

***AN INTRODUCTION TO THE
PROGRAM***

FullProf

2000

(Version July2001)

**Juan Rodríguez-Carvajal
Laboratoire Léon Brillouin (CEA-CNRS)
CEA/Saclay, 91191 Gif sur Yvette Cedex, FRANCE
e-mail: juan@llb.saclay.cea.fr**

Preface

This PDF document is the first serious attempt to a manual of the program *FullProf*. The manual is not yet totally finished and the author apologises for the errors it contains. The description of the main control file **CODFIL.pcr** is detailed in an appendix. A substantial part of the document is dedicated to the treatment of examples and the description of specialised problems. A beginner cannot start to use the program without any background in crystallography, magnetism, diffraction physics, and data analysis. Even an expert in these fields can experience difficulties the first time (or even the n-th time!) he (she) uses the program. An effort is presently being developed to facilitate the use of *FullProf*. For the moment only the new version *FullProf 2000* under Windows 9x/2k/NT, which is distributed together with *WinPLOT*, looks like a modern user-friendly application. A Linux version of the program *WinPLOT* will be prepared in order to provide an efficient Graphic User Interface (GUI) to *FullProf*. At present a platform independent GUI (written in JAVA) exist but it is still in development and the performance is still low.

The manual begins with a short description of the way to obtain *FullProf* from the anonymous **ftp-area**. It follows with a chapter where a brief description of the purpose of the program and the list of input and output files. The second chapter is dedicated to the description of strategies for using the program and the description of examples. The best way to start is using examples that work properly if the good parameters are provided to the program. To that end a kit of examples has been installed in an accessible area of the server *charybde.saclay.cea.fr*. This file (*pcr_dat.zip*) must be *unzipped* with the well-known WinZip program or a compatible product. The format of the files is for a PC, but they can be converted easily to UNIX with an editor or with a simple application like *dos2unix*. The Windows 9x/2k/NT version of *FullProf* is distributed in a single ZIP file containing the examples kit inside. In the third chapter the mathematical expressions used inside *FullProf* are written and discussed briefly. The rest of chapters are dedicated to the discussion of some specialised problems.

Juan Rodríguez-Carvajal
Saclay, July 2001

Disclaimer

The program **FullProf** is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY of being free of internal errors. In no event will the author be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs). The author is not responsible for erroneous results obtained with **FullProf**. This manual cannot substitute the lack of knowledge of users on crystallography, magnetism, diffraction physics, and data analysis. Powder diffraction is becoming more and more powerful but **FullProf** is not an *automatic* (black-box) program, as is usually found in single crystal structure determination. No attempt has been made in order to predict the behaviour of the program against bad input data. The user must check his (her) data before claiming a malfunction of the program. The author acknowledges all suggestions and notification of possible bugs found in the program.

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, Windows9x/NT)
ftp>cd windows                ! Go to Windows9x/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**. 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 treated 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 used for this task. On the PC-world a very useful program is **WinPLOTTR** (written by Thierry Roisnel in collaboration with the author at the LLB). **WinPLOTTR** can be obtained in the same **ftp-area** as **FullProf** in the directory *pub/divers/winplotr*. The program **WinPLOTTR** is also distributed with the Windows 9x/2k/NT version of **FullProf**. So users working with Windows 9x/2k/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** / **WinPLOTTR** 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 WINPLOTTR must also be created in order to use **FullProf** / **WinPLOTTR** without troubles. This may be done by inserting the following lines in the file *autoexec.bat* (normally this file is in c:\, in Windows NT it may be non-existent) :

```
SET WINPLOTTR=d:\My_FullProf_dir
SET FULLPROF=d:\My_FullProf_dir
Path=%Path%;d:\My_FullProf_dir
```

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 (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 2θ . 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 read 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 contents 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 2θ 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 of 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 1.9c - May2001-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 contents the diffraction conditions and crystallographic information needed by the program. The optional file **FILE.dat** (or **FILE.uxd**) contents 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/2k/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 contents 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 (written in blue) are done without explicit explanations; this means that they are explained in the appendix, where it is described in detail the content of the input files. The logical unit associated with each file is given to help the user in the case of runtime errors. Sometimes we shall give references to line numbers written in bold blue corresponding to labels of the different items in the input control file that is described in the appendix.

CODFIL.pcr (logical unit: **i_pcr** = 1)

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 commands¹. 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.inf**.

The following files are **optional**:

FILE.dat (logical unit: **i_dat** = 4)

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) (logical unit: **i_bac** = 12)

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 **Ins=0**:
 - First line: 2 θ /TOF/Energy (initial) step 2 θ /TOF/Energy (final), any comment
 - Rest of lines: list of intensities in free format.
2. The second format is adapted to the case where 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.

¹ This last format is not available at present

CODFILn.hkl or **hkln.hkl** (logical unit: **i_hkl** = 15)

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.

MYRESOL.irf (logical unit: **i_res** = 13)

File describing the instrumental resolution function. Any legal filename can be used and its content depends on the value of the parameter [Res](#).

global.shp or **CODFIL.shp** (logical unit: **i_shp** = 24)

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

CODFIL.cor (logical unit: **i_cor** = 25)

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.

CODFIL.int (logical unit: **i_int** = 26)

Single integrated intensity file when the program is used for refining with [Cry=1, 2, 3](#) and [Irf=4](#).

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 (logical unit: **i_out** = 7)

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** (logical unit: **i_prf** = 17) ([Prf](#) 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 (logical unit: **i_rpa** = 2) ([Rpa](#)=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 (logical unit: **i_sym** = 3) ([Syo=Sym=1](#))

List of symmetry operators

CODFIL.sum (logical unit: **i_sum** = 8)

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 [Ana=1](#).

CODFIL.fou (logical unit: **i_fou** = 9)

([Fou=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.

([Fou=2](#))

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

([Fou= -1](#) or [-2](#))

As above but the structure factors are calculated in another way. The F_{calc} in [Fou](#)>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 [Fou](#) is negative, F_{calc} are really the structure factors of the conventional cell in absolute units.

([Fou=3](#))

Format suitable for the program FOURIER

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

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.

([Fou=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 (logical unit: **i_shx** = 21) ([Fou=2](#))

Template of the input control file for the program SHELXS.

CODFILn.inp (logical unit: **i_shx** = 21) ([Fou=3, 4](#))

Template of (G)FOURIER *.inp file.

CODFILn.hkl (logical unit: **i_shkl** = 16)

Files that can be input or output files. The content depends on the value of [Irf\(n\)](#)

CODFIL.hkl (logical unit: **i_ghkl** = 10)

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

- [Hkl=1](#)
 - If [Job](#) < 2
Code, h, k, l , *mult*, d_{hkl} , 2θ , FWHM, I_{obs} , I_{calc} , $I_{obs} - I_{calc}$
 - If [Job](#) > 1
 h, k, l , *mult*, I_{calc} , 2θ , d_{hkl}
- [Hkl=2](#)
Output for EXPO
 h, k, l , *mult*, $\sin \theta / \lambda$, 2θ , FWHM, F^2 , $\sigma(F^2)$
- [Hkl=3,-3](#)
Output of real and imaginary part of structure factors (only for crystal structures)
 h, k, l , *mult*, F_{real} , F_{imag} , 2θ , Intensity

If [Hkl](#) 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.

- [Hkl=4](#)
Output of: $h, k, l, F^2, \sigma(F^2)$.
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 [Cry=1](#) and [Irf=4](#).
- [Hkl=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 recalculating the structure factors for each cycle. The F_{calc} values are given in absolute units for the conventional unit cell.

CODFIL.sav (logical unit: **i_sav** = 11) ([Rpa=2](#))

List of reflections between two selected angles. This file is output only if an interval in the scattering variable is given.

h, k, l , *mult*, I_{obs} , 2θ , d_{hkl}

CODFILn.dis (logical unit: **i_dis** = 28) ([JDIS=3](#))

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

DCONSTRn.hlp (logical unit: **i_cons** = 99) ([JDIS=3](#))

List of strings containing eventual distance and angle constraints for phase **n**. This file with fixed name, **DCONSTR**, is generated automatically when the user asks for angle/distance calculations. The user may edit this file and select the wished items, modify and paste them to the input control file **CODFIL.pcr** in order to provide soft constraints on distance and angles in a subsequent run.

CODFILn.mic (logical unit: **i_mic** = 29) ([Res ≠ 0](#))

File containing microstructural information. The use of a resolution file, as well as the profile function [Npr=7](#), is imperative to obtain this file.

CODFIL.sim (logical unit: **i_sim** = 14) (abs([Job](#)) > 1 and [Dum](#) = 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 (logical unit: **i_sub** = 23) ([lpr](#)=2, 3)

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

CODFILn.atm (logical unit: **i_atm** = 22) ([More](#)=1 and [Jdi](#)= ±1, 2)

For [Jdi](#)=±1 it is supposed that a magnetic phase is concerned ([Jbt](#)= ±1, 5, 10).

If [Jdi](#)=+1 the files contain the list of magnetic atom positions within a primitive unit cell corresponding to the phase **n**.

If [Jdi](#)=-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 [Jdi](#)= 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 (logical unit: **i_sch** = 18) ([More](#)=1 and [Jvi](#)=1, 2)

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

CODFILn.int (logical unit: **i_int** = 26) ([Jbt](#)=2, [More](#)=1 and [Jvi](#)=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*.

Getting started

For starting a profile refinement from the scratch, the best is to copy one of the PCR files accompanying the distribution of *FullProf*, and modify it according to the user's case: x-ray or neutron diffraction, crystal or magnetic structure refinement, synchrotron, TOF neutrons, etc. The provided PCR files can then be used as templates. Of course the closer to the user's case is the initial PCR file the easier is to modify it. An important aspect is the format of the data file that must be correctly given before attempting any kind of refinement.

The least squares method used in Rietveld refinements is described in the mathematical section. The user must be aware of the way he/she can control the refinement procedure: the number of parameters to be refined, fixing parameters, making constraints, etc. The control of the refined parameters is achieved by using *codewords*. These are the numbers C_x that 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:

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

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 (see equation 3.5 in mathematical section) will be multiplied before use.

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

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 [Sycos](#) and/or [Sysin](#) to correct for instrumental or physical 2θ 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 13**) to zero. Set to 0 the flag controlling the automatic assignment of refinement codes (**Aut=0**). 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 (variable called **Wdt**) 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 **Npr=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 times, 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 19** and the number of atomic positions given on lines **lines 25**.

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 ₂ standard	ceo2.dat
Ce2	"	"
Rutana	Conventional X-ray diffraction pattern: Rutile+Anatase	Rutana.dat
Tbbaco	Conventional X-ray diffraction pattern: Tb ₂ BaCoO ₅	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 ₄ 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 ₄ neutron data	Pbso4.dat
PbSO4	Crystal structure refinement of PbSO ₄	"
PbSO4a	Crystal structure refinement of PbSO ₄ (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 ₄	Pb_san.int
C60s	Compares C ₆₀ x-tal data to form-factor SPHS sin(Qr)/Qr	C60.int
C60	Refinement of C ₆₀ 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 ₆ Ge ₆	"
Dyb		"
Dyc		"
Hocu	Refinement of the magnetic structure of Ho ₂ Cu ₂ O ₅ (D1B data)	Hocu.dat
Hobb	Refinement with integrated intensities (Nuc+mag)	Hobb.int
Hob	Montecarlo search for mag. moments in Ho ₂ BaNiO ₅	"
Hobk1	Three different ways of refining the crystal	Hobk.dat
Hobk2	and magnetic structure of Ho ₂ BaNiO ₅	"
Hobk3		"
Cuf1k	Refinement of crystal & magnetic structure of CuF ₂ . Microstructural effects (D1A data)	Cuf1k.dat
Pb_san	Example of Simulated Annealing: solves the structure of PbSO ₄	Pb_san.int
La	Two ways for strain refinement in La ₂ NiO ₄ (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
Ybcacu	YBaCuO with Ca. Data from D1A	Ybcacu.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 ₃	Lamn_3t2.dat
Lamn_pol	TOF data from POLARIS at ISIS on the RT phase of LaMnO ₃	Lamn_pol.dat
Si3n4r	Quantitative phase analysis. Two polymorphs of Si ₃ N ₄ . (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 ₄	Pb_san.int
Maghem	Refinement of Fe ₂ O ₃ -Fe ₃ O ₄ 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 CuK_α 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 Tb₂BaCoO₅ 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 PbSO₄. 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: **ybcacu**, **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 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 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 $Pr_2NiO_{4+\delta}$.

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 La_2NiO_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 Ho_2BaNiO_5 at 1.5K, using different methods and conditions. The user can verify that **hob.pcr** can *solve* the magnetic structure of Ho_2BaNiO_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 $Ho_2Cu_2O_5$. 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.
- **Cuflk**: refinement of the magnetic structure of 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 $DyMn_6Ge_6$. 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 $PbSO_4$ 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).

ADDITIONAL NOTES

Magnetic Refinements

For a commensurate structure two descriptions of the magnetic structure are possible: the general formalism using Fourier components of magnetic moments through the propagation vectors or a crystallographic-like description using the magnetic unit cell. The simplest one, for a non-expert user, is to describe the magnetic structure in the magnetic unit cell. In that case the following important points must be taken into account:

- For magnetic structures described in a magnetic unit cell larger than the crystallographic cell, the coordinates of atoms must be changed consequently, as well as the unit cell parameters. The codes for common (or related) parameter with the crystallographic counterpart must be changed applying the correct multiplier factor. It is worth stressing that codes for cell parameters are actually applied to the "cell constants" (A, B, C, D, E, F) defined by the expression:

$$1/d^2 = A h^2 + B k^2 + C l^2 + D kl + E hl + F hk$$

Therefore, if you are dealing with an orthorhombic structure which has a magnetic structure with propagation vector $\mathbf{k}=(1/2, 0, 0)$ you have to use the magnetic unit cell $2 \times a, b, c$ and if the codes of the crystallographic unit cell a, b, c are for example:

Cell	a	b	c	90	90	90
CodeCell	81.00	91.00	101.00	0	0	0

the corresponding values for the magnetic counterpart are:

Cell	2a	b	c	90	90	90
CodeCell	80.25	91.00	101.00	0	0	0

The reason is that the crystallographic cell constant A is in this case: $A_c=1/a^2$ and the magnetic cell constant is $A_m=1/(2a)^2=0.25A_c$. As explained above, the multiplier factor is applied to shifts of parameters.

- The scale factors of the crystallographic and magnetic parts have to be related in some way in order to get good values of magnetic moments. This relation depends on the way the users describes the magnetic structure, however several rules can be useful to avoid bulk errors:
 1. Use the correct occupation numbers in the crystallographic part (=multiplicity of special position/general multiplicity)
 2. The number of magnetic atoms in the chemical unit cell must coincide with the description above, therefore the occupation numbers in the magnetic part are related to the number of symmetry operators given and the centrosymmetry (or not) of the magnetic structure.

If these requirements are satisfied and the magnetic unit cell is the same as the chemical one, the scale factors are strictly the same numbers; and, therefore represent the same parameter with a shift equal to unity. If the magnetic unit cell is a multiple one, and the above requirements are satisfied, the relation between the scale factors is a multiplier factor relating the two scale factors given by:

$$S_m = 1.0/(V_m/V_c)^2 S_c = (V_c/V_m)^2 S_c$$

Where S_c is the crystallographic scale factor and S_m the magnetic scale factor. In the case of large magnetic cells it can be more convenient to modify the occupation numbers of magnetic atoms in such a way that the two scale factors coincide.

For incommensurate magnetic structures the general formalism must be applied. When the magnetic structure is described using the formalism of propagation vectors, the components M_x , M_y , M_z no longer represent true magnetic moments (see mathematical section). The user should be cautious in interpreting the output files. The modulus of the "magnetic moment" represents the Fourier component modulus of an atomic magnetic moment which have to be calculated externally. The calculation of the intensity is based on the expression of magnetic structure factor given in mathematical section; therefore the user knows how to play with his input items in order to obtain physically sound results.

For the spherical description of the magnetic moments the following must be taken into account:

- The orthonormal system with respect to which are defined the spherical angles verifies:

X axis coincides with the crystallographic a -axis
 Y axis belongs to the plane a - b
 Z-axis is perpendicular to the plane a - b

The particular implementation of spherical components in magnetic structure refinements is that the Z-axis must coincide with c . This works in all crystallographic systems except for triclinic. The monoclinic setting must be changed to the setting $I 1 2/m$ to satisfy the above prescription.

Propagation vectors

A complete list of reflections can be generated when propagation vectors of an incommensurate structure are present. To each fundamental reflection it is added the corresponding satellites. For n propagation vectors $\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_n$, there are n satellites obtained from each fundamental reciprocal lattice vector \mathbf{h} :

Fundamental reflection: $\mathbf{h} = (h, k, l)$
 Satellites $\mathbf{h}_1 = \mathbf{h} + \mathbf{k}_1, \mathbf{h}_2 = \mathbf{h} + \mathbf{k}_2, \dots, \mathbf{h}_n = \mathbf{h} + \mathbf{k}_n$

In the present version of the program no symmetry analysis is performed. We recommend to use the triclinic space group of symbol $L -1$ (where $L = P, A, B, C, F, I, R$) in order to have a full set of reflections with the proper multiplicity when the true magnetic symmetry is not known.

The program generates first a list of unique reflections corresponding to the required space group and then adds the satellites. This method had to be modified for reflections belonging to the boundary planes and lines of the asymmetric region of the reciprocal space in order to obtain the correct number of reflections and not miss (or repeat) some of them. Be careful with propagation vectors \mathbf{k} equivalent to $-\mathbf{k}$! Two vectors \mathbf{k}_1 and \mathbf{k}_2 are "equivalent" if $\mathbf{k}_1 - \mathbf{k}_2$ is a vector of the reciprocal lattice. So, for $\mathbf{k}_1 = -\mathbf{k}_2$, if $\mathbf{H} = 2\mathbf{k}_1$ belongs to the reciprocal lattice, \mathbf{k}_1 is *single* and belongs to a point of high symmetry of the Brillouin Zone. In such cases only ONE propagation vectors should be introduced $N_{\mathbf{v}\mathbf{k}} = 1$, if the user puts $N_{\mathbf{v}\mathbf{k}} = -1$, the satellite reflections are not correctly generated.

For centred cells a propagation vector \mathbf{k} having components $\pm 1/2$, verifies that $2\mathbf{k}$ has integer components, but that does not mean that \mathbf{k} and $-\mathbf{k}$ are equivalent, because $2\mathbf{k}$ could not belong to the reciprocal lattice. For a C lattice the propagation vector $\mathbf{k}_1 = (1/2 \ 0 \ 0)$ is not equivalent to $\mathbf{k}_2 = (-1/2 \ 0 \ 0)$ because $\mathbf{K} = 2\mathbf{k}_1 = \mathbf{k}_1 - \mathbf{k}_2 = (1 \ 0 \ 0)$ does not belong to the reciprocal lattice: $h+k=2n$ is the lattice C condition for components (hkl) . On the contrary, the vector $(0 \ 0 \ 1/2)$ is *single* because (001) is a reciprocal lattice point of the C lattice.

