  becomes:

$$T_h \Rightarrow T_h S_r(\mathbf{h}) = T_h (1 - P(\mathbf{h})) \quad (3.35)$$

Where  $S_r(\mathbf{h})$  is given by the formula 6 of [25] (or  $P(\mathbf{h})$  is given by formula 17 of [26]).

$$P(\mathbf{h}) = P_0 + C_p \frac{t}{\sin q_h} \left( 1 - \frac{t}{\sin q_h} \right) \quad (3.36)$$

The limitations and degree of applicability of the above formula are explained in the cited references.

## Systematic line-shifts

Powder diffraction data are sometimes affected by systematic aberrations resulting from the sample itself or from an improper setting of the sample or diffractometer. **FullProf** gives the possibility to correct for two of the most commonly occurring errors by refining the parameters called SYCOS and SYSIN (Line 10-1). These parameters relate to errors having a  $\cos q$  and  $\sin q$  dependency, respectively. The corresponding errors originate from a different physical or/and geometrical problem depending on the diffraction geometry. They are summarised below:

### Bragg-Brentano parafocusing arrangement

The two largest systematic aberrations of  $q-2q$  powder diffractometers operating in this geometry arise from specimen displacement and transparency; the sample displacement error is one of the largest systematic errors affecting line positions in this geometry. It is given by:

$$\Delta 2q = \frac{-2s}{R} \cos q \quad [\text{in radians}] \quad (3.37)$$

where  $s$  is the displacement of the sample surface with respect to the axis of the goniometer and  $R$  the radius of the goniometer circle. The negative sign means that a displacement away from the centre of the focusing circle moves the diffraction lines to lower  $2q$  angle. The refinable parameter is  $SYCOS = -2s/R$ . This is by far the largest systematic aberration in this geometry. As the angles are expressed in degrees in **FullProf**, the sample offset can be calculated as:

$$s = \frac{P}{180} R SYCOS$$

The transparency correction is given by the relation:

$$\Delta 2q = \frac{1}{2\mu R} \sin 2q \quad [\text{in radians}] \quad (3.38)$$

where  $\mu$  is the linear absorption coefficient of the sample. This relation holds in the case of thick absorbing samples and the refinable parameter is  $SYSIN = 1/(2\mu R)$ . For thin transparent samples, the correction would write:

$$\Delta 2\mathbf{q} = \frac{t}{R} \cos \mathbf{q} \quad [\text{in radians}] \quad (3.39)$$

where  $t$  is the sample thickness; this (less usual) correction is not explicitly included in the code but can be accounted for by the displacement correction which turns out to show the same  $2\mathbf{q}$  dependency. Note however that samples requiring that kind of correction would also give biased integrated intensities; correction for this effect is not implemented in **FullProf**. For further details on systematic aberrations in Bragg-Brentano geometry, see [16].

### ***Debye-Scherrer***

The largest shifts of Debye-Scherrer rings result from sample off centring and absorption. Eccentricity perpendicular to the incident beam direction is normally a second order effect if both sides of the Debye-Scherrer ring are measured.

If only one side of the cone is measured, the line shift takes the form:

$$\Delta 2\mathbf{q} = \frac{e}{R} \cos \mathbf{q} \quad [\text{in radians}] \quad (3.40)$$

where  $e$  denotes the eccentricity, that is the refined parameter is  $SYCOS = e/R$ .

Eccentricity in the incident beam direction is observed in both cases and takes the form:

$$\Delta 2\mathbf{q} = \frac{e}{R} \sin 2\mathbf{q} \quad [\text{in radians}] \quad (3.41)$$

i.e., the refined parameter is  $SYSIN = e/R$ . The correction is negative for a shift along the beam direction towards the detector. For highly absorbing specimen with radius  $r$ , diffraction is limited to a cylindrical surface layer resulting in a maximum peak shift:

$$\Delta 2\mathbf{q} = \frac{r}{R} \cos \mathbf{q} \quad [\text{in radians}] \quad (3.42)$$

i.e., in this case,  $SYCOS = r/R$ . The latter effect also leads to asymmetric line profiles.

### ***Curved position sensitive detector with flat plate sample***

For the asymmetric geometry of diffractometers using a curved position sensitive detector (CPSD) with a flat-plate sample, the displacement correction takes the form:

$$\Delta 2\mathbf{q} = \frac{-e}{R \sin \mathbf{a}} \sin 2\mathbf{q} \quad [\text{in radians}] \quad (3.43)$$

where  $R$  is the radius of the CPSD and  $\mathbf{a}$  the incident beam angle (in degrees) at sample surface. Thus, the parameter refined by **FullProf** is  $SYSIN = -s/(R \sin \mathbf{a})$ . The negative sign means, as in the case of Bragg-Brentano geometry, that a displacement away from the center of the focusing circle moves the diffraction lines to lower  $2\mathbf{q}$  angle; the value of the sample offset is given by:

$$s = \frac{P}{180} R \sin \mathbf{a} SYSIN$$

## Crystal structure factors

The crystallographic structure factor  $F_{\mathbf{h}}$  is calculated in **FullProf** by using the formula:

$$F_{\mathbf{h}} = \sum_{j=1}^n O_j f_j(\mathbf{h}) \exp(-B_j |\mathbf{h}|/4) \sum_{s=1}^m T_{js}(\mathbf{h}) \exp\{2\pi i (\mathbf{h}^T S_s \mathbf{r}_j + \mathbf{h}^T \mathbf{t}_s)\} \quad (3.44)$$

where  $n$  is the number of atoms in the asymmetric unit,  $m$  is the number of the reduced set of symmetry operators (centring lattice translations and inversion center operators removed).  $O_j$  is the occupation factor,  $f_j(\mathbf{h})$  is the scattering length (in electron units for X-rays, in  $10^{-12}$  cm for neutrons),  $B_j$  is the isotropic temperature parameter given in  $\text{\AA}^2$ , and  $\mathbf{r}_j$  is the position vector of atom  $j$ . The symmetry operator  $\{S_s | \mathbf{t}_s\} = \{S | \mathbf{t}\}_s$  is applied to the scattering vector  $\mathbf{h}$  (treated here as a column matrix) instead of the position vector.  $\mathbf{h}^T$  is a row matrix with elements  $(hkl)$ . The temperature factor  $T_{js}(\mathbf{h})$  is given by:

$$T_{js}(\mathbf{h}) = \exp\{-\mathbf{h}^T S_s \mathbf{b}_j S_s^T \mathbf{h}\} \quad (3.45)$$

where  $\mathbf{b}_j$  is the symmetrical matrix representing the anisotropic temperature parameters of the atom  $j$ . For the first atom position ( $S_{s=1} = I$ ) the expression of the anisotropic temperature factor is the well-known expression:

$$T(hkl) = \exp\{-(\mathbf{b}_{11}h^2 + \mathbf{b}_{22}k^2 + \mathbf{b}_{33}l^2 + 2\mathbf{b}_{12}hk + 2\mathbf{b}_{13}hl + 2\mathbf{b}_{23}kl)\} \quad (3.46)$$

where the subscript  $j$  has been dropped for simplicity. Of course, if the isotropic temperature factor  $T_j(iso) = \exp(-B_j \sin^2 \theta / \lambda^2)$  is used the parameters  $\mathbf{b}_{abj}$  should be set to zero. The  $\mathbf{b}$ 's have no direct physical meaning. They have to be converted to a Cartesian  $\mathbf{U}$ -matrix that provides, after diagonalisation, the r.m.s (root mean squares) atom displacements (see [17]). **FullProf** does this work at the end of the refinement, the results are given in the file CODFIL.out.