Microstrains and domain size effects. HKL-dependent shifts and asymmetry

The microstructural effects within *FullProf* are treated using the Voigt approximation: both instrumental and sample intrinsic profile are supposed to be described approximately by a convolution of Lorentzian and Gaussian components. The TCH pseudo-Voigt profile function (Thompson, Cox and Hastings, *J. Appl. Cryst.* **20**, 79 (1987)) is used to mimic the exact Voigt function and it includes the Finger's treatment of the axial divergence (L.W. Finger, *J. Appl. Cryst.* **31**, 111 (1998)). The integral breadth

method to obtain volume averages of sizes and strains is used to output a microstructural file where an analysis of the size and strain contribution to each reflection is written. No physical interpretation is given by the program, only a phenomenological treatment of line broadening in terms of coherent domain size and strains due to structural defects is performed. The user should consult the existing broad literature to go further in the interpretation of the results. A recent book (*Microstructure Analysis from Diffraction*, edited by R. L. Snyder, H. J. Bunge, and J. Fiala, International Union of Crystallography, 1999), gathering different articles, is a good introduction to microstructural problems.

The new file containing information about the microstructure is output only if the user provides an input file containing the instrumental resolution function (IRF, see manual for the different ways of giving resolution parameters). At present, this option works only for constant wavelength mode.

The FWHM of the Gaussian (H_G) and Lorentzian (H_L) components of the peak profile have an angular dependence given by:

$$H_G^2 = (U + (1 - \xi)^2 D_{ST}^2(\mathbf{a}_D)) \tan^2 \theta + V \tan \theta + W + \frac{I_G}{\cos^2 \theta}$$

$$H_L = (X + \xi D_{ST}(\mathbf{a}_D)) \tan \theta + \frac{[Y + F(\mathbf{a}_Z)]}{\cos \theta}$$

If the user provides a file with the IRF, the user should fix V and W to zero, then the rest of parameters in the above formula have a meaning in terms of strains (U, \mathbf{a}_D, X) or size (Y, I_G, \mathbf{a}_Z). The functions $D_{ST}(\mathbf{a}_D)$ and $F(\mathbf{a}_Z)$ have different expressions depending on the particular model used of strain and size contribution to broadening. The parameter ξ is a mixing coefficient to mimic Lorentzian contribution to strains.

The anisotropic strain broadening is modeled using a quartic form in reciprocal space. This correspond to an interpretation of the strains as due to static fluctuations and correlations between metric parameters (J. Rodríguez-Carvajal, M.T. Fernández-Díaz and J.L. Martínez, *J. Phys: Condensed Matter* **3**, 3215 (1991)).

$$M_{hkl} = \frac{1}{d_{hkl}^2} = Ah^2 + Bk^2 + Cl^2 + Dkl + Ehl + Fhk = M(\alpha_i; hkl)$$

The metric parameters α_i (direct, reciprocal or any combination) are considered as stochastic variables with a Gaussian distribution characterized by the mean $\langle \alpha_i \rangle$ and the variance-covariance matrix C_{ij} .

Here we consider the set: $\{\alpha_i\} = \{A, B, C, D, E, F\}$. The position of the peaks is obtained from the average value of M_{hkl} given by: $\langle M_{hkl} \rangle = M(\langle \alpha_i \rangle; hkl)$. The broadening of the reflections is governed by the variance of M_{hkl} :

$$\sigma^2(M_{hkl}) = \sum_{i,j} C_{ij} \frac{\partial M}{\partial \alpha_i} \frac{\partial M}{\partial \alpha_j} = \begin{pmatrix} h^2 & k^2 & l^2 & kl & hl & hk \end{pmatrix} \begin{pmatrix} S_A^2 & C_{AB} & C_{AC} & C_{AD} & C_{AE} & C_{AF} \\ C_{AB} & S_B^2 & C_{BC} & C_{BD} & C_{BE} & C_{BF} \\ C_{AC} & C_{BC} & S_C^2 & C_{CD} & C_{CE} & C_{CF} \\ C_{AD} & C_{BD} & C_{CD} & S_D^2 & C_{DE} & C_{DF} \\ C_{AE} & C_{BE} & C_{CE} & C_{DE} & S_E^2 & C_{EF} \\ C_{AF} & C_{BF} & C_{CF} & C_{DF} & C_{EF} & S_F^2 \end{pmatrix} \begin{pmatrix} h^2 \\ k^2 \\ l^2 \\ kl \\ hl \\ hk \end{pmatrix}$$

Where the non diagonal terms may be written as product of standard deviations multiplied by correlation terms: $C_{ij} = S_i S_j \text{corr}(i, j)$. This original formulation can be used with a total control of the correlation terms that must belong to the interval [-1, 1]. When using this formulation the user cannot refine all parameters (up to 21) because some of them contributes to the same term in the quartic form in reciprocal

space, however this allows a better interpretation of the final results. Taking the appropriate caution one can test different degrees of correlation between metric parameters. There are several special formulations, within *FullProf*, for working with direct cell parameters instead of using reciprocal parameters.

Another formulation and a useful notation corresponding to a grouping of terms was proposed by Stephens (P. W. Stephens, *J. Appl. Cryst.* **32**, 281 (1999)) who also included a phenomenological Lorentzian contribution to the microstrains (the parameter ξ). The final grouping of terms simplifies to:

$$\sigma^2(M_{hkl}) = \begin{pmatrix} h^2 & k^2 & l^2 & kl & hl & hk \end{pmatrix} [C] \begin{pmatrix} h^2 \\ k^2 \\ l^2 \\ kl \\ hl \\ hk \end{pmatrix} = \sum_{\substack{HKL \\ \{H+K+L=4\}}} S_{HKL} h^H k^K l^L$$

The Stephens' notation can also be used within *FullProf*. A maximum of 15 parameters can be refined for the triclinic case. Whatever the model used for microstrains the mixing Lorentzian parameter, ξ , may be used. In *FullProf* the function $D_{ST}^2(\mathbf{a}_D)$, being \mathbf{a}_D the set of parameters C_{ij} or S_{HKL} , is given by:

$$D_{ST}^2(\mathbf{a}_D) = 10^{-8} 8 \text{Ln}2 \left(\frac{180}{\pi} \right)^2 \frac{\sigma^2(M_{hkl})}{M_{hkl}^2}$$

An example of anisotropic strain refined using this formulation is shown in Figure 1, where the neutron diffraction pattern of the low temperature phase of Nd_2NiO_4 is refined using the diffractometer D2B at ILL.

Concerning anisotropic size broadening it is possible to use a very general phenomenological model, using the Scherrer formula, that considers the size broadening can be written as a linear combination of spherical harmonics (SPH). At present the anisotropic size is supposed to contribute to the Lorentzian component of the total Voigt function. A Gaussian contribution will be introduced using a mixing

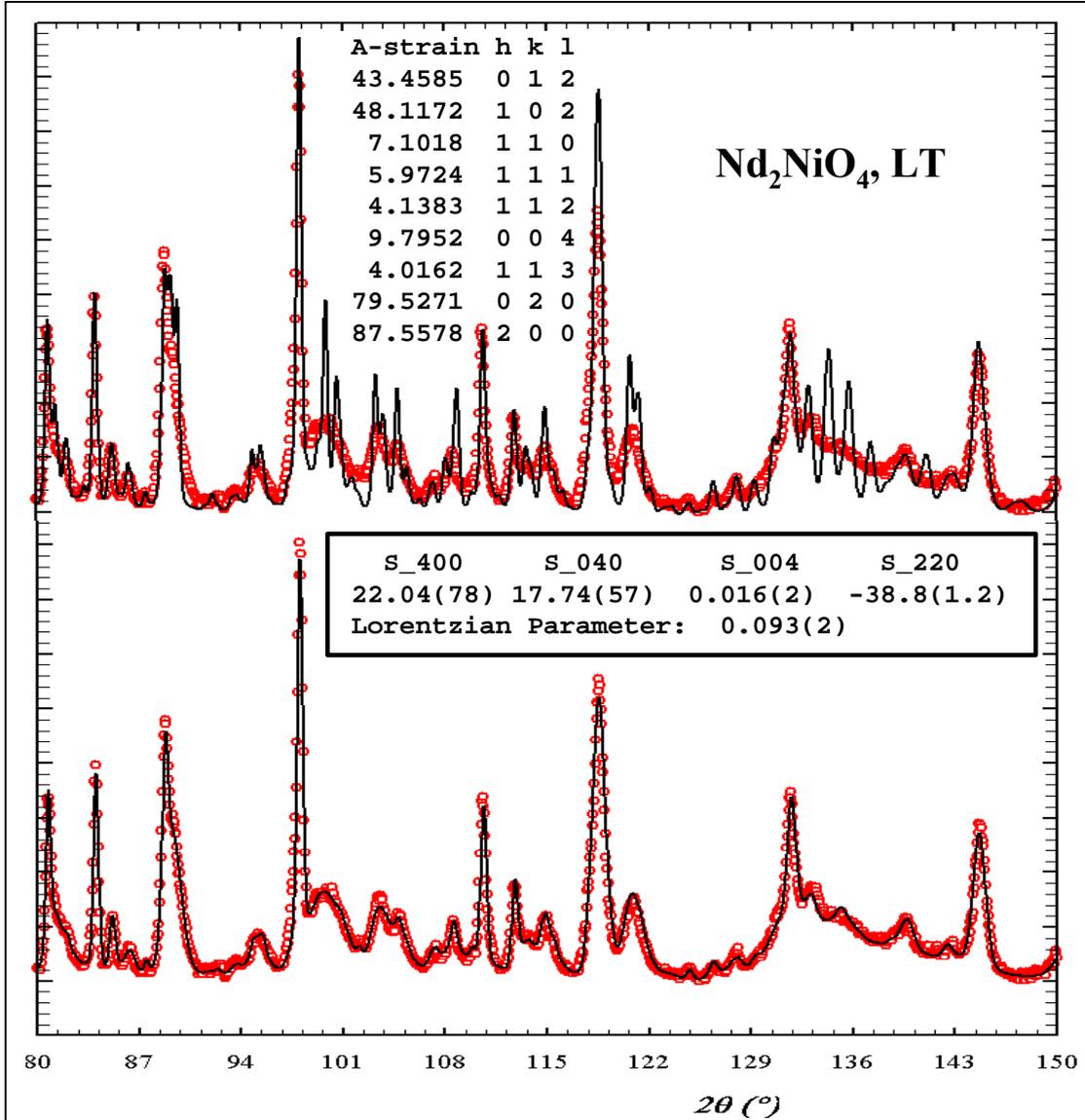
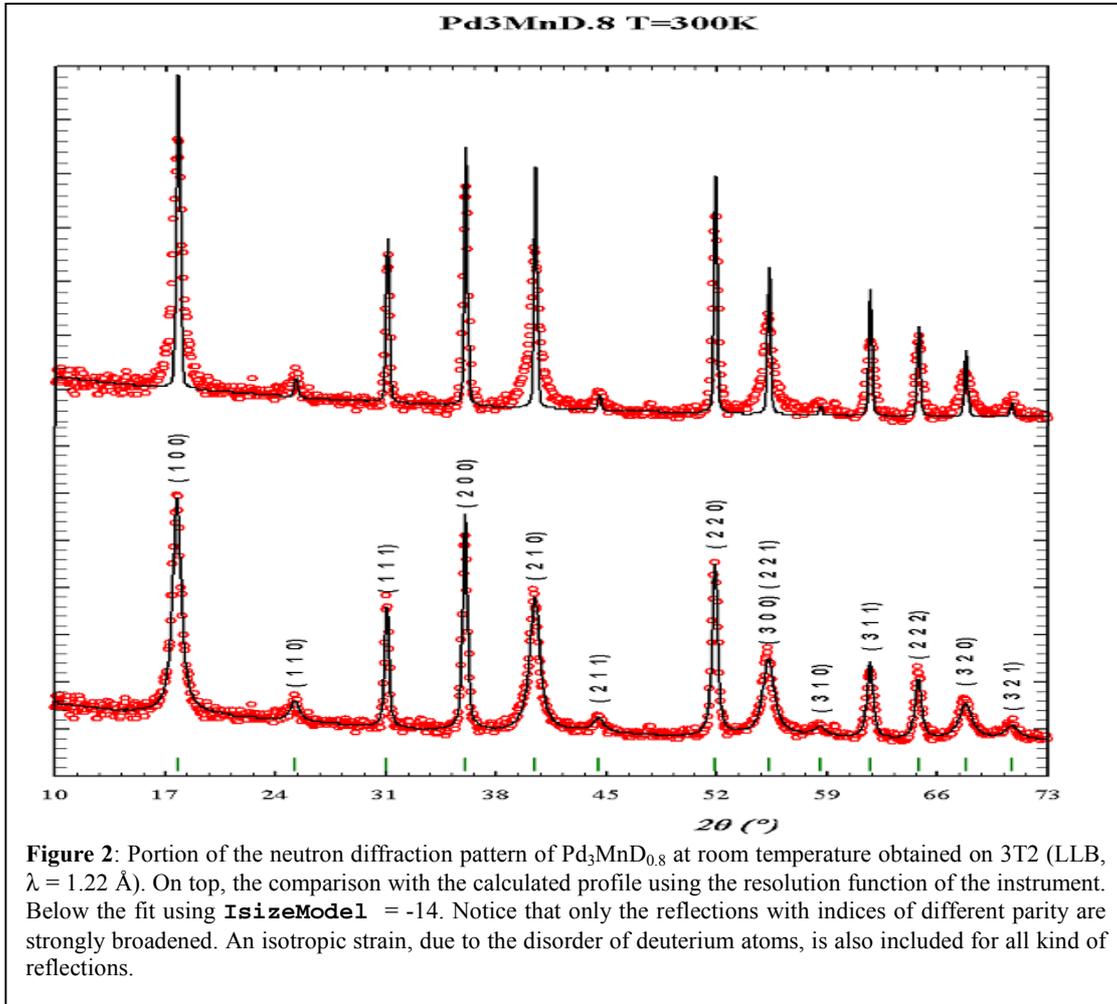


Figure 1: High angle part of the neutron powder diffraction pattern (D2B, ILL) of the low temperature phase of Nd₂NiO₄ [M. T. Fernández-Díaz, M. Medarde and J. Rodríguez-Carvajal (unpublished)]. (top) Comparison of the observed pattern with the calculated pattern using the resolution function of the diffractometer. (bottom) Observed and calculated pattern using an anisotropic model of strains with non-null values given in the panel. A list of apparent strains ($\times 10^{-4}$), extracted from the microstructure file, for a selected number of reflections is also given.

parameter similar to that used for anisotropic strain. The explicit formula for the SPH treatment of size broadening is the following:

$$\beta_h = \frac{\lambda}{D_h \cos \theta} = \frac{\lambda}{\cos \theta} \sum_{lmp} a_{lmp} y_{lmp} (\Theta_h, \Phi_h)$$

Where $\beta_{\mathbf{h}}$ is the size contribution to the integral breadth of reflection \mathbf{h} , $y_{imp}(\Theta_{\mathbf{h}}, \Phi_{\mathbf{h}})$ are the real spherical harmonics with normalization as in [M. Jarvinen, *J. Appl. Cryst.* **26**, 527 (1993)]. The arguments are the polar angles of the vector \mathbf{h} with respect to the Cartesian crystallographic frame. After refinement of the coefficients a_{imp} the program calculates the apparent size (in angstroms) along each reciprocal lattice vectors if the IRF is provided in a separate file.



An important type of defects that give rise to size-like peak broadening is the presence of anti-phase domains and stacking faults. These defects produce selective peak broadening that cannot be accounted using a small number of coefficients in a SPH expansion. In fact only a family of reflections verifying particular rules suffers from broadening. For such cases there is a number of size models built into *FullProf* corresponding to particular sets of reflections that are affected from broadening. In figure 2 it is represented the case of Pd₃MnD_{0.8} [P. Onnerud, Y. Andersson, R. Tellgren, P. Norblad, F. Bourée and G. André, *Solid State Comm.* **101**, 433 (1997)] of structure similar to Au₃Mn and showing the same kind of defects: anti-phase domains [B.E. Warren, "X-ray Diffraction", Dover Publications, Inc., New York, 1990]. In figure 3 a portion of the final microstructural file is shown.

```

! MICRO-STRUCTURAL ANALYSIS FROM FULLPROF (still under development!)
! =====
! Pattern No: 1 Phase No: 1 Pd3MnD.8 - CFC
!
! Integral breadths are given in reciprocal lattice units (1/angstroms)x 1000
! Apparent sizes are given in the same units as lambda (angstroms) ...
! Apparent strains are given in %% (x 10000) (Strain= 1/2 * beta * d)
! An apparent size equal to 99999 means no size broadening
! The following items are output:
!
! The apparent sizes/strains are calculated for each reflection using the formula:
!
! App-size (Angstroms) = 1/(Beta-size)
! App-strain (%%) = 1/2 (Beta-strain) * d(hkl)
!
! (Beta-size) is obtained from the size parameters contributing to the FWHM:
! FWHM^2 (G-size) = Hgz^2 = IG/cos^2(theta)
! FWHM (L-size) = Hlz = ( Y + F(Sz))/cos(theta)
!(Beta-strain) is obtained from the strain parameters contributing to the FWHM:
! FWHM^2 (G-strain) = Hgs^2 = (U+[(1-z)DST]^2) tan^2(theta)
! FWHM (L-strain) = Hls = (X+ z DST) tan(theta)
!
! In both cases (H,eta) are calculated from TCH formula and then
! Beta-pV is calculated from:
!
! beta-pV= 0.5*H/( eta/pi+(1.0-eta)/sqrt(pi/Ln2))
!
! The standard deviations appearing in the global average apparent size and
! strain is calculated using the different reciprocal lattice directions.
! It is a measure of the degree of anisotropy, not of the estimated error
!
... betaG betaL ... App-size App-strain h k l twtet ...
... 1.4817 11.5859 ... 93.58 41.6395 1 0 0 17.7931 ...
... 2.0954 11.9584 ... 93.58 41.6395 1 1 0 25.2665 ...
... 2.5664 1.5573 ... 99999.00 41.6395 1 1 1 31.0743 ...
... 2.9634 1.7982 ... 99999.00 41.6395 2 0 0 36.0343 ...
... 3.3132 12.6973 ... 93.58 41.6395 2 1 0 40.4625 ...
... 3.6294 12.8892 ... 93.58 41.6395 2 1 1 44.5207 ...
... 4.1909 2.5431 ... 99999.00 41.6395 2 2 0 51.8786 ...
... 4.4451 13.3842 ... 93.58 41.6395 3 0 0 55.2849 ...
... 4.4451 13.3842 ... 93.58 41.6395 2 2 1 55.2850 ...
... 4.6855 13.5301 ... 93.58 41.6395 3 1 0 58.5562 ...
... 4.9142 2.9820 ... 99999.00 41.6395 3 1 1 61.7169 ...
... 5.1327 3.1146 ... 99999.00 41.6395 2 2 2 64.7864 ...
... 5.3423 13.9286 ... 93.58 41.6395 3 2 0 67.7802 ...
... 5.5440 14.0510 ... 93.58 41.6395 3 2 1 70.7114 ...
.....

```

Figure 3: Portion of the microstructural file (extension *mic*) corresponding to the fitting of the neutron diffraction pattern in figure 2.

Other models for size broadening in *FullProf* following particular rules for each (*hkl*) are available. Moreover an anisotropic size broadening modeled with a quadratic form in reciprocal space is also available. The expression presently used in *FullProf* is the following:

$$F(\mathbf{a}_z) = k_s d^2 (\alpha_1 h^2 + \alpha_2 k^2 + \alpha_3 l^2 + \alpha_4 kl + \alpha_5 hl + \alpha_6 hk)$$

Where k_s is defined as $k_s = 360/\pi^2 \times \lambda \cdot 10^{-3}$ for the 2θ space and $k_s = 2/\pi \times \text{Dtt1} \cdot 10^{-3}$ for TOF and Energy space. Simple crystallite shapes as infinite platelets and needles (**ISizeModel** = 1, -1 respectively) are also available.

Together with the size broadening models built into *FullProf* and described above, there is another way of fitting independent size-like parameters for different sets of reflections. The user may introduce his(her) own rule to be satisfied by the indices of reflections provided the rule can be written as a linear equality

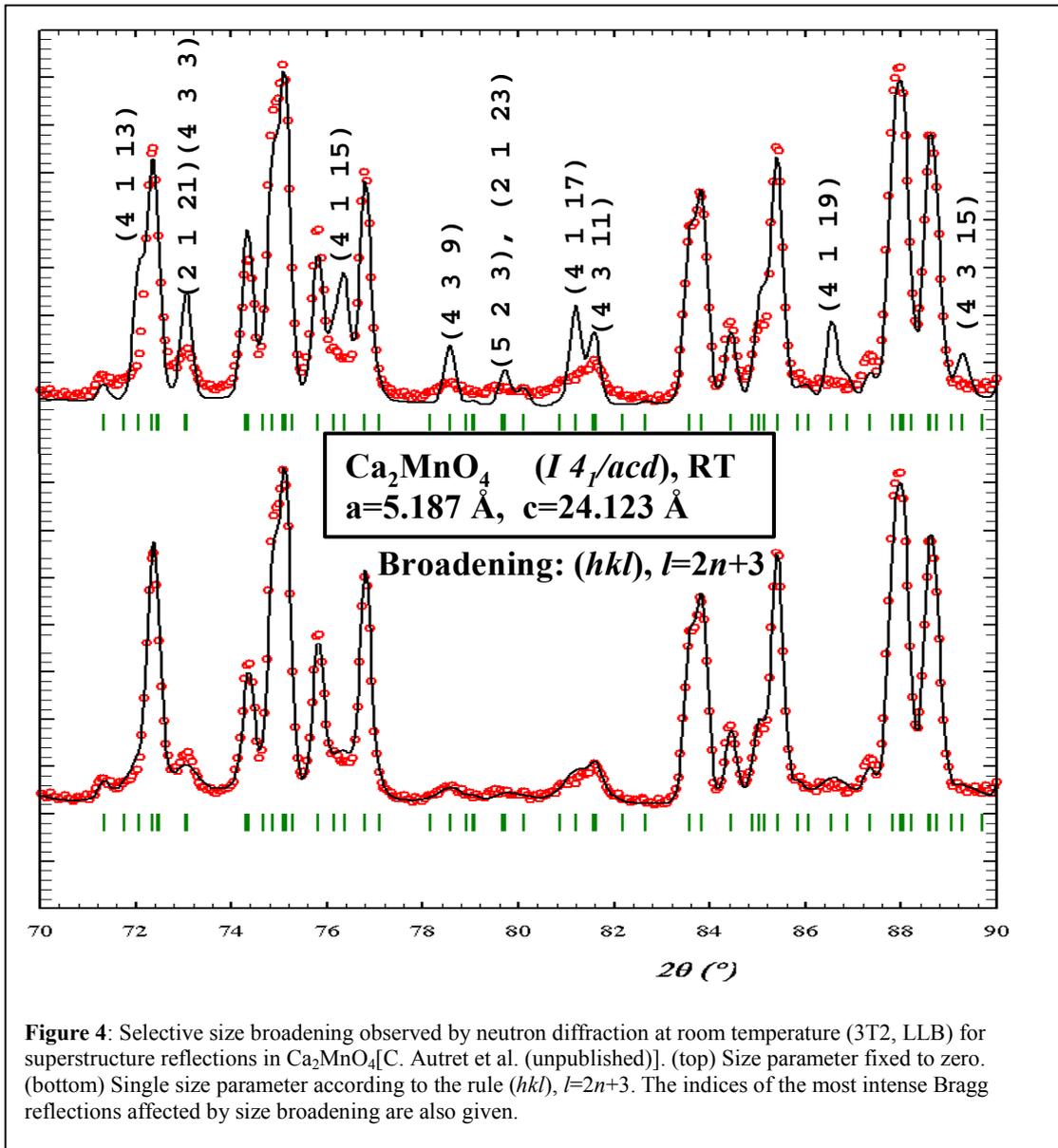


Figure 4: Selective size broadening observed by neutron diffraction at room temperature (3T2, LLB) for superstructure reflections in Ca_2MnO_4 [C. Autret et al. (unpublished)]. (top) Size parameter fixed to zero. (bottom) Single size parameter according to the rule $(hkl), l=2n+3$. The indices of the most intense Bragg reflections affected by size broadening are also given.

of the form: $n_1h + n_2k + n_3l = n_4n + n_5$. Where n is an arbitrary integer and $n_i (i = 1, 2, \dots, 5)$ are integers given by the user. A size parameter is associated to each rule (a maximum of nine rules may be given per phase) that may be refined freely or constrained using the codewords appropriately.

To access this option in *FullProf* the value of **IsizeModel** should be in the interval [-2,-9]. The absolute value of **IsizeModel** corresponds to the number of rules (independent parameters) to be given. If all $n_i=0$ the rule is not used. To give a single rule one must put **IsizeModel** = -2 and put zeros

```

. . . . .
! Selective Size-Broadening:
! hkl cond.      (n1.h + n2.k + n3.l=n n4 +/- n5)   Size-par      Code
  0  0  0          0      0      1      2      3      9.61440    661.000
  0  0  0          0      0      0      0      0      0.00000     0.000

```

Figure 5: Portion of the PCR file for **IsizeModel** = -2 corresponding to the refinement in Figure 4. The first set of zeros below the text 'hkl cond.' is not used at present.

for the last condition. This is needed in order to avoid the confusion with the case of an infinite needle. In Figure 4 we give an example using `IsizeModel = -2` and in Figure 5 the relevant part of the PCR file is written.

Finally, a general formulation for peak shifts, due to defects or to residual stresses, has also been implemented. For `Sol#0`, the lines corresponding to shift parameters are read in the PCR file. Selective shifts can be selected when `Mods < -1`. For this option a set of up to `ABS(Mods) (≤10)` lines can be given. The lines define rules to be satisfied by reflections undergoing shifts with respect to the theoretical Bragg position due to some kind of defects (stacking and twin faults for instance). The rules are similar to those of selective size broadening discussed above. The position of the reflections satisfying the rules are displaced according to the expressions:

$$\begin{aligned} 2\theta_s &= 2\theta_B + 2 \text{SHIFT} d^2 \tan\theta \times 10^{-2} \quad (2\theta \text{ space}) \\ \text{TOF}_s &= \text{TOF}_B - \text{SHIFT} d^3 \text{Dtt1} \times 10^{-2} \quad (\text{T.O.F. space}) \\ E_s &= E_B - \text{SHIFT}/(2d) \text{Dtt1} \times 10^{-2} \quad (\text{Energy space}) \end{aligned}$$

Where the index B stands for the theoretical Bragg position of the non defective material and `SHIFT` is the shift parameter to be refined.

The shift of Bragg reflections may also be due to external stresses or residual stresses. For those cases it is more appropriate to use the following generalized model for shifts. The model is implemented for `Mods = 100+NumLaue` (with `NumLaue` the number of the Laue class according to *FullProf* manual, see appendix), and a set of parameters corresponding up to quartic form in *hkl* can be refined. The position of a reflection is displaced according to the expressions:

$$\begin{aligned} 2\theta_s &= 2\theta_B + 2 S_h d^2 \tan\theta \times 10^{-2} \quad (2\theta \text{ space}) \\ \text{TOF}_s &= \text{TOF}_B - S_h d^3 \text{Dtt1} \times 10^{-2} \quad (\text{T.O.F. space}) \\ E_s &= E_B - S_h/(2d) \text{Dtt1} \times 10^{-2} \quad (\text{Energy space}) \end{aligned}$$

The expression used for calculating the scalar S_h for reflection \mathbf{h} is given by:

$$S_h = \sum_{\{H+K+L=2\}} D2_{HKL} h^H k^K l^L + \sum_{\{H+K+L=4\}} D4_{HKL} h^H k^K l^L$$

The free parameters for this option are the sets $D2_{HKL}$ and $D4_{HKL}$. To refine these parameters the average cell parameters of the non-stressed material should be fixed during the refinement.

FROM HERE TO THE STARTING OF MATHEMATICAL PART THE TEXT NEED TO BE CORRECTED, UPDATED AND COMPLETED

For anisotropic contributions `{DST(STR) and LorSiz}` the actual version of FullProf has the following set of microstructural models:

Size effect: {only Lorentzian component is taken into account for anisotropic broadening of size origin through `LorSiz`}

`Stt=0` and `IsizeModel <> 0` {Integer to select a particular model for `LorSiz` in subroutine SIZE}

`IsizeModel = 1` Platelet coherent domains. `LorSiz` is assumed to be of the form `LorSiz = SZ*cos(phi)`, where SZ is the refined parameter and phi is the acute angle between the scattering vector (h,k,l) and the vector defining the platelet shape of domains `{[Sz1, Sz2, Sz3] in line 36}`

IsizeModel =-1 Needle-like coherent domains. [LorSiz](#) is assumed to be of the form $\text{LorSiz}=\text{SZ}*\sin(\text{phi})$, where SZ is the refined parameter and phi is the acute angle between the scattering vector (h,k,l) and the vector defining the needle shape of domains {[Sz1,Sz2,Sz3] in line 11-9*}

IsizeModel = 2 to 7, in these cases the broadening is considered only for reflections of the form: (00l),(0k0),(h00), (hk0), (h0l) and (0kl). The function is: [LorSiz](#)=SZ.

IsizeModel =8 Only satellites reflections are considered to be broadened. In this case also [LorSiz](#)=SZ.

IsizeModel =9 Reflections (HKL) with H=2n+1 and K=2m+1 are broadened. In this case also [LorSiz](#)=SZ.

IsizeModel =10 Reflections (H0L) with H+L=2n are broadened.
In this case also [LorSiz](#)=SZ.

IsizeModel =11 Reflections (HKL) except (HHL) are broadened.
In this case also [LorSiz](#)=SZ.

IsizeModel =12 Reflections (HKL) with H=2n+1 are broadened.
In this case also [LorSiz](#)=SZ.

IsizeModel =13 Reflections (HKL) with H=2n and K=2m+1 are broadened.
In this case also [LorSiz](#)=SZ.

[Str](#) > 1 : A generalised formulation of the size effect is applied and the size parameters (up to six) are given in line 11-14*.

In that case the [LorSiz](#) function is given by:

$$F(p1\dots p6) = 0.180 \text{ Lambda} / \{ \pi * d(\text{hkl}) * \text{Quad}(p1\dots p6) \}$$

$$\text{Quad} = \text{Sqrt} \{ p1 \text{ h}^2 + p2 \text{ k}^2 + p3 \text{ l}^2 + 2 (p4 \text{ hk} + p5 \text{ hl} + p6 \text{ kl}) \}$$

where the size parameters are p1,p2,...p6.

This formulation assumes that the average diameter <Dhkl> can be expressed as an ellipsoid of refinable parameters p1..p6. Therefore

$$\langle \text{Dhkl} \rangle = d(\text{hkl}) * \text{Quad}$$

Strain effects: {only Gaussian component is taken into account for anisotropic broadening due to strains through DST(STR)}

[Str](#)=0 and IstrainModel<>0 {Integer to select a particular model for DST(STR) in subroutine STRAIN}

$$\text{FWHM}(\text{strain}) = \text{DST} \tan\Theta = c \text{ Sigma}(1/d^2)/(1/d^2) \tan\Theta$$

the constant c is equal to $2 \sqrt{\text{Ln}2} (180/\pi)$. In the program the effective value used is $c*10e-3$, therefore it is necessary to divide the refined strain parameters by 1000. The user can consult the modules FDUM's.FOR for details and reference 17.

IstrainModel = 1 Orthorhombic strain in a tetragonal lattice.
Refined parameter STR1.

IstrainModel = 2 Strain along "a" in an orthorhombic lattice.
Refined parameter STR1.

IstrainModel = 3 Strain along a diagonal of the ab plane in an orthorhombic lattice for which an equivalent monoclinic double cell has been used.
Refined parameter STR1.

IstrainModel = 4 Same as 3 but strains along the two diagonals.
Refined parameters STR1, STR2.

IstrainModel = 5 Strain in an orthorhombic lattice with fluctuations along "a" and "b" and correlation between "a" and "b".
Refined parameters [Str1](#), [Str2](#), [Str3](#).

IstrainModel = 6 Strain in an orthorhombic lattice with fluctuations along "a", "b" and "c" and corr(a,b)=-1, corr(a,c)=1, and corr(b,c)=0.
Refined parameters [Str1](#), [Str2](#), [Str3](#).

IstrainModel = 7 Uniaxial microstrain. DST(STR) is assumed to be of the form DST(STR)=STR1*cos(phi), where STR1 is the refined parameter and phi is the angle between the scattering vector (h,k,l) and the vector defining the axial direction of strains {[[St1](#), [St2](#), [St3](#)] in [line 37](#)}

IstrainModel = 8 General anisotropic strain of hexagonal symmetry.
(See ref. 17)

$$STR1 = Saa = \text{Sigma}(A), STR2 = Scc = \text{Sigma}(C) \quad STR3 = Cac = \text{Correl}(A,C)$$

$$1/d^2 = SQ = A (h^2 + k^2 + hk) + C l^2 = A m^2 + C l^2$$

$$\text{Var}(SQ) = Saa^2 m^2 + Scc^2 l^4 + 2 Saa Scc Cac m l^2$$

$$\text{FWHM}^2 (\text{strain}) = \text{Var}(SQ) (P/SQ)^2$$

$$\{P = 180/\pi * \sqrt{8 \text{Ln}2} \cdot 1.E-03 \tan(\text{theta})\}$$

IstrainModel = 9 Strain adapted for compounds LaNi5-like. Formula by:
P.Thompson et al, J.Less Comm Met 129, 105-114 (1987)

Four parameters: Gaussian FWHM = Sqrt(Sumj[S(j)]) tan(theta)

$$STR1 \rightarrow S(1) = \frac{x \text{ Par}(1)}{(h^2 + k^2 + l^2)}$$

$$STR2 \rightarrow S(2) = \frac{l^4}{(h^2 + k^2 + l^2)} x \text{ Par}(2)$$

$$STR3 \rightarrow S(3) = \frac{(h^2 k^2 + k^2 l^2)}{(h^2 + k^2 + l^2)} x \text{ Par}(3)$$

$$STR4 \rightarrow S(4) = \frac{h^2 k^2}{(h^2 + k^2 + l^2)} x \text{ Par}(4)$$

IstrainModel = 10 General anisotropic strain for orthorhombic symmetry.
(See ref. 17)

6 strain parameters corresponding to :

$$STR1 = Saa = \text{Sigma}(A), \quad STR2 = Sbb = \text{Sigma}(B), \quad STR3 = Scc = \text{Sigma}(C),$$

$$STR4 = Cab = \text{Corr}(A,B), \quad STR5 = Cac = \text{Corr}(A,C), \quad STR6 = Cbc = \text{Corr}(B,C)$$

Where:

$$SQ = A h^2 + B k^2 + C l^2$$

$$\text{Var}(SQ) = Saa^2 h^4 + Sbb^2 k^4 + Scc^2 l^4 +$$

$$2 Saa Sbb Cab h^2 k^2 +$$

$$2 Saa Scc Cac h^2 l^2 +$$

$$2 S_{bb} S_{cc} C_{bc} k^2 l^2$$

$$FWHM^2 = \text{Var}(SQ)(P/SQ)^2$$

$$\{P=180/\pi*\sqrt{8Ln2} 1.E-03 \tan(\theta)\}$$

IstrainModel =11 General anisotropic strain for monoclinic symmetry being the setting $\gamma \gg 0$ (2-fold axis along 001)

(See ref. 17)

{limited to 8 parameter by putting arbitrarily zero
two correlation values (C_{ad}=C_{bd}=0)}

8 strain parameters corresponding to :

STR1=S_{aa}=Sigma(A), STR2=S_{bb}=Sigma(B), STR3=S_{cc}=Sigma(C), STR4=S_{dd}=Sigma(D),
STR5=C_{ab}=Corr(A,B), STR6=C_{ac}=Corr(A,C), STR7=C_{bc}=Corr(B,C), STR8=C_{cd}=Corr(C,D)

Where:

$$SQ = A h^2 + B k^2 + C l^2 + D hk$$

$$\text{Var}(SQ) = \begin{pmatrix} S_{aa}^2 & S_{aa}S_{bb}C_{ab} & S_{aa}S_{cc}C_{ac} & 0 \\ S_{aa}S_{bb}C_{ab} & S_{bb}^2 & S_{bb}S_{cc}C_{bc} & 0 \\ S_{aa}S_{cc}C_{ac} & S_{bb}S_{cc}C_{bc} & S_{cc}^2 & S_{cc}S_{dd}C_{cd} \\ 0 & 0 & S_{cc}S_{dd}C_{cd} & S_{dd}^2 \end{pmatrix} \begin{matrix} (h^2) \\ (k^2) \\ (l^2) \\ (hk) \end{matrix}$$

$$FWHM^2 = \text{Var}(SQ)(P/SQ)^2$$

$$\{P=180/\pi*\sqrt{8Ln2} 1.E-03 \tan(\theta)\}$$

[Str=1,3](#) The generalised formulation of the strain broadening is used and 10 parameter are read in line 11-13*. The formalism is that described in reference 17. The strain parameters correspond to the fluctuations and correlations of direct cell parameters. The lower symmetry is monoclinic with the conventional setting. The hexagonal lattice is treated as a special case.

$$\text{Strain parameters: } P_x = \sigma(x), \quad x,y = a,b,c, \beta$$

$$\text{corr}(x,y) = \sin(P_{xy})$$

The parameter P_{xy} is an angle in degrees. This particular description is for taking into account that $\text{corr}(x,y) = \text{covar}(x,y)/(\sigma(x).\sigma(y))$ and must belong to the interval [-1,1]. The strain parameters are read in the following order:

$$P_a, \quad P_b, \quad P_c, \quad P_{\beta}, \quad P_{ab}$$

$$P_{ac}, \quad P_{a\beta}, \quad P_{bc}, \quad P_{b\beta}, \quad P_{c\beta}$$

Note:

In some cases the use of size-strain options of the program, the computing time is greatly increased. This is related to the fact that the number of reflections contributing to each particular point of the diagram increases because the broadening is quite important. An excessive computing time may indicate a divergence of the refinement that has attributed a too much large FWHM to some reflections.

HKL-dependent shifts.

As a part of microstructural effects, the program can handle some cases of peak-shifts and asymmetry effects. No hkl-dependent asymmetry model is currently available. Two models are available for peak-shifts:

ModS = 1/-1

Uniaxial shifts along a direction [S1,S2,S3]

Shift of Bragg reflections of the form:

Shift(hkl) = Shift1 * cos(Phi(hkl)) (ModS= 1)

or Shift(hkl) = Shift1 * sin(Phi(hkl)) (ModS=-1)

Where Phi is the angle: between [hkl] and [S1,S2,S3]

ModS = 2

Shift(hkl) = Shift1 * Coeff(hkl) * tan(Theta(hkl))

The parameter Coeff(hkl) must be given by user in the file from which Bragg reflection indices and multiplicities are read. This option cannot be used simultaneously with the output of a Fourier file, so Fou is set to zero by the program.