Apart from the standard scattering factor for individual atoms existing in an internal library of **FullProf**, the version 3.2 and higher can handle complex form-factors as a standard option. In the general expression of the nuclear structure factor (see (3.44)) the form factor  $f_j(\mathbf{h})$  is normally dependent on the module of  $\mathbf{h}$ . For molecular plastic crystals the treatment of rotating molecules cannot be done using an atomic description. The approach of a molecular form-factor that takes into account the particular dynamics of the object is more reliable.  $f_j(\mathbf{h})$  depends on a series of parameters for different types of objects. Coefficients of Symmetry Adapted Spherical Harmonics, geometrical parameters (radius of a sphere, length and radius of a cylinder or disk, etc), scattering density, for instance, may serve for describing the scattering factor of a complex object. In the present version of **FullProf** the available (or projected) objects are the following: spherical shell (SPHS), dense sphere (SPHE), ellipsoid (ELLI), cylinder/disk of elliptical section (DISK), etc.

For the calculation of the integrated intensities (3.2) two types of factors are applied to  $F_{\mathbf{h}}^2$ . One is an overall temperature factor  $\left(\exp\{-B_{ov} |\mathbf{h}|/2\}\right)$  and the other is the constant:  $NLAT = (LAT \cdot ICEN)^2$ , where  $LAT$  is the number of centring translations (1 for P, 2 for A,B,C and I, 3 for R and 4 for F) and  $ICEN=2$  for centrosymmetric and  $ICEN=1$  for non-centrosymmetric space groups.

## Magnetic scattering calculations

For a magnetic phase  $F_h^2$  is calculated using the general formula of Halpern and Johnson:

$$F_h^2 = |\mathbf{F}_\perp(\mathbf{h})|^2 = |\mathbf{F}_m(\mathbf{h})|^2 - (\mathbf{e} \cdot \mathbf{F}_m(\mathbf{h}))^2 \quad (3.47)$$

where  $\mathbf{F}_m(\mathbf{h})$  is the magnetic structure factor, and  $\mathbf{e}$  is the unit vector along the scattering vector  $\mathbf{h}$ . Here the scattering vector is written as  $\mathbf{h} = \mathbf{H} + \mathbf{k}$  where  $\mathbf{H}$  is a reciprocal lattice vector of the crystal structure and  $\mathbf{k}$  the propagation vector corresponding to the current magnetic reflection. The magnetic structures that can be refined with **FullProf** must have a distribution of magnetic moments that can be expanded as a Fourier series:

$$\mathbf{m}_{lj} = \sum_{\{\mathbf{k}\}} \mathbf{S}_{\mathbf{k}j} \exp\{-2\pi i \mathbf{k} \mathbf{R}_l\} \quad (3.48)$$

In such a case the magnetic structure factor is given by:

$$\mathbf{F}_m(\mathbf{H} + \mathbf{k}) = p \sum_{j=1}^{n_c} f_j(\mathbf{H} + \mathbf{k}) \mathbf{S}_{\mathbf{k}j} \exp\{2\pi i (\mathbf{H} + \mathbf{k}) \mathbf{r}_j\} \quad (3.49)$$

The sum is over *all the atoms* in the crystallographic cell. The constant  $p = (r_g/2)$  is 0.2695 and allows the conversion of the Fourier components of magnetic moments, given in Bohr magnetons  $\mathbf{m}_b$  to scattering lengths units of  $10^{-12}$  cm.  $f_j(\mathbf{H} + \mathbf{k})$  is the magnetic form factor and  $\mathbf{r}_j$  is the vector position of atom  $j$ . In the above expression the atoms have been considered at rest. If thermal motion is considered and if symmetry relations are established for coupling the different Fourier components  $\mathbf{S}_{\mathbf{k}j}$ , we obtain the general expression of the magnetic structure factor used in **FullProf** for the reflection with scattering vector  $\mathbf{h} = \mathbf{H} + \mathbf{k}$ :

$$\mathbf{F}_m(\mathbf{h}) = p \sum_{j=1}^n O_j f_j(\mathbf{h}) T_j(iso) \sum_s M_{js} \mathbf{S}_{\mathbf{k}j} T_{js} \exp\{2\pi i [\mathbf{h} \{S|\mathbf{t}\}_s \mathbf{r}_j - \mathbf{y}_{\mathbf{k}js}]\} \quad (3.50)$$