Quantitative phase analysis

For quantitative analysis it is essential that two conditions be fulfilled:

- sample must be carefully prepared to comply to the definition of a powder: homogeneity, sufficient number of particles with random orientation
- structure factors must be correctly calculated

According to Brindley, it is convenient to classify mixed powders according to the value of the product mD where m is the linear absorption coefficient and D a measure of the linear size of a particle. Four cases must be considered:

- fine powders: mD < 0.01

The individual particles of the powder have negligible absorption and no correction has to be applied to the data

- medium powders: 0.01 < mD < 0.1
- coarse powders: 0.1 < mD < 1
- very coarse powders: mD > 1

In a mixture of N crystalline phases the weight fraction W_j of phase j is given by:

$$W_j = \{ S_j Z_j M_j V_j / t_j \} / \text{Sum}(i)[S_i Z_i M_i V_i / t_i]$$

where S_j is the scale factor of phase j

Z_j is the number of formula units per unit cell for phase j, M_j is the mass of the formula unit, V_j is the unit cell volume t_j is the Brindley particle absorption contrast factor for phase j defined as:

$$t_j = (1/V_j) \text{Integ}[\exp\{-(m_j - \mu)x\} dV_j]$$

where V_j is the volume of a particle of phase j m_j is the particle linear absorption coefficient μ is the mean linear absorption coefficient of the solid material of the powder x is the path of the radiation in the particle of phase j when reflected by the volume element dV_j

The latter parameter accounts for microabsorption effects that become important when the compounds of the powder have rather different linear absorption coefficients. Its calculation requires only the knowledge of the particle radius R and linear absorption coefficient m. Values of t as a function of the product (m_j-μ)R have been tabulated by Brindley and are reproduced below:

(mj-mu)R t(2Theta=180)	t(2Theta=0)	t(2Theta=90)	t(2Theta=180)	(mj-mu)R t(2Theta=180)	t(2Theta=0)	t(2Theta=90)	t(2Theta=180)
-0.50	2.068	2.036	2.029	-0.40	1.813	1.807	1.827
-0.30	1.508	1.573	1.585	-0.20	1.352	1.353	1.362
-0.10	1.159	1.162	1.163	-0.09	1.142	1.143	1.144
-0.08	1.124	1.125	1.125	-0.07	1.107	1.108	1.108
-0.06	1.090	1.091	1.091	-0.05	1.074	1.073	1.074
-0.04	1.059	1.058	1.059	-0.03	1.043	1.042	1.042
-0.02	1.028	1.027	1.027	-0.01	1.014	1.014	1.014
0.00	1.000	1.000	1.000	0.01	0.986	0.986	0.986
0.02	0.972	0.973	0.973	0.03	0.959	0.960	0.960
0.04	0.945	0.946	0.947	0.05	0.932	0.933	0.934
0.06	0.918	0.919	0.921	0.07	0.905	0.906	0.908
0.08	0.892	0.893	0.895	0.09	0.878	0.879	0.882
0.10	0.865	0.866	0.870	0.20	0.742	0.753	0.760
0.30	0.640	0.653	0.671	0.40	0.545	0.569	0.587
0.50	0.468	0.496	0.529				

R.J. Hill & C.J. Howard, J. Appl. Cryst. 20, 467-476 (1987)
G.W. Brindley, Phil. Mag. 36, 347-369 (1945)

User-supplied parameters & subroutines for structure-factor calculation

Some powder diffraction applications need user-supplied parameters and subroutines calculating the structure factors. Examples of this kind of applications are the following:

- Form factor determination.
- Rigid body-generalised co-ordinates refinements.
- TLS refinements.
- Description of incommensurate phases in real space (not Fourier components) by some simple parameters.
- Anharmonic displacement parameters.

To do that, the user can prepare his(her) own subroutines and link the object FDUM1,FDUM2,FDUM3,... with the rest of the modules and the FullProf library.

For crystal structures the subroutine to calculate the structure factor squared for a reflection (hkl) must be called STRMOD. For a magnetic phase the name is MAGMOD. Examples of these are contained in the public source code FDUM's.FOR.

The user can also modify or introduce new models for strains and size effects in the subroutines SIZE and STRAIN which are also included in FDUM's.FOR

Useful information for users concerning the internal parameters:

The parameters (refinable variables) used in FullProf are divided in three categories: Global parameters, Atom parameters and Phase parameters. For each category there is an array to store the values: GLB, XL and PAR.

The global parameters are the following:

GLB(J) : J=1,2,...40

J=1 : T0 Zero

J=2,...7: b1,b2,...b6 Background parameters

J=8,9 : SYCOS,SY SIN Systematic 2Theta-shifts/ DTT1,DTT2 for TOF

J=10..15: Bc1,...Bc6 Background coefficients for Debye-like function: DBback

J=16..21: d1,.... d6 distances in Angstroms in DBback
 J=22,23,24,25: a,b,c,d Background transforming coefficients
 J=26 : Flambda Wavelength (for CW neutron)/ 2SinTheta for TOF
 J=27,28,29: P0,Cp,Tau Microabsorption coefficients
 J=30..35: Additional background parameters
 J=36..40: Not used at present

XL(I,J): Atom parameters, the index I runs over atoms

I=1,2,..N1,N1+1,N1+2,....N1+N2,N1+N2+1,....N1+N2+N3....N

N1: number of atoms of phase 1

N2: number of atoms of phase 2

.....
 N : total number of atoms : N1+N2+N3+...+N([Nph](#))

The index J represents different atom parameters

J=1,2,3 : (x,y,z) fractional co-ordinates
 J=4 : B isotropic temperature factor
 J=5 : Occ. Occupation factor.
 J=6...8: (Mx,My,Mz) Magnetic moment components
 9...11: (Mxi,Myi,Mzi) Imaginary Magnetic moment components
 J=12 : [MagPh](#) Magnetic phase of the atom
 J=13..18: (betas11,22..) Anisotropic thermal factors
 J=19..32: Coefficients of generalised form-factors

These atomic parameters can be used by the subroutines STRMOD and MAGMOD for other purposes.

Moreover, there are other non-atomic parameters which can be used also for defining generalised co-ordinates. They are internally stored in the array PAR(K,J) where the index K run over the phases and J represents different parameters:

PAR(K,J):K=1,2,....[Nph](#)

J=1 : S Scale factor
 J=2 : Bov Overall temperature factor
 J=3,4,5 : U,V,W Halfwidth parameters
 J=6,..11 : A,B,C,D,E,F Cell constants
 J=12,13 : G1,G2 Preferred orientation parameters
 J=14 : As1 First Asymmetry parameter
 J=15,16 : [X,Y](#) Shape parameters (depend on [Npr](#))
 J=17 : eta(m) Shape parameter of p-Voigt(Pearson)
 J=18..27 : Str1...Str10 Strain parameters
 J=28 : [GausSiz](#) Isotropic Gaussian size parameter
 J=29..34 : Siz1...Siz6 Anisotropic size parameters
 J=35,36,37 : As2,As3,As4 Additional asymmetry parameters
 J=38 : Asymhkl HKL-dependent asymmetry parameter
 J=39,40 : [Shf1,Shf2](#) HKL-dependent shift parameters
 J=41,42 : Shap1,Shap2 Additional shape parameters
 J=43,44,45 : [U2,V2,W2](#) U,V,W parameters for lambda(2)
 J=46,47,48 : Ud,Vd,Wd U,V,W for right part of SpV(x)
 J=49,50 : Eta0d, Xd Eta and X for right part of SpV(x)
 J=51,..59 : CYi Coefficients of Spherical Harmonics (size broad.)
 J=60..MPAR : P1,P2,.... User-supplied parameters
 LASTMPAR=59 Number of the Last parameter used internally. To be used as offset for user-supplied subroutines.

Examples of user-supplied subroutines for structure-factor calculation

CRYSTAL STRUCTURE REFINEMENTS

In FullProf the default `Jbt=4` option is an updated subroutine written by V.Rodriguez which handles rigid body objects.

Guide for the General Rigid-Body-Constraints/TLS Subroutine for Rietveld Refinements

V1.4 (January 1997)

Dr V.Rodriguez

Laboratoire de Spectroscopie Moleculaire et cristalline *
(U.A.124 CNRS)

Universite de Bordeaux I, 351 cours de la Liberation
F-33405 TALENCE Cedex

Tel : (33) 56.84.63.61

Fax : (33) 56.84.84.02

e-mail: vincent@loriot.lsmc.u-bordeaux.fr

<http://loriot.lsmc.u-bordeaux.fr>

* UMR Physico-Chimie Moleculaire since January 1st 1997

For further details see

"A routine for crystal-structure refinements based on rigid-body model with constrained generalised co-ordinates and mean thermal displacements"

Rodriguez V. and Rodriguez-Carvajal J.,
J. Applied Cryst., (to be published)

In the present version of Fullprof the default `Jbt=4` option refers to a subroutine which handles rigid body objects. Here we give a short guide in order to facilitate the use of this subroutine. This routine should be used with caution and the user should be familiar enough with "conventional" Rietveld refinements. The subroutine is still under improvement. At this time, all the options have been well tested except option Nr 4 (see below). I acknowledge all suggestions and notifications of possible bugs found in the program.

It should be pointed out that option `Jbt=4` is not restricted to "perfect" rigid-body-groups. In the standard definition of Rietveld atomic positions, reduced co-ordinates are used. Within this subroutine, molecules or atomic groups are defined by spherical internal co-ordinates and six additional parameters, which define the groups' position and orientation in the crystal.

In addition, this extended version of Rietveld refinement allows distorting groups in a final refinement step if required.

Note: Each parameter referring to the orthonormal crystal system is recognised by the appended "c". Each parameter referring to the orthonormal molecular system is recognised by the appended "m".

Description of the parameters

- Any kind of rigid body groups built from atoms can be generated.
- All parameters described below are refinable except optional parameters P6 (always true) and P16 (specific to "satellite" rigid body groups).
- Each group of atoms (rigid body group, RBG) is identified by 1 or 2 letters and a number from 1 to 99 (both compatible with the standard DBWS format and the free format of Fullprof). The number indicates that a RBG is defined. The Atom(two letters max.) and the number of the atom must not be separated.

- Representation of RBG is performed as follow :

1) Definition of the isolated group in an orthonormal molecular system. Each atomic position (xm,ym,zm) is defined through spherical co-ordinates (dm=distance, thetam, phim). The centre of this molecular system may or may not coincide with an atom.

2) Definition of the absolute position (Xc,Yc,Zc) and orientation (angles THETAc, PHIC and CHIC) of the molecular orthonormal system in the crystallographic orthonormal system. These are standard EULER angles definitions. In this way, each group is entirely defined within a crystal.

- Anti-clockwise rotations are always applied to spherical angles. These standard spherical angles are defined as follow :

THETAc / thetam : inclination with respect to the Zc/zm axis of the corresponding orthonormal system [-Pi, Pi].

PHIC / CHIC and phim: rotation around the Zc/zm axis of the corresponding orthonormal system [0, 2*Pi].

- The orthonormal crystallographic system, with respect to which the spherical angles THETAc, PHIC and CHIC of any RBG are defined, fulfils the following conditions:

Xc axis coincides with the crystallographic direction a

Yc axis belongs to the plane (a, b)

Zc axis is perpendicular to the plane (a, b) (parallel to c*)

One can imagine how to place a rigid molecule in the correct position in the unit cell by making the following operations:

- The RBG has been completely defined by the spherical co-ordinates (dm,thetam,phim) of each atom in the internal orthogonal system, that coincides at the beginning with the orthogonal crystallographic system.

- Perform a rotation CHIC of the whole RBG around the zm axis of matrix:

$$\begin{pmatrix} \text{COS}(\text{CHIC}) & -\text{SIN}(\text{CHIC}) & 0 \\ \text{SIN}(\text{CHIC}) & \text{COS}(\text{CHIC}) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$M1 = \begin{pmatrix} \text{COS}(\text{CHIC}) & -\text{SIN}(\text{CHIC}) & 0 \\ \text{SIN}(\text{CHIC}) & \text{COS}(\text{CHIC}) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$$

- Perform a rotation of the whole RBG inclining their z-axis to the angles THETAc and PHIC of matrix:

$$\begin{pmatrix} \text{COS}(\text{THETAc}) * \text{COS}(\text{PHIC}) & -\text{SIN}(\text{PHIC}) & \text{SIN}(\text{THETAc}) * \text{COS}(\text{PHIC}) \\ \text{SIN}(\text{THETAc}) * \text{COS}(\text{PHIC}) & \text{COS}(\text{PHIC}) & \text{COS}(\text{THETAc}) \\ -\text{SIN}(\text{THETAc}) & 0 & \text{COS}(\text{THETAc}) \end{pmatrix}$$

$$M2 = \begin{pmatrix} \text{COS}(\text{THETAc}) * \text{SIN}(\text{PHIC}) & \text{COS}(\text{PHIC}) & \text{SIN}(\text{THETAc}) * \text{SIN}(\text{PHIC}) \\ \text{SIN}(\text{THETAc}) * \text{SIN}(\text{PHIC}) & \text{SIN}(\text{PHIC}) & \text{SIN}(\text{THETAc}) * \text{COS}(\text{PHIC}) \\ -\text{SIN}(\text{THETAc}) & 0 & \text{COS}(\text{THETAc}) \end{pmatrix}$$

$$\begin{pmatrix} -\text{SIN}(\text{THETAc}) & 0 & \text{COS}(\text{THETAc}) \end{pmatrix}$$

- Translate the origin of the RBG to the position (Xc,Yc,Zc) which are transformed in reduced co-ordinates (xo, yo, zo) with xo=Xc/a, yo=Yc/b, zo=Zc/c

- "Free atoms" (unconstrained or isolated) can be added with number 0 or without number. Each isolated atom has the same definition of parameters as described in the Fullprof manual (see lines 11.4.1 and 11.4.3 of the Fullprof guide). Current information about "free atoms" is printed to the screen when parameter P6 (see below) is set to any negative value. Note that the following parameters: P6, P7, P8, P15 and P16, are obsolete in this case.

- Rigid body satellite groups (RBSG) can be also included in this version, for example a methyl group within a rigid group such as [N(CH3)4]+(tetramethyl ammonium). The definition and the structure of the parameters are almost the same as those for a main RBG. The co-ordinates of the centre of the satellite group should not be specified since they are specified through the knowledge of its absolute position in the input file "x.pcr". The orientation THETA_c,PHI_c of satellites groups is also defined following the value of parameter P16 of the 1st satellite atom (see below, option abs(P6)=2).

One easy external degree of freedom of the RBSG is the rotation around the z-molecular axis i.e. the N-C bond. This degree of freedom is accessible through the EULER angle PHI_c that can be refined.

- All output is identical for isolated and/or constrained atoms.

- Every atom has 16 items stored in (X(Iphase_j),j=1,16) and each parameter is denoted here as P1, P2, ...P16. The reading of these parameters is described in lines 11-4-1 and 11-4-3 of the Fullprof guide.

Example:

```

          P1  P2  P3  P4  P5  P6  P7-THETAc P8-PHIc
TD1 N      .38900 .54090 .28796 4.00000 0.16667 1.000 -.290 .142
          .00 .00 .00 .00 .00 .00 71.00 61.00
1.49300 1.57080 -.95532 .33333 .66667 .25000 1.04100 0
          .00 .00 .00 .00 .00 .00 51.00

```

```

P9-dm P10-thetam P11-phim P12-xo P13-yo P14-zo P15-CHIc P16
          x   y   z
TD2 C      .47744 .84054 .28796 4.00000 0.16667 .000 .000 .000
          .00 .00 .00 .00 .00 .00 .00 .00
.75000 1.57080 1.57080 .00000 .00000 .00000 .00000
          .00 .00 .00 .00 .00 .00 .00

```

P9-dm P10-thetam P11-phim

```

          x   y   z
O  O      .28146 -.04125 .41456 .00000 1.00000 -1.000 .000 .000
          .00 .00 .00 .00 .00 .00 .00 .00
.03957 .07078 .09069 .00314 .01571 -.01501 0
          .00 .00 .00 .00 .00 .00
beta11 beta22 beta33 beta12 beta13 beta23

```

-> In a RBG (here, the generic name is TD) each atom (TD1 and TD2) has its internal co-ordinates stored in the following items:

```

distance to the centre dm : P9
Spherical angle thetam   : P10
Spherical angle phim    : P11

```

The atomic reduced co-ordinates are only for information with option 1 (RBG), see below for details.

```

Reduced co-ordinate x : P1
Reduced co-ordinate y : P2
Reduced co-ordinate z : P3

```

The parameters P4 and P5 have their usual meaning for each atom:

```

Isotropic temperature factor: P4
Occupation number           : P5

```

-> The first atom of each RBG (atom No 1, here TD1) contains the fractional co-ordinates (xo, yo, zo) of the centre of the RBG (P12, P13, P14) and the three EULER orientation angles THETAc, PHIc and CHIc (in radians, respectively P7=-.290, P8=.142 and P15=1.041) of the whole group. Here, the centre of the RBG does not coincide with an atom (N or C in this example) since the first atom has a non-zero value dm=P9=1.493. Of course, it is possible to build a RBG with an atom coinciding with the centre of the RBG.

-> The third atom (oxygen, Number 0 (or none which is equivalent to 0)) is unconstrained ("free atom") and its items are the standard ones : here, reduced co-ordinates (x, y, z) + atomic temperature parameters (betaij).

-> No terminal output is required for the whole RBG TD (P6>0). Terminal output at each cycle is required for "free atom" oxygen (P6<0).

Practical details

1- The item abs(P6) of the first atom indicates the option selected. If P6<0, the current information is printed on the screen and in the file CODFIL.OUT. (file x.out). If the value of P6 is 0 for a RBG, the parameter defaults to "1", corresponding to the standard RBG option.

2- The distance are expressed in the same unit as wavelength and cell parameters (usually in angstroms) and the angles are always expressed in radians:

3- The listing of the different options is a function of the main optional parameter P6. In the following, the temperature parameters are exclusively Boveral or Biso (isotropic Debye-Waller factors) except for TLS option with abs(P6)=5.

abs(P6)= 1.0x : Normal rigid body option. If x=1 the fractional co-ordinates of the center of mass is output. That supposes that every atom of the Molecule has been given explicitly in the asymmetric unit.

2.xx : Satellite group (RBSG) option (Int(abs(P6))=2). The integer value xx=100*(int(abs(P6))-2) gives the absolute number of the reference group (as they appear following the writing order, whatever the number of phase).

--> The parameter P15 is assigned to the rotation CHIc of RBSG as for RBG.

--> The parameter P16 of the first atom of the RBSG is defined as follows: P16 = N1.N2 with N1=Int(P16) : Nr of the first reference atom of the reference RBG.

N2=100*(P16-Int(P16)): Nr of the second reference atom of the reference RBG.

N1: Centre of the RBSG

N2: Optional (if N2 is given then N1-N2 defines the internal z-axis)

Example: 3.02 1st reference atom = Nr 3 2nd reference atom = Nr 2 Atom Nr 3 is the centre of the RBSG and the z-axis of the RBSG is oriented in the direction: atom Nr 2-> atom Nr 3. Of course, the (xo, yo, zo) fractional co-ordinates of the RBSG are not needed. The program calculates automatically the corresponding values.

a) If the second atom is defined (N2#0), the spherical angles of the RBSG are calculated from these two reference atoms.

b) If the second atom is not defined (N2=0), the centre of the main RBG is taken as the second reference atom of the RBSG.

c) If N2=N1, the spherical orientation angles of the RBSG are those of the main reference RBG.