The sum over  $j$  concerns the atoms of the magnetic asymmetric unit for the wave vector  $\mathbf{k}$  (the Fourier component with index  $\mathbf{k}$  contributes only to the  $\mathbf{k}$ -satellite). The sum over  $s$  concerns the different symmetry operators of the crystal space group that belong to the wave vector group. The matrix  $M_{js}$  transform the components of the Fourier term  $\mathbf{S}_{\mathbf{k}j}$  of the starting atom  $j1$  to that numbered as  $js$  in the orbit of  $j$ . The anisotropic temperature factor,  $T_{js}$ , is simplified in the calculation to only diagonal terms ( $\mathbf{b}_{aa}$ ), but it is not generally necessary to use these terms in magnetic refinements. The phase factor  $\mathbf{y}_{\mathbf{k}js}$  has two components:

$$\mathbf{y}_{\mathbf{k}js} = \Phi_{\mathbf{k}j} + \mathbf{f}_{\mathbf{k}js} \quad (3.51)$$

$\Phi_{\mathbf{k}j}$  is a phase factor that is not determined by symmetry. It is a refinable parameter and it is significant only for an independent set of magnetic atoms with respect to another one.  $\mathbf{f}_{\mathbf{k}js}$  is a phase factor determined by symmetry. The Fourier component  $\mathbf{k}$  of the magnetic moment of atom  $j1$ ,  $\mathbf{S}_{\mathbf{k}j}$ , is transformed to

$$\mathbf{S}_{\mathbf{k}js} = M_{js} \mathbf{S}_{\mathbf{k}j} \exp\{-2\pi i \mathbf{f}_{\mathbf{k}js}\} \quad (3.52)$$

The sign of  $\mathbf{f}_{\mathbf{k}js}$  changes for  $-\mathbf{k}$ . The reflection  $\mathbf{H} + \mathbf{k}$  has the *negative* sign indicated in the above formulae and the reflection  $\mathbf{H} - \mathbf{k}$  has the positive sign.

In the general case  $\mathbf{S}_{kj}$  is a complex vector (in general there are six components. The magnetic phases are given in fractions of  $2\mathbf{p}$ . For the scattering vector  $\mathbf{H}-\mathbf{k}$  the Fourier component is the complex conjugate of that used for calculating the structure factor for  $\mathbf{H}+\mathbf{k}$ . The program takes into account automatically this fact. If  $\mathbf{k}$  is at the interior of the Brillouin Zone a factor 1/2 is applied to the Fourier coefficient. Let us consider a single index  $j$  for all the magnetic atoms in the unit cell as in (3.50). The Fourier coefficient for the sublattice  $j$  is explicitly given by:

$$\mathbf{S}_{kj} = \frac{1}{2} \left\{ R_{xj} \mathbf{e}_1 + R_{yj} \mathbf{e}_2 + R_{zj} \mathbf{e}_3 + i \left[ I_{xj} \mathbf{e}_1 + I_{yj} \mathbf{e}_2 + I_{zj} \mathbf{e}_3 \right] \right\} \exp \{ -2\mathbf{p}i\mathbf{f}_{kj} \} \quad (3.53)$$

The vector  $-\mathbf{k}$  must also be given either explicitly or implicitly by giving NVK < 0 (see NVK on Line 11-2). If NVK < 0 the program applies the factor 1/2 because it is supposed that  $\mathbf{k}$  is non equivalent to  $-\mathbf{k}$  even if  $\mathbf{k}$  belong to the surface of the Brillouin zone. If the option JHELIX=1 is used, the number of free parameters per magnetic atom is reduced. The Fourier coefficients are considered of the form:

$$\mathbf{S}_{kj} = \frac{1}{2} \left[ m_{1j} \mathbf{u}_j + im_{2j} \mathbf{v}_j \right] \exp \{ -2\mathbf{p}i\mathbf{f}_{kj} \} \quad (3.54)$$

where  $\mathbf{u}_j$  and  $\mathbf{v}_j$  are orthogonal unit vectors. If  $m_{1j} = m_{2j}$  the magnetic structure for the sublattice  $j$  corresponds to a classical helix (or spiral) of cylindrical envelope. All  $j$  atoms have a magnetic moment equal to  $m_0$ . If  $m_{1j} \neq m_{2j}$  the helix has an elliptical envelope and the moments have values between  $\min(m_{1j}, m_{2j})$  and  $\max(m_{1j}, m_{2j})$ . If  $m_{2j} = 0$  the magnetic structure corresponds to a modulated sinusoid of amplitude  $A = m_{1j}$ .

In general, the user has to calculate the real magnetic moments from the refined values of the Fourier components: the term "Magnetic Moment" in the output file means the modulus of the corresponding Fourier component. The program MOMENT has been written in order to help the user with these calculations. In any case, the calculation of the magnetic moment of the atom  $j$  in the unit cell of index  $l$ , should be done by using the formula:

$$\mathbf{m}_l = \sum_{\{\mathbf{k}\}} \mathbf{S}_{kj} \exp \{ -2\mathbf{p}i\mathbf{k}\mathbf{R}_l \}$$

$$\mathbf{m}_l = \sum_{\{\mathbf{k}\}} \left\{ \mathbf{R}_{kj} \cos 2\mathbf{p} \left[ \mathbf{k}\mathbf{R}_l + \mathbf{y}_{kj} \right] + \mathbf{I}_{kj} \sin 2\mathbf{p} \left[ \mathbf{k}\mathbf{R}_l + \mathbf{y}_{kj} \right] \right\}$$

where the last sum is extended for half the number of propagation vectors, i.e. over the number of pairs  $(\mathbf{k}, -\mathbf{k})$ .

If the propagation vector  $\mathbf{k}$  is commensurate (rational components) one can use the magnetic unit cell and  $\mathbf{S}_{kj}$  can be identified with the magnetic moment at site  $j$ . In this case it is possible to describe the magnetic structure with  $\Psi_{kj} = 0$  and  $\mathbf{Q} = \mathbf{H}$ , being  $\mathbf{H}$  an integer vector of the reciprocal lattice of the magnetic cell. If  $\mathbf{k} = \frac{1}{2}\mathbf{H}$ , the use of the chemical unit cell and real magnetic moments is also possible. In such a case only one propagation vector is needed: if NVK is given as negative the generation of magnetic reflections could be in error. For centred crystallographic unit cells one can use only the content of a primitive cell and generate the satellites from the symbol of the centring followed by -1 (e.g. I -1 for a I-centred cell). In order to take the advantage of the crystallographic conventions (propagation vector given with respect to the reciprocal basis of the conventional cell) one can use the dimensions and the metrics of the conventional cell provided that, putting the content of a primitive cell in the conventional cell frame, the occupation factors are multiplied by the number of centring vectors. See the files **hob\*.pcr** in the example directory of the anonymous FTP-area.

## Codewords

Codewords  $C_x$  they are entered for each refined parameter. A zero codeword means that the parameter is not being refined. For each refined parameter, the codeword is formed as:

$$C_x = \text{sign}(a) (10p + |a|) \quad (3.55)$$

where  $p$  specifies the ordinal number of the parameter  $x$  (i.e.  $p$  runs from 1 to MAXS) and  $a$ (multiplier) is the factor by which the computed shift will be multiplied before use.

The calculated shifts are also multiplied by a relaxation factor before being applied to the parameters.

## Standard deviations

Standard deviations are estimated from the formula:

$$s_i = |a| \cdot [c^2 M_{ii}]^{1/2} \quad (3.56)$$

where  $a$  is the coefficient of the codeword for the parameter,  $M_{ii}$  is the corresponding diagonal element in the inverted matrix. The  $c^2$  index used in the above formula is always calculated for the points in the pattern having Bragg contributions, thus  $s_i$  could be greater than the corresponding value calculated with other programs. As  $c^2$  is calculated in two ways (see below) the user can easily calculate the other value.