Apart from these constraints, a RBSG is treated as a normal RBG with as many atoms as desired. The refinement of the centre of the RBSG as well as the orientation angles THETAc and PHIc are performed in the main RBG.

3.0x : The spherical co-ordinates (parameters P9, P10 and P11) are generated from the reduced co-ordinates (P1, P2 and P3) at the first cycle. Then the parameter P6 is automatically set to 1 and the sign of the option is kept i.e. the next option is RBG. For selecting the internal orthogonal system the user has to give in the parameters (P12,P13,P14) the origin of the internal orthogonal frame and in (P9,P10,P11) the co-ordinates of an atom (that may be fictitious) for defining the plane "xz". The (P1,P2,P3) co-ordinates of the first atom defines the z-axis of the internal frame, which is in the direction V3=(P1-P12,P2-P13,P3-

P14) in the conventional crystal frame (fractional components). the "y" axis is perpendicular to the plane "xz" defined by the vectors:

$$V3=(P1,P2,P3)-(P12,P13,P14) \rightarrow z\text{-axis}$$

$$V1=(P9,P10,P11)-(P12,P13,P14) \rightarrow \text{within the xz plane y-axis is in the direction } V3 \times V1$$

This option is very useful as it facilitates the use of standard input file with [Jbt#4](#), the conversion of published structures into spherical internal co-ordinate systems, etc... If x=1 the fractional co-ordinates of the center of mass is output as in option 1.

4.xx : Option to generate an idealised molecule like aliphatic chains, planar or helical molecules..., where xx is the number of atomic planes along the RBG Zc axis which is the reference axis. The spherical co-ordinates are calculated from three parameters P9,P10 and P11. These parameters are lost after processing. They are defined as:

P9 - distance of one atom to the center of the axis

P10 - angle between two atoms lying in consecutive planes and the corresponding mid-point lying along the Zc-axis.

P11 - order of the generating axis (integer)

Examples: 1-For an aliphatic chain Cn along Zc in the (Xc, Zc) plane (conformation all trans).

distance = 1.54/2. (angstroms)

angle = 114.*pi/180. (radians)

axis order = 2

2-For benzene lying in the plane (Xc, Yc)

distance # 1.40*2. (angstroms)

angle = 0.*pi/180. (radians)

axis order = 6

The parameter P15 is assigned to CHlz as in the RBG option and the parameter abs(P6) is set to 1 in the 1st cycle, the sign of the option is kept. Therefore, the generated group is treated as a normal RBG in the next cycles.

Remark : Further improvements are scheduled for this option that is not very reliable at the present time.

5.0x : TLS option for RBG including satellite groups, if any. This TLS version is based on the formalism of V.Schomaker and K.N.Trueblood, Acta Cryst.(1968), B24, p-63. In this case, the refinement is performed with the so-called one step process i.e. atomic positions and temperature factors TLS are refined together.

The origin of the main RBG *must be* the centre of mass of the entire group concerned !!! The subroutine calculates the centre of mass of the molecule if x=1. This option **MUST** be set when the centre of the group does not coincide with the centre of mass. In this case, make sure that all atoms constituting the RBG are defined in the input file since the RBG option does not generate atomic positions. Such a situation can occur in non-centrosymmetric space group, as in urea SPG: P -4 21 M ,since the molecule is located on a C2v site (mm.). Here, one must enter the entire molecule in the asymmetric unit.

The elements of the T(6), L(6) and S(9) matrices are read in "further parameters [Furth](#)" (see Fullprof guide, line 11.12) following the usual order i.e. 11 22 33 12 13 and 23 for T and L. As the S matrix is not symmetric, 9 elements can be required in the general case. The six first elements are as defined previously, and the three last ones are respectively S21, S31 and S32.

Similarly to the atomic temperature parameters betaij, the components of the TLS matrices have symmetry constraints, which are imposed by the symmetry of the crystallographic site (not the molecular symmetry !). These symmetry relations can be found in the paper by Schomaker and Trueblood (1968).

The T elements are expressed in angstroms², the Lcomponents in radians² and the S ones in angstroms*radians. For convenience, the output to the screen and to unit 7 (x.out) of the TLS components are expressed in the following units: angstroms² for T, in degrees² for L and degrees*angstroms for S.

Remark : When using this option, one refines the observed structure factors by assuming that the atomic temperature factors are constrained *ab initio* to satisfy the rigid-body hypothesis. It is well known that the RBG/TLS can greatly reduce the number of atomic and thermal parameters, especially when the molecule is located at a site of high symmetry, but the user should be familiar enough with the TLS hypothesis not to perform inconsistent refinements.

4- It is advised to use specific rigid body refinement codes as damping parameters, especially for the rotational parameter which need to be currently three time higher than the other ones. It is highly recommended to use the options with screen output (P6 with a negative value) to control each RBG refinement steps since good constrained refinements are not so easy to perform.

5- Concerning the refinements: when a main RBG has satellites groups, the derivatives of the general RBG parameters, i.e. the 3 orientation angles THETAc, PH1c and CH1c and the co-ordinates (xo, yo, zo) of the centre of the group, contain the contributions of the satellites groups. In contrast, the derivatives of the internal spherical parameters do not account for these contributions.

6- When this subroutine is used, three additional files of fixed names are created at the end: x#.m, x#.bs and x#.ortep (where # stands for the number of the phase), containing atomic parameters in a format suitable to be used with well-known software packages (Molview, Balls & Sticks and ORTEP). Note that files for Molview are directly executable. The output for the two other software are incomplete in the sense that they contain either Cartesian co-ordinates (Balls & Sticks) or reduced co-ordinates + atomic temperature parameters (ORTEP) of atoms located in the asymmetric unit.

MAGNETIC STRUCTURE REFINEMENTS

In the present version of FullProf the default [Jbt=5](#) option corresponds to the calculation of the magnetic structure factor for a conical magnetic structure. The magnetic intensities are calculated following standard formula as given, for instance, in the paper by J.M.Hasting & L.M.Corliss published in PhysRev 126(2),556 (1962). No symmetry operation can be introduced: all the magnetic atoms within a primitive unit cell must be given (constraints have to be introduced through the codes of the parameters). The subroutine MAGMOD, in this case, does not take into account symmetry for calculations (see the modules FDUM's.FOR for details). However [Isy](#) must be set to 1, the value of the four parameters [Nsym](#), [Cen](#), [Laue](#), [MagMat](#) should be: 0 1 1 0.

The atom parameters correspond to the following variables:

- > P1,P2,P3,P4 and P5 correspond to x,y,z,B,occ of the atom
- > P6 is the magnetic moment of the atom (in Bohr magnetons)
- > P7 is the half-angle cone of the atom (degrees)
- > P8 is the magnetic phase of the atom (in fractions of 2 pi)
- > P9 of the first atom correspond to Phi (degrees)
- > P10 of the first atom correspond to Theta (degrees)
- (Phi,Theta) are the spherical angles(degrees) of the cone axis

The orthonormal system with respect to which are defined the spherical angles verifies:

- X axis coincides with the crystallographic A
- Y axis belongs to the plane A,B
- Z axis is perpendicular to the plane A,B

The particular implementation of spherical components in magnetic structure refinements is that Z axis must coincide with C. That works in all crystallographic systems except for triclinic. The monoclinic setting must be changed to -> 1 1 2/m

It is recommended to generate the reflections using P-1 as Space group symbol and use for conical structures the value [Irf=0](#) (Generates satellites+ fundamentals). For a pure helix [Irf=-1](#) (only satellites are generated). The number of propagations vectors must be set to -1 (to generate + and - satellites). For calculating the magnetic moments in different cells the following formula should be used:

The magnetic intensity is given by the following formula:

$$Fm^{**2} = (p \cdot \sin w)^{**2} \cdot \text{Sum}(j) \{ m(j) f(H,j) \cos(bj) \exp(2\pi i \cdot H \cdot rj) \}$$
$$Fm^{**2} = p^{**2} \cdot (1 + \cos w)^{**2} / 4 \cdot \text{Sum}(j) \{ m(j) f(H+k,j) \sin(bj) \exp(2\pi i \cdot H \cdot rj - f_j) \}$$

The correspondence with the parameters P_1, \dots, P_{10} is the following:

$$\begin{aligned} r_j &= (P_1, P_2, P_3)_j \\ f(H^+ - k, j) &= P_5_j \cdot \text{FormFactor}(H^+ - k, j) \exp(-P_4_j (\sin \theta / \lambda)^2) \\ m(j) &= P_6_j \\ b_j &= P_7_j \\ \tilde{f}_j &= P_8_j \end{aligned}$$

The unitary vector defining the axis of the cone is given by:

$$n = (\cos P_9 \sin P_{10}, \sin P_9 \sin P_{10}, \cos P_{10})$$

$\cos w = n \cdot Q / \text{mod}(Q)$, where Q is the scattering vector

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 input control file **CODFIL.pcr** described in the appendix.

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 $\{T_i, y_i\}_{i=1, \dots, n}$. In the case of data that have been manipulated or normalised in some way the three arrays $\{T_i, y_i, \sigma_i\}_{i=1, \dots, n}$, where σ_i is the standard deviation of the profile intensity y_i , are needed in order to properly weight the residuals in the least squares procedure. 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:

$$y_{c,i} = \sum_{\phi} S_{\phi} \sum_{\mathbf{h}} I_{\phi, \mathbf{h}} \Omega(T_i - T_{\phi, \mathbf{h}}) + b_i \quad (3.1)$$

The vector \mathbf{h} (= \mathbf{H} , or = $\mathbf{H} + \mathbf{k}$) labels the Bragg reflections, the subscript ϕ 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* $I_{\phi, \mathbf{h}}$. This includes the usual meaning of a phase and also the case of the magnetic contribution to scattering (treated usually as a different *phase*) coming from a single crystallographic phase in the sample. The general expression of the integrated intensity is:

$$I_{\phi, \mathbf{h}} = \{L A P C F^2\}_{\phi, \mathbf{h}} \quad (3.2)$$

For simplicity we will drop the ϕ -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_{ϕ} is the scale factor of the phase ϕ
- $L_{\mathbf{h}}$ contains the Lorentz, polarisation and multiplicity factors
- $F_{\mathbf{h}}$ is the structure factor¹
- $A_{\mathbf{h}}$ is the absorption correction
- $P_{\mathbf{h}}$ is the preferred orientation function
- Ω is the reflection profile function that models both instrumental and sample effects

¹ The ratio of the intensities for the two wavelengths handled by the program is absorbed in the calculation of $F_{\mathbf{h}}^2$, so that only a single scale factor is required. For magnetic scattering $F_{\mathbf{h}}^2$ is the square of the magnetic interaction vector (see below).

- C_h includes special corrections (non linearity, efficiencies, special absorption corrections, extinction, etc)
- b_i 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 $\{y_{c,i}(\mathbf{\alpha})\}_{i=1,\dots,n}$ against the parameter vector $\mathbf{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_p)$. The function minimised in the Rietveld Method is:

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

with $w_i = \frac{1}{\sigma_i^2}$, being σ_i^2 the variance of the "observation" y_i . In more complex cases the user may consider several diffraction patterns, or some chemical constraints. For those cases the general expression of the function to be minimised is:

$$\chi_T^2 = \sum_{p=1}^N \omega_p \chi_p^2 + \chi_G^2 = \sum_{p=1}^N \omega_p \left[\sum_{i=1}^n w_i \{y_i - y_{c,i}(\mathbf{\alpha})\}^2 \right]_P + c \sum_{j=1}^m \frac{1}{\sigma_{gj}^2} \{g_j - g_{c,j}(\mathbf{\alpha})\}^2 \quad (3.4)$$

Where χ_p^2 and χ_G^2 are the chi-square of the pattern P and the chi-square of soft constraints. The weight factors ω_p are provided by the user and are internally normalised in order to get $\sum_{p=1}^N \omega_p = 1$, for the N patterns. The quantity g_j is the prescribed value of a constraint (distance, angle, valence, magnetic moment, etc) with standard deviation σ_{gj} . The smaller the value of σ_{gj} the higher is the strength of the constraint. The calculated value of the constraint $g_{c,j}(\mathbf{\alpha})$ is performed as a function of a subset of components of the vector parameter $\mathbf{\alpha}$. The normalisation constant c is taken as the current value of the global reduced chi-square for all the diffraction patterns. For simplicity we shall consider the expression (3.3) to explain some standard points concerning the least squares optimisation. If the optimum set of free parameters is $\mathbf{\alpha}_{opt}$, the necessary condition for a minimum of (3.3) is that the gradient of χ^2 should be

$$\text{zero: } \left(\frac{\partial \chi^2}{\partial \mathbf{\alpha}} \right)_{\mathbf{\alpha}=\mathbf{\alpha}_{opt}} = 0$$

A Taylor expansion of $y_{ic}(\mathbf{\alpha})$ around an initial guest $\mathbf{\alpha}_0$ allows the application of an iterative process. The shifts to be applied to the parameters at each cycle for improving χ^2 are obtained by solving a linear system of equations (normal equations)

$$\mathbf{A} \delta_{\mathbf{\alpha}_0} = \mathbf{b} \quad (3.5)$$

where the components of the $p \times p$ matrix \mathbf{A} and vector \mathbf{b} in the Gauss-Newton algorithm, used within **FullProf**, are given by the expressions:

$$\begin{aligned} A_{kl} &= \sum_i w_i \frac{\partial y_{c,i}(\mathbf{\alpha}_0)}{\partial \alpha_k} \frac{\partial y_{c,i}(\mathbf{\alpha}_0)}{\partial \alpha_l} \\ b_k &= \sum_i w_i (y_i - y_{c,i}) \frac{\partial y_{c,i}(\mathbf{\alpha}_0)}{\partial \alpha_k} \end{aligned} \quad (3.6)$$

The shifts of the parameters δ_{α_0} obtained by solving the normal equations are added to the starting parameters giving rise to a new set $\alpha_1 = \alpha_0 + \delta_{\alpha_0}$. The new parameters are considered as the starting ones in the next cycle and the process is repeated until a convergence criterion is satisfied. The shifts applied to the current parameters may be pre-multiplied by a user defined factor that depend on each individual parameter (through the codeword) and a relaxation factor depending on a whole class of parameters (see appendix). The standard deviations of the adjusted parameters are calculated by the expression:

$$\sigma(\alpha_k) = |a_k| \sqrt{(\mathbf{A}^{-1})_{kk} \chi_v^2} \quad (3.7)$$

Where the reduced chi-square is defined as:

$$\chi_v^2 = \frac{\chi^2}{n - p} \quad (3.8)$$

The quantity a_k is the coefficient of the codeword for the parameter α_k . The χ_v^2 quantity used in the above formula is always calculated for the points in the pattern having Bragg contributions, thus σ_i could be greater than the corresponding value calculated with other programs. In **FullProf** the quantity χ_v^2 is also calculated for all points considered in the refinement, so the user can easily re-calculate the alternative value of the standard deviation.

In normal least square refinements the weighting scheme is based in fixed variances of the profile intensities. This is appropriate when the statistics is good enough to be considered as Gaussian. For low counting statistics a maximum likelihood refinement is more appropriate. This is equivalent to calculate the variance according to the model instead of using experimental fixed variances that are very bad estimated when low counting rate is dominant. Maximum likelihood can be used by setting the variable to 1. In such a case the weights of the observations are calculated at each cycle as: $w_i = 1/\sigma_{calc,i}^2$ supposing a Poissonian distribution and correcting for eventual normalisation factors used in the input data file to estimate the experimental variances.

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 ($n = \text{NPTS-NEXC} = \text{total number of points in the pattern minus total number of excluded points}$). 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:

$$R_p = 100 \frac{\sum_{i=1,n} |y_i - y_{c,i}|}{\sum_{i=1,n} y_i} \quad (3.9)$$

Weighted Profile Factor:

$$R_{wp} = 100 \left[\frac{\sum_{i=1,n} w_i |y_i - y_{c,i}|^2}{\sum_{i=1,n} w_i y_i^2} \right]^{1/2} \quad (3.10)$$

Expected Weighted Profile Factor:

$$R_{exp} = 100 \left[\frac{n-p}{\sum_i w_i y_i^2} \right]^{1/2} \quad (3.11)$$

Goodness of fit indicator:

$$S = \frac{R_{wp}}{R_{exp}}$$

Reduced chi-square:

$$\chi_v^2 = \left[\frac{R_{wp}}{R_{exp}} \right]^2 = S^2 \quad (3.12)$$

Bragg Factor:

$$R_B = 100 \frac{\sum_{\mathbf{h}} |I_{obs,\mathbf{h}}' - I_{calc,\mathbf{h}}|}{\sum_{\mathbf{h}} |I_{obs,\mathbf{h}}'|} \quad (3.13)$$

Crystallographic R_F -Factor:

$$R_F = 100 \frac{\sum_{\mathbf{h}} |F_{obs,k}' - F_{calc,\mathbf{h}}|}{\sum_{\mathbf{h}} |F_{obs,\mathbf{h}}'|} \quad (3.14)$$

where $n-p$ is the number of degrees of freedom. The meaning of n has been given above and p is the number of refined parameters.

The experimental points within the *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. In the denominators the quantity y_i is changed to $y_i - b_i$. The magnetic R-factor is defined as the Bragg R_B -factor but is applied to magnetic intensities.

The *observed* integrated intensity ' $I_{obs,\mathbf{h}}'$ ' is in fact calculated from the Rietveld formula:

$$'I_{obs,\mathbf{h}}' = I_{calc,\mathbf{h}} \sum_i \left\{ \frac{\Omega(T_i - T_{\mathbf{h}})(y_i - b_i)}{(y_{c,i} - b_i)} \right\} \quad (3.15)$$

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 reflection \mathbf{h} ($I_{calc,\mathbf{h}}=0$), the observed integrated intensity is also zeroed: ' $I_{obs,\mathbf{h}}'$ '=0, even if it is obvious that ' $I_{obs,\mathbf{h}}'$ ' is non zero from the experimental pattern. This has as a consequence that the reflections with $I_{calc,\mathbf{h}}=0$ do not contribute to the Bragg R-factor.

The observed structure factor appearing in (3.14) is obtained from (3.15) correcting for multiplicity and the Lorentz-polarisation factor using the formula:

$$'F_{obs,\mathbf{h}}' = \sqrt{\frac{'I_{obs,\mathbf{h}}'}{L_{\mathbf{h}}}} \quad (3.16)$$

In other programs the correction $L_{\mathbf{h}}$ is set to 1, so be careful when comparing R_F values.

Although commonly used, the R_p , R_{wp} 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 [A. Antoniadis, J. Berruyer & A. Filhol, Acta Cryst. **A46**, 692 (1990)] is defined as:

$$D = 2 \sum_i \{y_i \ln(y_i / y_{c,i}) - (y_i - y_{c,i})\} \quad (3.17)$$

- 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:

$$Q = D + a \cdot p \quad (3.18)$$

where p 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 $a=\ln(p)$.

- The Durbin-Watson statistic parameters: d and Q_D . The use of these two quantities to assess the quality of the refinement has been advocated by Hill and Flack [R.J. Hill & H.D. Flack, J. Applied Cryst. **20**, 356 (1987)]. This statistic which measures the correlation between adjacent residuals (serial correlation) is defined as:

$$d = \frac{\sum_{i=2}^n \{ [w_i(y_i - y_{c,i}) - w_{i-1}(y_{i-1} - y_{c,i-1})]^2 \}}{\sum_i \{ [w_i(y_i - y_{c,i})]^2 \}} \quad (3.19)$$

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

$$Q_D = 2 \left\{ (n-1)/(n-p) - 3.0901/\sqrt{n+2} \right\} \quad (3.20)$$

Three cases may occur:

- 1) If $d < Q_D$, 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 $Q_D < d < 4 - Q_D$, there is no correlation
- 3) If $d > 4 - Q_D$, there is negative serial correlation: successive values of the residuals tend to have opposite sign.

Background

The background intensity y_{bi} at the i th step is obtained (see [Nba](#) on [LINE 4](#) and the lines [LINE 10](#) or [LINE 17](#)) either from an user-supplied table of background intensities, or from a refinable background functions:

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

$$y_{bi} = \sum_{j=0}^{11} B_j \cos\left(j \frac{T_i}{C}\right) \quad (3.4')$$

The origin of the background polynomial in (3.4) is given by a selectable input parameter [Bkpos](#) ([LINE 8](#)) and should be supplied by the user. The second sum is used only if [Nba=-1](#) (see appendix for details). The parameters to be refined are: $B_0, B_1, \dots, B_{11}, B_{C1}, \dots, B_{C6}, r_1, \dots, r_6$. Q_i is the modulus of the scattering vector ($\mathbf{Q} = 2\pi\mathbf{h}$) at the i th step, and is given by:

$$Q_i = 4\pi \frac{\sin \theta_i}{\lambda} \text{ or } Q_i = \frac{2\pi}{d_i} \quad (3.5)$$

where $\lambda = \lambda_1$ in the case of two different wavelength contributing to the diffraction pattern. The parameters r_j are distances in Å.

The 12-terms Fourier cosine series (3.4') contains a constant C that depends on the type of powder diffraction pattern in use. For constant wavelength (2θ scale) $C=180/\pi$ and for TOF or energy dispersive techniques $C=T_{\max}$.

When the user provides a table with values of the scattering variable and the corresponding background value there are two possibilities: 1) use the given values for the background with linear interpolation for intermediate values 2) refine the individual background points together with the linear interpolation. For using this last option the user must provide codewords to refine the values.

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:

$$\text{Background}(2\theta) = b_1 \text{Backg} [(1 + b_3)2\theta + b_4] + b_2 \quad (3.6)$$

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 (2θ). 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.

Another 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 help of the formula:

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

where $\text{Filtered}[y_{\text{obs}} - y_{\text{cal}}]$ is a strongly smoothed version of $y_{\text{obs}} - y_{\text{cal}}$. 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. If the [FWINDOW](#) parameter is too small the calculated background enters into the regions inside Bragg peaks giving rise to a wrong estimation of structural parameters. This option is only justifiable in cases of a very wavy background. The starting background should be close to the real one.

The use of refinable background points in a user provided list may behave similarly to the Fourier filtering technique if there is a large number of background points.

Peak-shape functions for constant wavelength data

The profile function Ω is selected by the control variable [Npr \(LINE 2\)](#). 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 [Npr](#), is the following:

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

where:

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

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

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

where:

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

The integral breadth of the Lorentzian function is: $\beta_L = \frac{1}{a_L} = \frac{\pi H}{2}$

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

where:

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

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

where:

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

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

where:

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

$$\text{PSEUDO-VOIGT} : \Omega_5(x) = pV(x) = \eta L'(x) + (1-\eta)G'(x) \quad 0 \leq \eta \leq 1 \quad (3.13)$$

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

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

$$\beta_{pV} = \frac{\pi H/2}{\eta + (1-\eta)\sqrt{\pi \ln 2}}$$

The profile function $\text{Npr}=4$, $\Omega_4(x)$, is a superposition of three $pV(x)$ functions (see below (3.22)). The difference between $\text{Npr}=5$ and $\text{Npr}=7$ is that in the later the parameter η 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:

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

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 (η, 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 (β_L) and Gaussian components (β_G):

$$V(x) = V(x, H_L, H_G) = V(x, \beta_L, \beta_G)$$

$$V(x) = \beta_G^{-1} \text{Re} \left[\text{erf} \left(\frac{\sqrt{\pi}}{\beta_G} |x| + i \frac{\beta_L}{\beta_G \sqrt{\pi}} \right) \right] \quad (3.15)$$

where:

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

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

$$(H, \eta) = F(H_G, H_L)$$

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

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

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

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

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

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

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

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

The dependence of the peak-shape parameters (including FWHM) on the scattering variable (2θ 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 $\text{Npr}=0$ to $\text{Npr}=6$:

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

For $\text{Npr}=5$ (pseudo-Voigt) the η parameter can be dependent on X through the formula:

$$\eta = \eta_0 + X 2\theta \quad (3.21)$$

For $\text{Npr}=4$ (tripled pseudo-Voigt), the three components are assumed to have the same η_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:

$$D = \frac{S_{h1}}{d \cos \theta}$$

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 η_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 $\text{Npr}=6$ (Pearson-VII) the m parameter can be dependent on X and Y through the formula:

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

For $\text{Npr}=7$, the FWHM of the Gaussian (H_G) and Lorentzian (H_L) components is calculated as:

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

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

The units of parameters U, V, W, I_G are $(degrees\ 2\theta)^2$. 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:

$$FWHM^2 = (U + DST^2) \tan^2 \theta + V \tan \theta + W + IG / \cos^2 \theta$$

For $Npr = 4$ (tripled pseudo-Voigt), the three components are assumed to have the same η_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

$$D = Shape1 / d \cos \theta$$

$$pV(x) = \eta_0 L(x) + (1 - \eta_0) G(x)$$

So, apart from the FWHM that is calculated from [U, V, W](#), DST and [GausSiz](#) parameters for a single component, the profile function has FOUR shape parameters η_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 $Npr = 5$ and 12 (pseudo-Voigt) the η parameter can be dependent on [X](#) through the formula:

$$pV(x) = \eta L(x) + (1 - \eta) G(x)$$

$$\eta = \eta_0 + X \cdot 2\theta$$

For $Npr = 11$ (split pseudo-Voigt) DST and [GausSiz](#) 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 $Npr=11$ is

$$FWHM^2(L) = (Ul + DST^2) \tan^2 \theta + Vl \tan \theta + Wl + IG / \cos^2 \theta + Shape1 / \tan^2 2\theta$$

[Shape1](#) is applied only for $2\theta \leq 90^\circ$. The value of η for the left part is given by:

$$\eta_l = \eta_0 + Xl \cdot 2\theta$$

The expression of the right part is:

$$FWHM^2(R) = (Ur + DST^2) \tan^2 \theta + Vr \tan \theta + Wr + IG / \cos^2 \theta + Shape2 / \tan^2 2\theta$$

Shape2 is applied only for $2\theta > 90^\circ$. The value of η for the right part is given by:

$$\eta_r = \eta_0 + Xr \cdot 2\theta$$

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

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

$$m = m_0 + 100X / 2\theta + 1000Y / 2\theta^2$$

For $N_{pr} = 7$, the FWHM of the two components is calculated as

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

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

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

New formulae:

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

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

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

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

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

$$(H_S, \eta_S) = F(H_{GS}, H_{LS})$$

Substituting the expressions of the G and L size we obtain similar expressions where H_G and H_L are replaced by

$\text{SQRT}(\text{GausSiz}) = I_G^{1/2}$ and $Y_L = Y + \text{LorSiz}$ respectively. The term in $1/\cos \theta$ cancels out if we call $H_S = Z_S / \cos \theta$

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

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

From Z_S and η_S we obtain the integral breadth corresponding to size effects just from the corresponding expression for the normalised pseudo-Voigt function.

$$\beta_{PV}(\text{Size}) = \frac{\pi Z_S / 2}{\eta_S + (1 - \eta_S) \sqrt{\pi \ln 2}} \times \left(\frac{\pi}{180} \right) \times \frac{1}{\cos \theta}$$

If Z_S and η_S have been calculated from the values of GausSiz 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:

$$\beta_h = \beta_{PV}(\text{size}) = \frac{\lambda}{D_h \cos \theta} = \frac{\pi Z_S / 2}{\eta_S + (1 - \eta_S) \sqrt{\pi \ln 2}} \times \left(\frac{\pi}{180} \right) \times \frac{1}{\cos \theta}$$

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:

$$D_h = \frac{\eta_s + (1 - \eta_s)\sqrt{\pi \ln 2}}{Z_s} \times \left(\frac{360\lambda}{\pi^2}\right)$$

Where Z_s and η_s have been obtained from $I_G = \text{GausSiz}$ and $Y_L = Y + \text{LorSiz}$ using the TCH formula.

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

Let us call the non-normalised function as:

$$pV_n(x) = \eta_n L_n(x) + (1 - \eta_n) G_n(x) \quad pV_n(0) = 1$$

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:

$$\beta_{pVn} = \frac{\int \Omega(x) dx}{\Omega(0)} = \frac{\int I_0 pV_n(x) dx}{I_0 pV_n(0)} = \frac{\int pV_n(x) dx}{1}$$

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

$$\beta_{pVn} = \eta_n \frac{1}{a_L} + (1 - \eta_n) \frac{1}{a_G} = \eta_n \frac{\pi H}{2} + (1 - \eta_n) \frac{H}{2} \sqrt{\frac{\pi}{\ln 2}}$$

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 breadth, so the numerical relation between the η values is given by:

$$\text{Non-normalised: } \beta_{pVn} = \eta_n \frac{\pi H}{2} + (1 - \eta_n) \frac{H}{2} \sqrt{\frac{\pi}{\ln 2}}$$

$$\text{Normalised: } \beta_{pV} = \frac{\pi H/2}{\eta + (1 - \eta)\sqrt{\pi \ln 2}}$$

$$\eta_n \frac{\pi H}{2} + (1 - \eta_n) \frac{\pi H}{2\sqrt{\pi \ln 2}} = \frac{\pi H/2}{\eta + (1 - \eta)\sqrt{\pi \ln 2}}$$

$$\eta_n = \frac{\eta}{\eta + (1 - \eta)\sqrt{\pi \ln 2}}$$

$$\eta = \frac{\eta_n}{\eta_n + (1 - \eta_n) \frac{1}{\sqrt{\pi \ln 2}}}$$

Peak-shape functions for Neutron Time of Flight

The profile function Ω for TOF corresponds to the control variable $N_{pr}=9, 10$. 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.

Monochromator, Lorentz and geometrical corrections

Bragg-Brentano or Debye-Scherrer geometry

Monochromator polarisation correction (C_{thm} and K in LINE 4); the Lorentz polarisation factor L_p is calculated as :

$$L_p = \frac{1 - K + K \cos^2 2\theta_{Monok} \cos^2 2\theta}{2 \sin^2 \theta \cos \theta} \quad (3.26)$$

with $CTHM = \cos^2 2\theta_{Monok}$. For instance with a Graphite monochromator, $C_{thm} = 0.8351$ and 0.7998 for CuK_{β} and CuK_{α} respectively.

Note that K is used only for synchrotron data ($I_{lo}=3$) 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, $I_{lo}=2$) the Lorentz factor is:

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

and the polarisation correction is given by:

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

where

$$c_1 = |\cos 2\theta_{Monok}| = \sqrt{CTHM}$$
$$c_2 = \cos^2 2\theta_{Monok} = CTHM$$

K = fraction of perfect crystal contribution

Asymmetry

The default asymmetry correction is a multiplier term to the peak shape. The asymmetry correction adopts the form:

$$A_s(z) = 1 + \frac{P_1 F_a(z) + P_2 F_b(z)}{\tanh \theta_h} + \frac{P_3 F_a(z) + P_4 F_b(z)}{\tanh 2\theta_h} \quad (3.29)$$

where

$$z = \frac{2\theta_i - 2\theta_h - S_{shf}}{FWHM}$$

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 $Nph \leq 0$ 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 $Npr=5$ and $Npr=7$ are implemented for this correction) calculated at displaced points:

$$\Omega(x) = \frac{1}{90} \{7g(x) + 32g(x + c_2 P) + 12g(x + c_3 P) + 32g(x + c_4 P) + 7g(x + c_5 P)\} \quad (3.30)$$

where $x = 2\theta_i - 2\theta_h - S_{shf}$, P is the asymmetry parameter and $\cot 2\theta = 16c_2 = 4c_3 = \frac{16}{9}c_4 = c_5$.

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:

- $Nor=0 \rightarrow$ the exponential function:

$$P_h = G_2 + (1 - G_2) \exp(G_1 \alpha_h^2) \quad (3.31)$$

Where G_1 and G_2 are refinable parameters and α_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 [Pr1](#) [Pr2](#) [Pr3](#) specified in Line 11-2.

- [Nor](#)=1 → Modified March's function:

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

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

- $G_1 < 1$: Platy habit (α 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 (α 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:

$$T_{\mathbf{h}} = \exp \left\{ -(1.7133 - 0.0368 \sin^2 \theta_{\mathbf{h}}) \mu R + (0.0927 + 0.375 \sin^2 \theta_{\mathbf{h}}) (\mu R)^2 \right\} \quad (3.33)$$

Where μ is the linear absorption coefficient and R the radius of the cylindrical sample.

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

$$T_{\mathbf{h}} = \frac{\exp(-mt / \cos \theta_{\mathbf{h}})}{\cos \theta_{\mathbf{h}}} \quad (3.34)$$

Where t is the "effective" thickness of the sample. The product mt is given in the same place as μR .

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

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

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

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

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 $\cos\theta$ and $\sin\theta$ 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 $\theta-2\theta$ 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:

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

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 2θ 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:

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

The transparency correction is given by the relation:

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

where μ 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:

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

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 2θ 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:

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

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:

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

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:

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

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:

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

where R is the radius of the CPSD and α the incident beam angle (in degrees) at sample surface. Thus, the parameter refined by **FullProf** is $SYSIN = -s/(R \sin \alpha)$. 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 2θ angle; the value of the sample offset is given by:

$$s = \frac{\pi}{180} R \sin \alpha SYSIN$$

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 \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:

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

where β_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:

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

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 $\beta_{\alpha\beta j}$ should be set to zero. The β s have no direct physical meaning. They have to be converted to a Cartesian 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 $(\exp\{-B_{ov} |\mathbf{h}|/2\})$ 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 $F_{\mathbf{h}}^2$ is calculated using the general formula of Halpern and Johnson:

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

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:

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

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 $p = (r_e \gamma / 2)$ is 0.2695 and allows the conversion of the Fourier components of magnetic moments, given in Bohr magnetons μ_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 $\mathbf{h} = \mathbf{H} + \mathbf{k}$:

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

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 ($\beta_{\alpha\alpha}$), but it is not generally necessary to use these terms in magnetic refinements. The phase factor $\psi_{\mathbf{k}js}$ has two components:

$$\psi_{\mathbf{k}js} = \Phi_{\mathbf{k}j} + \phi_{\mathbf{k}js} \quad (3.51)$$

$\Phi_{\mathbf{k}j}$ 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. $\phi_{\mathbf{k}js}$ 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

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

The sign of ϕ_{kj} 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 2π . 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:

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

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 $Hel=1$ is used, the number of free parameters per magnetic atom is reduced. The Fourier coefficients are considered of the form:

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

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:

$$\mu_{lj} = \sum_{\{\mathbf{k}\}} \mathbf{S}_{kj} \exp \{ -2\pi i \mathbf{k} \mathbf{R}_l \}$$

$$\mu_{lj} = \sum_{\{\mathbf{k}\}} \left\{ \mathbf{R}_{kj} \cos 2\pi \left[\mathbf{k} \mathbf{R}_l + \psi_{kj} \right] + \mathbf{I}_{kj} \sin 2\pi \left[\mathbf{k} \mathbf{R}_l + \psi_{kj} \right] \right\}$$

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.

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.

$$R_p = 100 \sum_i |y_{obs,i} - y_{calc,i}| / \sum_i |y_{obs,i}| \quad (3.68)$$

where $y_{calc,i}$ is calculated with the help of a Poissonian function, $Genpoi(x)$, from $y_{obs,i}$. 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 $y_{obs,i}$ and $var_{obs,i}$ as follows:

$$c_1 = y_{obs,i} / var(y_{obs,i})$$

$$x = y_{obs,i} c_1$$

$$y_{calc,i} = Genpoi(x) / c_1$$

- 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:

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

"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:

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

where Sum(i) is a sum over all the reflections contributing to the allowed areas of the diffraction pattern. Sum(ni) is the sum extended to the reflections near to the reflection "i" (including this reflection). The symbol x[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 Enref(Iphase): Effective number of reflections of the phase Iphase.

The program calculates these numbers (Enref's) and the ratios (Refni's) for three values of the parameter p (1, 1/2 and 1/4).

SINGLE CRYSTAL AND INTEGRATED INTENSITY REFINEMENTS

General comments

FullProf permits the refinement of integrated intensity data. Single crystal and/or powder integrated intensities can be included (or used alone) as observations for refining a structural model. The structure factor calculation is exactly the same as in powder diffraction, except that for powder diffraction there is a simplification due to the fact that the reflections \mathbf{h} and $-\mathbf{h}$ are always overlapped, so all the available features can be directly used with integrated intensity data.

The function optimised when using integrated intensities is:

$$M = \sum_n w_n (G_{obs,n}^2 - \sum_k G_{calc,k}^2)^2$$

The index n runs over the observations (1,.. N_{obs}). The index k runs over the reflections contributing to the observation n . G^2 is the square of the structure factor (intensity corrected for Lp -factor). In case of powder diffraction $G_k^2 = I_k = (jLpF^2)_k$, so clusters of integrated intensities are used.

The R-factors in single crystal work are calculated according to the following formulae:

RF² -factor:

$$R_{F^2} = 100 \frac{\sum_n \left| G_{obs,n}^2 - \sum_k G_{calc,k}^2 \right|}{\sum_n G_{obs,n}^2}$$

RF²w-factor:

$$R_{wF^2} = 100 \sqrt{\frac{M}{\sum_n w_n G_{obs,n}^2}}$$

RF-factor:

$$R_F = 100 \frac{\sum_n \left| G_{obs,n} - \sqrt{\sum_k G_{calc,k}^2} \right|}{\sum_n G_{obs,n}}$$

$$G_{obs,n} = \sqrt{G_{obs}^2}$$
$$\chi^2(I) = \frac{M}{N_{obs} - p}$$

Notice that RF2w for w(n)=1.0 is not the same value as RF2.

When the option `lwg=2` (unit weights) is used. The inverse of the normal matrix is not multiplied by Chi2 as is usual in weighted refinements.

The extinction correction

At present only the "empirical correction" (a single parameter), as used in SHELX, has been implemented. I hope to have time for putting the anisotropic Becker-Coppens extinction coefficients into the program.

Mixed refinements

This option is still in an exploring stage. For the moment only a single powder diffraction profile can be used with different sets of integrated intensity data that are related to each phase. A global powder diffraction pattern can be given as primary information, and some "phases" can be given in addition with their own integrated intensity data. For mixing X-ray and neutron refinement, the best thing to do is to provide a "powder" neutron diffraction pattern as the global data information and a set of integrated intensities for the crystallographic phase to be refined. At present it is necessary to "repeat" the same phase, as in that modelling the neutron pattern, using the appropriate constraints in the refined parameters (using the same code for physically identical parameters).