## Method of refinement

**Least squares refinement (the standard method)**

$$M_p = \sum_i \left[ w_i (y_{obsi} - y_{calci})^2 \right] \quad (3.57)$$

the weights of the observations are calculated as:  $w_i = 1/\text{var}_{obsi}$

**Maximum likelihood refinement**

The weights of the observations are calculated at each cycle as:  $w_i = 1/\text{var}_{calci}$

## Agreement factors

The quality of the agreement between observed and calculated profiles is measured by a set of nowadays-conventional factors. In **FullProf** two sets of indices are calculated, according to the meaning of the integer N. In the first set N is the total number of points used in the refinement (N=NPTS-NEXC = total number of points in the pattern total number of excluded points). In the second set only those points where there are Bragg contributions are taken into account. The definition of the indices is as follows:

Profile Factor:

$$R_p = 100 \sum_i |y_{obsi} - y_{calci}| / \sum_i |y_{obsi}| \quad (3.58)$$

Weighted Profile Factor:

$$R_{wp} = 100 \left[ \sum_i w_i |y_{obsi} - y_{calci}|^2 / \sum_i w_i |y_{obsi}|^2 \right]^{1/2} \quad (3.59)$$

Bragg Factor:

$$R_B = 100 \sum_k |I_{obs,k} - I_{calc,k}| / \sum_k |I_{obs,k}| \quad (3.60)$$

The expected R Factor:

$$R_{exp} = 100 \left[ (N - P + C) / \sum_i w_i y_{obs,i}^2 \right]^{1/2} \quad (3.61)$$

The goodness of fit Factor:

$$c^2 = [R_{wp} / R_{exp}]^2 \quad (3.62)$$

where N-P+C is the number of degrees of freedom (The meaning of N has been given above, P the number of refined parameters and C the number of strict constraint functions).

Excluded regions are always excluded from the calculation of all agreement factors. Conventional Rietveld R-Factors:  $cR_p$ ,  $cR_{wp}$  are calculated as above but using background-corrected counts. The magnetic R-factor is defined as the Bragg  $R_B$ -factor but is applied to magnetic intensities.

The "observed" integrated intensity  $I_{obs,k}$  is in fact calculated from the Rietveld formula:

$$I_{obs,k} = I_{calc,k} \sum_i \left\{ \Omega(T_i - T_k)(y_{obs,i} - B_i) / (y_{calc,i} - B_i) \right\} \quad (3.63)$$

This formula is equivalent to a "proportional sharing" of the integrated intensity of a cluster between its components according to the actual model. Then, if the model contains a strictly zero integrated intensity for the component "k" ( $I_{calc,k} = 0$ ), the observed integrated intensity is also zeroed:  $I_{obs,k} = 0$ , even if it is obvious that

$I_{obs,k}$  is non zero from the experimental pattern. This has as a consequence that the reflections with  $I_{calc,k} = 0$  do not contribute to the Bragg R-factor.

Although commonly used in crystallography, the  $R_p$ ,  $R_w$ ,  $R_B$  agreement factors are not satisfactory from a statistical point of view. Therefore, a number of statistically more significant parameters are calculated by **FullProf**:

- The deviance [15] defined as:

$$D = 2 \sum_i \left\{ y_{obs,i} \ln(y_{obs,i} / y_{calc,i}) - (y_{obs,i} - y_{calc,i}) \right\} \quad (3.64)$$

- From the deviance, one can derive two other measures of discrepancy which are useful as model selection criteria (somewhat analogous to Hamilton's criterion). These criteria take account of both the goodness of fit of a model and of the number of parameters used to achieve that fit. They take the form:

$$Q = D + a \cdot maxs \quad (3.65)$$

where *maxs* is the number of refined parameters and *a* represents the "cost" of fitting an additional parameter. Akaike's information criterion uses *a*=2 while Schwarz's criterion has *a*=ln(maxs).