APPENDIX

***DETAILED DESCRIPTION OF INPUT
FILES FOR FullProf***

Introduction

In this appendix we describe the following input files:

- Input control file: **CODFIL.pcr**
- Input data files: **FILE.dat**
- Background file: **CODFIL.bac** or **FILE.bac**
- Reflection files: **CODFILn.hkl**, **hkln.hkl**, **CODFIL.int**
- Instrument Resolution File: **MYRESOL.irf**
- Numerical profile file: **CODFIL.shp** or **global.shp**
- Intensity correction file: **CODFIL.cor**

CODFIL.PCR

This file is free format. This does not mean free format in Fortran (*)-sense: a procedure interprets the items given by the user that must obey the order given below. A space is needed between each item (except when the second is a negative number). When the program is run, messages of error reading a line of this file are normally due to a previous error. For example, the number of atoms you really wrote does not correspond to the number you put in the line following the name of the phase. Empty lines as well as lines starting with the symbol “!” in the first column are considered as comments and are ignored by the program. This file is re-written by *FullProf* at the end of a job. The original file may be suppressed or conserved depending on the value of a flag. The program generated PCR-file contains comments (mnemonics for each variable) to facilitate the recognition of the different variables.

If the user starts his (her) CODFIL.PCR file with the left-adjusted capital “COMM”, the new CODFIL.PCR file conserves the initial format; that is the usual single pattern format, otherwise the program generates a PCR-file with a format adapted to the multiple pattern refinement. If the user introduce his(her) own comments, they are not saved in the new version of the file. The best to start working with *FullProf* is to edit a previously existent PCR-file and modify it as wished. The inexperienced user can use some of the programs that are specially developed to edit a PCR-file (see for instance the program AXES by Hugo Mandar [Hugo Ma'ndar <hugo@physic.ut.ee>]). A dedicated tool will be soon available for the operating systems Windows and Linux.

In the following we use the term LINE as a label for reference purposes in the document. The line number does not correspond to the effective ordinal number of the line appearing in an actual PCR-file. When a block of similar lines have to be repeated for each pattern (it is supposed that we are dealing with NPATT patterns simultaneously) the keyword NPATT Lines starts the block and the block is finished by the keyword *End* NPATT Lines, this corresponds to a loop on the index of the pattern. Sometimes the number of lines is less than NPATT. This occurs when the line is optional and depends on the value of a pattern-dependent variable.

In this document we shall describe the format of the PCR-file adapted to multipattern refinements. This format derives from the single pattern one by just splitting some lines where flags depending on the pattern number were written in a single line. The program is, and will continue to be, totally compatible with the old single pattern format even if new items corresponding to new options are concerned. When there are differences between the two formats we will give the lines of the single pattern format in **black font** preceded with the word **LINE n** together with the new ones in **blue font** starting with **LINE n**. In fact the places where differences, apart from the fact that the loops on the number of patterns reduce to 1, are the **LINES 2, 3, 4, 5, 7, 9, 19, 19-1, 19-3, and 19-4**.

Sometimes we shall use mathematical formulas to clarify the meaning of a variable, the complete explanation is given in the Mathematical section of the *FullProf* manual. The scattering variable is represented, in general, by the symbol T . When the context requires an explicit reference we use 2θ , TOF or E . For multiple patterns refinement we shall use the variable n_{pat} to index a diffraction pattern. We will write **XXX(n_{pat})** for the name of a variable **XXX** that depends of the diffraction pattern n_{pat} . When the context is clear the variable will be referenced simply as **XXX**.

The content of the file CODFIL.pcr is described in the following.

Title+Choice of the format: multipattern/single-pattern

(LINE1-3: Line 1 is compulsory, Lines 2-3 are **optional**)

LINE1: Title line, made of any 80 characters, and used to label the printout. TITL and/or COMM (for single pattern format) at the beginning of the title line, are reserved keywords.

Examples:

COMM My sample, at this temperature : (old) Single pattern format

My sample, at this temperature : Mutipattern format

If the first four character of TITLE correspond to the word TITL the file is given in "command mode" (not available yet). If the first four characters of TITLE correspond to **COMM**, the new CODFIL.PCR file conserves the initial format; that is the usual single pattern format, otherwise the program generates a PCR-file with a format adapted to the multiple pattern refinement.

The generated PCR-file include comment lines give a keyword for each variable in order to be easily recognised by the user. This comment line has been included below to facilitate the recognition of the different lines.

The next two lines should not be given if one wishes to use the conventional single pattern format of the PCR-file. However, the new multi-pattern format works properly also for a single pattern.

LINE2: Keyword NPATT (OPTIONAL, if COMM is NOT given).

Example:

NPATT 3

(5 characters +1 integer)

Keyword NPATT followed by an integer, NPATT, corresponding to the number of patterns to be treated simultaneously.

This line may be suppressed for a single pattern (NPATT=1) and the rest of the file follows the same format as that of previous versions of **FullProf** for single pattern refinement.

Example:

LINE3: Keyword W_PAT (OPTIONAL), only if NPATT is provided)

Example:

W_PAT 0.5 0.25 0.25

(5 characters + NPATT reals)

Keyword W_PAT followed by NPATT reals corresponding to the weight of each pattern in the refinement. The program normalises the given values in order to get 1 for the sum of all weights.

Control flags defining the type of the job

(LINE4 is compulsory)

This line contains the codes used to select the type of the job: combinations of the given codes allow a number of different possibilities for the user. Important codes to handle the job are the following.

For multipattern powder work, and for each pattern (patterns are, for the moment describing powder data only!):

The calculated or the experimental data to be refined can be done for X-ray or Neutron radiation (selected through [Job](#)). These patterns can have possibly different scattering variables (selected via [Uni](#)). The number of magnetic or structural phases needed to model the diffraction patterns is given in [Nph](#). Also important, if the “pattern matching” mode will have to be used for one of the phases (see [Jbt](#) in the [line 19](#)), [Dum](#) have to be set to 1.

To perform single crystal work:

[Cry](#) is used to select if the main job will handle a list of integrated intensities (typically for single crystal work) or only powder data. If it is the case, the [Irf](#) code in the [line 19](#) of the specific phase(s) needed to describe the single crystal data will have to be set to 4. If a single crystal work is to be done, the different values of [Cry](#) allows selecting among different methods of refinement.

For mixed single crystal /powders refinement, see the section [mixed refinements](#) in “[SINGLE CRYSTAL AND INTEGRATED INTENSITY REFINEMENTS](#)”.

The other codes are used, depending on the setting of the important codes listed above. Their meaning is detailed below.

Input format

Multi-Pattern:

A specific line (4n) should be given for each pattern

Comment line: ! Nph Dum Ias Nre Cry Opt Aut

LINE 4 [Common line]: [Nph](#), [Dum](#), [Ias](#), [Nre](#), [Cry](#), [Opt](#), [Aut](#) (6 integers)

Comment line: ! Job Npr Nba Nex Nsc Nor Iwg Ilo Res Ste Uni Cor

LINE 4n [NPATT_Lines]: [Job](#), [Npr](#), [Nba](#), [Nex](#), [Nsc](#), [Nor](#), [Iwg](#), [Ilo](#),
[Res](#), [Ste](#), [Uni](#), [Cor](#), (12 integers)

[(Old) Single pattern format]

The items of the different multi-pattern lines 4 and 4n are merged in a single line for the single patter format as:

Comment line: !Job Npr Nph Nba Nex Nsc Nor Dum Iwg Ilo Ias Res Ste Nre Cry Uni Cor Opt Aut

LINE 4: [Job](#), [Npr](#), [Nph](#), [Nba](#), [Nex](#), [Nsc](#), [Nor](#), [Dum](#), [Iwg](#), [Ilo](#),
[Ias](#), [Res](#), [Ste](#), [Nre](#), [Cry](#), [Uni](#), [Cor](#), [Opt](#), [Aut](#) (19 integers)

Common flags for Single/Multi-Pattern

Nph - Number of phases

Number of phases

Dum – Control of the divergence for specific jobs, ex: profile matching

- =1 If equal to 1 and some of the phases are treated with Profile Matching modes, the criterion of convergence when shifts are lower than a fraction of standard deviations is not applied.
- =2 If equal to 2, the program is stopped in case of local divergence: $\text{chi2}(\text{icycle}+1) > \text{chi2}(\text{icycle})$
- =3 If equal to 3 the reflections near excluded regions ($T_{\text{lim}} \pm \text{Wdt} * \text{FWHM}$) are not taken into account to calculate the Bragg R-factor. These reflections are omitted in the *output files* with hkl's.

If $\text{ABS}(\text{Job}(n_pat)) > 1$ (pattern calculation mode, see below) and **Dum** is different of zero a file CODFIL.sim is generated

Ias – Reordering of reflections

- =0 The reordering of all reflections is performed only at the first cycle
- =1 All reflections are ordered at each cycle.
(If **Jbt**=2 for one phase, **Ias** is changed to 1 by the program)

Nre – Number of constrained parameters

Number of parameters to be constrained within given limits. At the end of the file you must give a list of **Nre** lines specifying the number and the limit of each parameter. This variable must be given in the case of using Montecarlo or Simulated Annealing techniques.

Cry – Single crystal job and refinement algorithm type

- ≠0 Only integrated intensity data will be given. No profile parameters are needed. The format of the file changes slightly in the following.
- =1 Refinement of single crystal data or integrated intensity powder data.
- =2 No least-squares algorithm is applied. Instead a Montecarlo search of the starting configuration is performed. A selected number of parameters **Nre** are moved within a box defined by the **Nre** relations fixing the allowed values of the parameters. The best (lowest R-factor) **NSOLU** solutions are printed and the CODFIL.pcr file is updated with the best solution. This option is only efficient for a small number of parameters (3-4). The use of the next option is recommended for large number of parameters.
- =3 The Simulated Annealing optimisation method is chosen. A selected number of parameters **Nre** are moved within a box defined by the **Nre** relations fixing the allowed values of the parameters. Different boundary conditions may be used. See below.

Opt – Calculation optimisations

- =0 The general procedures are used.
- =1 The program tries to optimise the calculations looking for the particular options used in the job. In some cases the calculation proceeds up to a 30% faster.

Aut – Automatic mode for the refinement codes numbering

- =0 The program treats the codewords as usual. The user has the total control of the numbering of parameters. The maximum number of parameters to be refined is fixed manually.
- =1 The program treats the codewords of the refined parameters automatically. In this case the user may put, by hand, the codes for making constraints as usual, and just put "1.00" to inform the program that the corresponding parameter will be refined. The program

will attribute automatically the codeword. In the automatic mode there is no “holes in the matrix” and the number of refined parameters may be different from the specified by the user in the corresponding line. The automatic mode is useful when one has to fix a parameter in the middle of many others: just put the codeword (including the multiplier) to zero. Sometimes the message “hole in the matrix” still appears. In such cases you have just to increase artificially the number of parameters to be refined or just put it equal to zero, or, in the worst case, suppress a large number of codewords (just leaving the multiplier 1.0). Be careful in using this option together with the automatic re-writing of the PCR-file.

Flags dependent on the pattern number

Job – Select the simulation or refinement mode and the type of the radiation X-ray, CW or T.O.F neutrons

- = 0 X-ray case
- = 1 Neutron case (constant wavelength, nuclear and magnetic)
- = 2 Pattern calculation (X-ray)
- = 3 Pattern calculation (Neutron, constant wavelength)
- =-1 Neutron case (T.O.F., nuclear and magnetic)
- =-3 Pattern calculation (Neutron, T.O.F.)

The value of **Job**(*n_pat*) may be different for each pattern when one wants to perform combined refinements: X-ray + neutron diffraction patterns treated simultaneously.

If **ABS**(**Job**)>1 and **Dum**=1 (see below) a calculated pattern is created with the name CODFIL.SIM in format corresponding to **Ins**(*n_pat*)=0. A Poissonian noise is added to the deterministic calculated pattern. The statistics is controlled by the value of the scale factor. This pattern corresponds to an “ideal observed” pattern and can be use for simulation purposes in order to investigate the effect of systematic errors on the structural parameters and on the reliability factors.

Npr – Defaut profile to be used

Default value for selection of a normalised peak shape function. Particular values can be given for each phase, in that case the local value is used.

- =0 Gaussian.
- =1 Cauchy (Lorentzian).
- =2 Modified 1 Lorentzian.
- =3 Modified 2 Lorentzian.
- =4 Tripled pseudo-Voigt.
- =5 pseudo-Voigt.
- =6 Pearson VII.
- =7 Thompson-Cox-Hastings pseudo-Voigt convoluted with axial divergence asymmetry function (Finger, Cox & Jephcoat, J. Appl. Cryst. 27, 892, 1994).
- =8 Numerical profile given in CODFIL.shp or in GLOBAL.shp.
- =9 T.O.F. Convolution pseudo-Voigt with back-to-back exponential functions.
- =10 T.O.F. Same as 9 but a different dependence of TOF versus d-spacing.
- =11 Split pseudo-Voigt function.
- =12 Pseudo-Voigt function convoluted with axial divergence asymmetry function.
- =13 T.O.F. Pseudo-Voigt function convoluted with Ikeda-Carpenter function.

Nba – Background type

- =0 Refine background with a polynomial function.
- =1 Read background from file CODFIL.bac. The format of this file is explained in this appendix.
- =2,3,...,N Linear interpolation between the N given points. If **Nba**<0 but **ABS**(**Nba**)>4 the interpolation is performed using cubic splines
- =-1 Refine background with Debye-like + polynomial function.

- =2 Background treated iteratively by using a Fourier filtering technique. An extra parameter is read below. The starting background is read from file FILE.bac as for [Nba](#)=1.
- =3 Read 6 additional polynomial background coefficients

[Nex](#) – Number of regions to exclude in powder data

Number of excluded regions.

[Nsc](#) – Number of user defined scattering factors

Number of scattering sets (zero in most cases). In the case of giving a table of values for the scattering factor and [Nsc](#)>0, the program performs an internal fit in order to get the appropriate coefficients of the exponential expansion (see below) approximating the scattering factor. If [Nsc](#) is negative, a linear interpolation between the values of the table is performed.

[Nor](#) – Preferred orientation function type

- =0 Preferred orientation function No 1
- =1 Preferred orientation function No 2 (March)

[Iwg](#)- Refinement weighting scheme

- =0 Standard least squares refinement
- =1 Maximum likelihood refinement
- =2 Unit weights

[Ilo](#) – Lorentz and polarization corrections

- =0 Standard Debye-Scherrer geometry, or Bragg-Brentano if the illuminated area does not exceed the sample surface. If Bragg-Brentano geometry is used but the above condition is not fulfilled, the intensity data must be corrected for the geometric effect before attempting any refinement. A partial correction can be performed by using the parameter [Sent0](#).
- =1 Flat plate PSD geometry
- =-1 The Lorentz-Polarisation correction is not performed. It is supposed that the profile has been previously corrected for Lorentz-Polarisation.
- =2 Transmission geometry. Flat plate with the scattering vector within the plate (Stoe geometry for X-rays)
- =3 Special polarisation correction is applied even if the format of the DAT-file does not correspond to one of the synchrotron explicitly given formats (see below). This must be used for synchrotron data given in a (X, Y, Sigma) format ([Ins](#)=10).

[Res](#) – Resolution function type

- =0 Resolution function of the instrument is not given
- ≠0 The next [line](#) contains the name of the file where the instrumental resolution function is given for an instrument using as scattering variable 2θ . This options works, at present, only for constant wavelength type of data. The profile is assumed to be a Voigt function ([Npr](#)=7). 12 parameters or a table determine the resolution function. $U_i, V_i, W_i, X_i, Y_i, Z_i$ ($i=1,2$ for λ_1 and λ_2)

The different types of functions are:

$$\text{Res}=1 \quad H_G^2 = (U_i \tan \theta + V_i) \tan \theta + W_i$$

$$H_L = X_i \tan \theta + \frac{Y_i}{\cos \theta} + Z_i$$

$$\text{Res}=2 \quad H_G^2 = (U_i \tan \theta + V_i) \tan \theta + W_i$$

$$H_L = (X_i 2\theta + Y_i) 2\theta + Z_i$$

$$\text{Res}=3 \quad H_G^2 = (U_i 2\theta + V_i) 2\theta + W_i$$

$$H_L = (X_i 2\theta + Y_i) 2\theta + Z_i$$

Res =4 List of values 2θ , $H_G(2\theta)$, $H_L(2\theta)$
(a linear interpolation is applied for intermediate 2θ)

The format of this file is described below in this [appendix](#).

Ste – Number of data points reduction factor in powder data

=1,2,3,...,N

If **Ste**>1 the number of data points is reduced by a factor of **Ste**. Only those points corresponding to the new **Step** size **Ste**×**Step** (see [LINE 9](#) below) are taken into account in the refinement. Useful for speed-up preliminary refinements.

Uni - Scattering variable unit

=0 2θ in degrees
=1 T.O.F. in micro-seconds
=2 Energy in keV.

Cor – Intensity correction

=0 No correction is applied
=1 A file with intensity corrections is read.
=2 A similar file is read but the coefficients of an empirical function and their standard deviations are read instead of directly the corrections.
The format of this file is described in this appendix.

Name of the data files

(LINE5 is compulsory)

Comment line: ! File names of data (patterns) files

Line 5: [NPATT_Lines] FILEDAT(*n_pat*) (Character)

Example:

Mydatafile_patt1.dat

Mydatafile_patt2.dat...

Name of the pattern files including the extension in case that is different from “.dat”

Name of the instrumental resolution files

(LINE6 is optional, read only if **Res** ≠0 on [LINE 4](#))

Comment line: ! Resolution File names for Pattern #N

Line 6: [NPATT_Lines] FILERES(*n_pat*) (Character)

Example:

Myresolutionfile_patt1.irf

Myresolutionfile_patt2.irf...

Name of the file with the instrumental resolution function. To be given only in the case of **Res** ≠0. The items in **FILERES** are read in free format. The format of this file is described below in this appendix.

The effective number of lines may be less than **NPATT**. Only the names of the files for which **Res**(*n_pat*)≠0 should be given.

Control flags handling the input CODEFIL.pcr, *.dat and output files

(LINE7 is compulsory)

Input format

Multi-Pattern:

A specific line (7n) should be given for each pattern

Comment line: ! Mat Pcr Syo Rpa Sym Sho

LINE 7[Common line]: Mat, Pcr, Syo, Rpa, Sym, Sho (6 integers)

Comment line: ! Ipr Ppl Ioc Ls1 Ls2 Ls3 Prf Ins Hkl Fou Ana

LINE 7n [NPATT Lines]: Ipr, Ppl, Ioc, Ls1, Ls2, Ls3, Prf, Ins, Hkl, Fou, Ana
(11 integers)

[(Old) Single pattern format]

The items of the different multi-pattern lines7 and 7n are merged in a single line for the single patten format as:

Comment line: !Ipr Ppl Ioc Mat Pcr Ls1 Ls2 Ls3 Syo Prf Ins Rpa Sym Hkl Fou Sho Ana

LINE 7: Ipr, Ppl, Ioc, Mat, Pcr, Ls1, Ls2, Ls3, Syo, Prf, Ins, Rpa, Sym, Hkl, Fou,
Sho, Ana (17 integers)

Common flags for Single/Multi-Pattern

Mat – Correlation matrix output

- =0 No action
- =1 The correlation matrix is written in the file CODFIL.out
- =2 The diagonal of Least Squares matrix is printed before inversion at every cycle.