- The Durbin-Watson statistic parameters: *d* and *Q*. The use of these two quantities to assess the quality of the refinement has been advocated by Hill and Flack [16]. This statistic which measures the correlation between adjacent residuals (serial correlation) is defined as:

$$d = \sum_{i=2}^N \left\{ \left[ w_i (y_{obs,i} - y_{calc,i}) - w_{i-1} (y_{obs,i-1} - y_{calc,i-1}) \right]^2 \right\} / \sum_i \left\{ \left[ w_i (y_{obs,i} - y_{calc,i}) \right]^2 \right\} \quad (3.66)$$

Serial correlation is tested (at the 99.9% confidence level) by comparing the value of *d* to that of *Q* that is given by the relation:



$$Q = 2 \left\{ (N-1)/(N-P) - 3.0901/\sqrt{N+2} \right\} \quad (3.67)$$

Three cases may occur:

- 1) If  $d < Q$ , there is positive serial correlation: successive values of the residuals tend to have the same sign. This is the most common situation in profile refinement.
- 2) If  $Q < d < 4-Q$ , there is no correlation
- 3) If  $d > 4-Q$ , there is negative serial correlation: successive values of the residuals tend to have opposite sign.

## Analysis of the refinement

An empirical analysis of the refinement has been introduced at the end of the file CODFIL.SUM. A part from some subjective comments that could appear in that analysis there are some important quantities that have to be known by the users because they have not been published yet.

- Expected  $R_p$  factors are calculated supposing the best possible model.

$$R_p = 100 \sum_i |y_{obsi} - y_{calci}| / \sum_i |y_{obsi}| \quad (3.68)$$

where  $y_{calc,i}$  is calculated with the help of a Poissonian function,  $Genpoi(x)$ , from  $y_{obsi}$ . The argument of Genpoi is an integer value representing an observation which is equal to their variance, Genpoi returns another possible value compatible with the deterministic value  $x$  of variance  $x$ . The value of  $x$  is calculated from  $y_{obsi}$  and  $var_{obsi}$  as follows:

$$c_1 = y_{obsi} / var(y_{obsi})$$

$$x = y_{obsi} c_1$$

$$y_{calci} = Genpoi(x) / c_1$$

- The percentage of the contribution to the total integrated intensity. (Sum(Iobs) for all phases) of each phase is now written.
- The number of refined parameters distributed in three classes are written:  
*Nglb*: Number of global parameters (not depending of the phase index)  
*Nprofp*: Number of profile parameters. the meaning of the integer  
*Nintdp*: Number of intensity-dependent parameters ( $x, y, z, B, occ, Mx...$ ). The preferred orientation parameters are included in this class.
- An effective number of reflections is calculated in order to get the ratio  $Refni = (\text{Effective Number of Reflections}) / Nintdp = Enref / Nintdp$ .
- The concept of effective number of reflections is introduced in order to take into account the effect of the resolution in the refinement. It is clear that well separated independent reflections give better results than when the reflections are overlapped. The program calculates a global effective number of reflections. For each phase, a similar indicator is written.

A reflection contributes as  $x/(x+nearest)$ , where " $x$ " is the fraction of the total area of the current phase and " $nearest$ " is the number of adjacent reflections verifying the formula:

$$2\theta - p \cdot FWHM \leq 2\theta(\text{adjacent}) \leq 2\theta + p \cdot FWHM$$

"nearest" is weighted by the corresponding " $x(s)$ ", and  $p$  is a parameter lower than unity.

The general formula for calculating the global effective number of reflections is:

$$Enref = \text{Sum}(i) \{ x[ph:i] / \text{Sum}(ni) \{ x[ph:ni] \} \}$$

where  $\text{Sum}(i)$  is a sum over all the reflections contributing to the allowed areas of the diffraction pattern.  $\text{Sum}(n_i)$  is the sum extended to the reflections near to the reflection "i" (including this reflection). The symbol  $x[\text{ph}:i]$  is the "x"-value of the phase to which the reflection "i" belongs. The same formula restricted to reflections of a single phase is applied to calculate  $\text{Enref}(\text{Iphase})$ : Effective number of reflections of the phase  $\text{Iphase}$ . The program calculates these numbers ( $\text{Enref}$ 's) and the ratios ( $\text{Refni}$ 's) for three values of the parameter  $p$  (1,  $1/2$  and  $1/4$ ).