Pcr – Update of the .pcr after refinement

- =0
- =1 CODFIL.pcr is re-written with updated parameters
- =2 A new input file is generated conserving the old one. The new file is called CODFIL.new

Syo – Output of the symmetry operators

- =0
- =1 Symmetry operators are written in CODFIL.out. The file CODFIL.sym is also generated if Sym=1.

Rpa – Output .rpa/.sav file

- =0
- =1 Prepares the output file CODFIL.rpa. If the file exists before running the program the new data are APPENDED.
- =2 Prepares file CODFIL.sav (sequential refinements) "

Sym – Output .sym file

- =0
- =1 Prepares CODFIL.sym (if 1, Syo must be set to 1)

Sho – Reduced output during the refinement

- =0
- =1 Suppress the output from each cycle. Only the information from the last cycle is printed.

Pattern number dependent flags

Ipr – Profile integrated intensities output or generation of .sub files

- =0 No action
- =1 Observed and calculated profile intensities written in CODFIL.out
- =2 The files CODFILn.sub with the calculated profile of each phase are generated.
- =3 As 2 but the background is added to each profile.

Ppl – Various types of calculated output -I

- =0 No action
- =1 Line printer plot in CODFIL.out
- =2 Generates the background-file FILE.bac
- =3 Puts difference pattern in file FILE.bac

Ioc – Various types of calculated output -II

- =0 No action
- =1 List of observed and calculated integrated intensities in CODFIL.out
- =2 The reflections corresponding to the second wavelength are also written if different from the first one.

Ls1 - Various types of calculated output -III

- =0 No action
- =1 Reflection list before starting cycles is written in CODFIL.out

Ls2 - Various types of calculated output –IV

- =0 No action
- =1 Corrected data list (profile intensities before refinement) written in CODFIL.out.
- =4 In some versions of **FullProf** a plot of the diffraction pattern is displayed on the screen at each cycle of refinement.

Ls3 - Various types of calculated output –V

- =0 No action
- =1 Merged reflection list written in CODFIL.out

Prf - Output format of the rietveld plot file CODFIL.prf

Generates the file CODFIL.prf containing the information to plot the observed versus calculated diffraction pattern as well as the reflection positions, etc. The output format

- =0
- =1 Format suitable for **WinPLOT**R, and other plotting programs.
- =2 Format suitable for IGOR (MacOS, Windows software)
- =3 Format suitable for KaleidaGraph (MacOS, Windows software) and **WinPLOT**R.
- =4 Format suitable for PICTURE, Xvgr (Sun-Unix Software)

Ins – Data file format

(The detailed explanation of the formats is given in the **FILE.dat** section)

- =0 Data supplied in free format. Up to seven comment lines are accepted. The first three real numbers found at the beginning of a line are interpreted as T_i , $step$ and T_f and. The following lines after T_i , $step$ and T_f must contain $NPTS=(T_f-T_i)/step+1$ values of the intensity profile. Data format of TOF raw data from Argonne are also interpreted by this value of **Ins**.
- =1 D1A/D2B format (original Rietveld-Hewat format: the first line must be T_i , $step$ and T_f)
- =2 D1B old format (DEC-10)

- =3 Format corresponding to the ILL instruments D1B and D20.
- =±4 Brookhaven synchrotron data.
 4: First line: $2\theta_i$, step, $2\theta_f$ (free format). Rest of file: pairs of lines with 10 items like
 $Y1\ Y2\ \dots\ Y10\ \text{--}\ (10F8)\ \text{intensities}$
 $S1\ S2\ \dots\ S10\ \text{--}\ \text{"}\ \text{standard deviations}$
 -4: Format given by DBWS program for synchrotron data. (Version DBW3.2S-8711)
- =5 Data from GENERAL FORMAT for TWO AXIS instrument. Three lines of text followed by two lines with the items:
 $NPTS, TSample, Tregul, Ivvari, Rmon1, Rmon2$
 $T_i, step, T_f$
 Set of lines containing 10 items corresponding to the Intensities in format 10F8.1, up to $NPTS$ points ($NPTS=(T_f-T_i)/step+1$), followed by the corresponding standard deviations in format (10f8.2) if $Ivvari=1$. If $Ivvari=0$ the standard deviations are calculated as
- $$\sigma(y) = \sqrt{y \times \frac{Rmon1}{Rmon2}}$$
- =6 D1A/D2B standard format prepared by D1A(D2B)SUM (ILL), ADDET(LLB), MPDSUM (LLB) or equivalent programs.
- =7 Files from D4 or D20L
- =8 Data from DMC at Paul Scherrer Institute.
- =10 X, Y, Sigma format with header lines. In all cases the first six lines are considered as comments. If in the first line (left adjusted) appears the keyword XYDATA, then the following five lines are considered as the heading of the file. Among these five lines the following keywords and values have a meaning to the program:
INTER $fac_x\ fac_y\ Interpol\ Stepin$
TEMP $tsamp$
 fac_x internal multiplier of X-values
 fac_y internal multiplier of Y and Sigma-values
 $Interpol$
 =0 Variable step is used in the program
 =1 The variable step data are interpolated internally to the constant step $Stepin$.
 =2 Data are supplied directly at constant step
- If no sigma values are provided the program assumes that $\sigma(y) = \sqrt{y}$. You can add comments to the data file if they start with the character “!” in the first position of the line. These lines are ignored by the program.
- =11 Data from variable time X-ray data collection. The first four lines are considered as comments. The following lines are:
 $2Theta_i, step, 2Theta_f\ Comment$
 (Time, Intensity) in format 5(F6, I10). The program uses the information contained in Time to normalise the observed intensities to the average time $\langle Time \rangle$ and to calculate the variance of the normalised values.
- =12 The input data file conforms to GSAS standard data file. BINTYP = LOG6, TIME_MAP and LPSD are not yet available.

Hkl – Output of reflection list in CODEFIL.hkl

Prepares CODFIL.hkl. See the section [Output files](#) for details.

- =0 No action
- =1 Outputs: Code, h, k, l , $mult, d_{hkl}, 2\theta$, FWHM, I_{obs} , I_{calc} , $I_{obs} - I_{calc}$
 or if ABS(**Job**)>1 : h, k, l , $mult, I_{calc}, 2\theta, d_{hkl}$
- =2 Output for SIRPOW.92: h, k, l , $mult, \sin \theta / \lambda, 2\theta, FWHM, F^2, \sigma(F^2)$
- =±3 Output of Real and Imaginary parts of Structure Factors: h, k, l ,
 $mult, F_{real}, F_{imag}, 2\theta, Intensity$
- =4 Output of: $h, k, l, F^2, \sigma(F^2)$.

=5 Output of: $h, k, l, mult, F_{calc}, T_{hkl}, d_{hkl}, Q_{hkl}$

Fou- Output of CODFIL.fou files

Prepares CODFIL.fou. See the section [Output files](#) for details.

- =0 No action
- =1 Cambridge format
- =2 SHELXS format are also in (Prepares also the file CODFILn.ins)
- =3 FOURIER format (Prepares also the file CODFILn.inp)
- =4 GFOURIER format (Prepares also the file CODFILn.inp)

Ana –Reliability of the refinement analysis

- =0 No action
- =1 Provides an analysis of the refinement at the end of the summary file CODFIL.sum.

Powder data experimental set-up I-fixed parameters

(LINE8 is optional, Read only if powder patterns are used, hence, if [Cry=0-LINE 4](#))

The input format depend on [Uni](#)

Multi-Pattern:

A specific line (8n) should be given for each pattern

[For 2 θ]

Comment line:! λ_1 λ_2 Ratio Bkpos Wdt Cthm muR AsyLim Rpolarz -> Patt #N

LINE 8n: [Lambda1](#), [Lambda2](#), [Ratio](#), [Bkpos](#), [Wdt](#), [Cthm](#), [muR](#), [AsymLim](#), [Rpolarz](#)

(9 reals)

[For T.O.F or Energy dispersive data]

Comment line:![Bkpos](#) [Wdt](#) [Iabscor](#) for Pattern# N

LINE 8n: [Bkpos](#), [Wdt](#), [Iabscor](#) (2 reals, 1 integer)

LINE 8n: [Bkpos](#), [Wdt](#), [Iabscor](#) (2 reals, 1 integer)

[(Old) Single pattern format]

This time, the single pattern format has the same format as the multi pattern

Only one line of the above format has to be given

Common variables for each 2theta, TOF and energy dispersive pattern

Bkpos – Origin of the polynomial-bckg

Origin of polynomial for background (in 2 θ degrees or μ seconds for TOF)

Wdt – Cut-off of the peak profile tails

Width (range) of calculated profile of a single Bragg reflection in units of FWHM (typically 4 for Gaussian and 20-30 for Lorentzian, 4-5 for TOF). The value of the peak shape function is set to zero for $ABS(x) > \text{Wdt} \times \text{FWHM}$, with $x = T_i - T_h$.

Variables specific to 2theta pattern

Lambda1

Wavelength λ_1

Lambda2

Wavelength λ_2 ($=\lambda_1$ for monochromatic beam)

Ratio – of the two wavelength weight

Intensity Ratio I_2 / I_1

If Ratio < 0 the parameters U,V,W (see below) for the second wavelength are read separately.

Cthm – Monochromator polarization correction

Coefficient for monochromator polarisation correction. See Mathematical section.

muR – Absorption correction

Absorption correction coefficient μR , used only for refinement on cylindrical samples and flat samples with symmetrical θ - 2θ scanning (the scattering vector lying within the sample plane).

μ = effective absorption coefficient

R = radius or thickness of the sample

AsymLim – Limit angle for asymmetry correction

Peaks below this 2θ limit are corrected for asymmetry.

Rpolarz – Polarization factor

Polarisation factor (synchrotron, Ilo=3)

Fraction of mosaic-crystal (transmission geometry, Ilo=2)

Variables specific to TOF or energy dispersive pattern

Iabscor - absorption correction for T.O.F

Type of absorption correction for T.O.F. data

- =1 Flat plate perpendicular to the incident beam
- =2 Cylindrical sample
- =3 Exponential correction $Abs = \exp[-c\lambda^2]$

Refinement parameters and powder data range

(LINE 9 is compulsory)

Input format depend on Uni

Single and Multi-Pattern:

A specific line (9n) should be given for each pattern

Comment line: !NCY Eps R_at R_an R_pr R_gl
LINE 9: NCY, Eps, R_at, R_an, R_pr, R_gl (1 integer - 5 reals)
Comment line: ! Thmin Step Thmax PSD Sent0 -> Patt #N
LINE 9n: [2 θ]: Thmin, Step, Thmax, PSD, Sent0 (5 reals)
Comment line: ! TOF-min <Step> TOF-max PSD Sent0 -> Patt# N
LINE 9n: [T.O.F]: Thmin, Step, Thmax (3 reals)
Comment line: ! Emin <Step> Emax -> Patt# N
LINE 9n: [Energy dispersive]: Thmin, Step, Thmax (3 reals)

[(Old) Single pattern format]

The items of the different multi-pattern lines 9 and 9n are merged in a single line for the single patten format as:

Comment line: ! NCY Eps R_at R_an R_pr R_gl Thmin Step Thmax PSD Sent0
Comment line: ! NCY Eps R_at R_an R_pr R_gl TOF-min <Step> TOF-max
Comment line: ! NCY Eps R_at R_an R_pr R_gl Emin <Step> Emax
LINE 9: NCY, Eps, R_at, R_an, R_pr, R_gl , Thmin, Step, Thmax, PSD, Sent0 (1 integer, 10 reals)

Common variables for all patterns

NCY – Number of refinement cycles

Number of cycles of refinement

Eps – Control of the convergence precision

Forced termination when *shifts* < Eps × e.s.d. A reasonable value is Eps=0.2 or lower.

R_at, R_an, R_pr, R_gl- 4 relaxation factors of the shifts of the refined parameters

The four relaxation factors for shifts:

1. Atomic parameters: co-ordinates, magnetic moments, site occupancies and isotropic displacement (temperature) factors
2. Anisotropic displacement (temperature) factors
3. Profile parameters, asymmetry, overall displacement (temperature), cell constants, preferred orientation parameters, strains, size, propagation vectors & user-supplied parameters.
4. Global parameters, zero-shift, background, displacement and transparency.

Variables dependent on the pattern number

Thmin – Starting scattering variable value of a pattern

Starting angle 2θ /TOF/Energy for calculated pattern in degrees/micro-seconds/keV. For normal refinement the triplet Thmin, Step, Thmax is superseded by reading the provided file with profile intensities.

Step - Step of the scattering variable in a pattern

Step size in degrees 2θ /micro-seconds/keV

Thmax - scattering variable

Ending angle 2θ /TOF/Energy for calculated pattern in degrees/micro-seconds/keV

PSD - Incident beam angle

Incident beam angle at sample surface in degrees

Sent0 – Max angle to where the primary beam contribute

Theta angle at which the sample intercepts completely the x-ray beam. Below Sent0 part of the beam doesn't touch the sample and the intensity of reflections below Sent0 have to be multiplied by the factor:

$$sclow = \sin \theta / \sin(SENT0)$$

Interpolated background - Pattern's regions to exclude

(LINE 10 is **optional**, read only if $Cry=0$ -Line 4, and if at least one of

$Nba(n_pat) \geq 2$ –linear interpolation

or $Nba(n_pat) \leq -4$ – cubic splines interpolation)

(LINE 11 is **optional**, read only if $Cry=0$ -Line 4, and if at least one of $Nex(n_pat) > 0$)

Input format

For each pattern, in the order they are given in LINE 5, two set of lines should be given sequentially, depending on $Nba(n_pat)$ and $Nex(n_pat)$.

The first one is a list of $Nba(n_pat)$ bckdg points if $Nba(n_pat) \geq 2$ or $Nba(n_pat) \leq -4$.

The second one is a list of $Nex(n_pat)$ lines of excluded regions

This makes a maximum total of $\sum_{n_pat=1}^{NPATT} |NBCKGD(n_pat)| + NEXCRG(n_pat)$ lines

Note: Special values of $Nba(n_pat)$ ($=-3,-2,-1,1$) are reserved. In this case, some background coefficients are read elsewhere, in the [line 17](#)

Comment line: !2Theta/TOF/E(Kev) Background for Pattern #N

LINE 10n: POS, BCK ($Nba(n_pat)$ lines table of 2 reals)

Comment line: !Excluded regions (LowT HighT) for Pattern #N

LINE 11n: ALOW, AHIGH ($Nex(n_pat)$ lines table of 2 reals)

Variables

Units of **POS**, **ALOW**, **AHIGH** depend on the scattering variable unit, defined as [Uni\(n_pat\)](#):

POS

Position in degrees/micro-seconds/keV

BCK

Background counts at this position

If $Nba(n_pat) > 0$ linear interpolation

If $Nba(n_pat) < 0$ cubic splines interpolation

ALOW

Low scattering variable bound in degrees/micro-seconds/keV

AHIGH

High scattering variable bound in degrees/micro-seconds/keV

User defined scattering lengths and form factors

(LINE 12 is optional, it is to be given only if at least one of $\text{Nsc}(n_pat) \neq 0$ -Line 4)

Input format

For each pattern, in the order they are given in LINE 5, a list of $\text{Nsc}(n_pat)$ (if $\neq 0$) user provided scattering factors or form factors can be given.

Comment line: ! Additional scattering factor for Pattern #N

Line12: **NAM, DFP, DFPP, ITY** (Character A4 - 2 reals - 1 integer) ($\text{Nsc}(n_pat)$ Lines)

(Optional LINE 12b read for user-defined form factors)

LINE 12b: A1, B1, A2, B2, A3, B3, A4, B4,C

(9 reals, for X-Rays form factors, **Job**=0 or 2, **ITY**=0 and $\text{Nsc}(n_pat) > 0$)

LINE 12b: A, a, B, b, C, c, D (7 reals)

(7 reals, for Neutrons magnetic form factors, if **Job**=1 or 3, **ITY**=1 and $\text{Nsc}(n_pat) > 0$)

LINE 12b: A, a (2 reals)

(2 reals form factors table, given for X-Rays if **ITY**=0, or for Neutron if **ITY**=1, if

$\text{Nsc}(n_pat) < 0$)

Variables

NAM – Name of the chemical element

Symbol identifying this set (left justified). This symbol is converted to lower case for X-ray diffraction global data.

DFP – Df' (X-rays) or b (neutron)

Df' or Neutron Scattering length b

DFPP – Imaginary part Df'' (X-rays)

Df'' (ignored in the neutron case)

ITY – Options for form factor definition

There are many options, depending on the radiation, the value of **ITY**, and the sign of $\text{Nsc}(n_pat)$:

- For Xrays, (**Job**=0 or 2):

The scattering lengths for X-rays are user defined, and Df' and Df'' are read in **DFP** and **DFPP**. The form factor will depend on the following options:

ITY =0

- If $\text{Nsc}(n_pat) > 0$:

The option allows the user to provide the $\sin\theta/\lambda$ dependent part of the X-rays form factor f : in this case, a new line is read with the format:

LINE 12b: A1, B1, A2, B2, A3, B3, A4, B4,C (9 reals)

A1,B1,A2,B2,A3,B3,A4,B4,C give the coefficients for the analytic approximation to the X-ray form factor f . The expression is the following:

$$f\left(\frac{\sin\theta}{\lambda}\right) = \sum_{i=1}^4 a_i \exp\left\{-b_i\left(\frac{\sin\theta}{\lambda}\right)^2\right\} + c$$

If A2 is set to zero, only A1 are B1 used

- **If $\text{Nsc}(n_pat) < 0$:**
Allows the user to define his own form factors as a table of $\sin\theta/\lambda$ - f values
For each scatterer, the user has to provide a table made of a set of lines in the form: $\sin\theta/\lambda$ - f :
LINE 12b: A, a (2 reals)
to define the form factors. The set have to be terminated by a line with the number “-100” in first position.

ITY =2

Indicates that you are giving just Df' and Df'' and the program will use tabulated coefficients for the $\sin\theta/\lambda$ dependent part of f (X-rays). The name **NAM** must correspond in this case to a valid tabulated name (See Notes(1, 2) below). At variance with the name used for determining the scattering factor in the description of atoms, the chemical symbol used in **NAM** must be LOWER CASE. This is the most simple way of giving anomalous dispersion parameters for synchrotron data.

- For neutrons, (Job=1 or 3):

ITY =0 Only reads a user defined atomic Fermi length b for the species **NAM** in **DEF**.

ITY =1 Allows the user-definition of the magnetic form factors:

- **If $\text{Nsc}(n_pat) > 0$:**
The option allows you to give a magnetic form factor. In this case, *a new line is read with the format:*
LINE 12b: A, a, B, b, C, c, D (7 reals)
Where A,a,B,b,C,c,D give the coefficients for the analytic approximation to the magnetic form factor f (P.J. Brown, Vol C new ed. ITC) similar to the X-ray form factor $f\left(\frac{\sin\theta}{\lambda}\right)$ given above for Xrays, but with the sum extended to 3 terms only.
If B is set to zero, only the parameters A and a are used.
- **If $\text{Nsc}(n_pat) < 0$:**
Allows the user to define his own form factors as a table of $\sin\theta/\lambda$ - f values
For each scatterer, the user has to provide a table made of a set of lines in the form: $\sin\theta/\lambda$ - f :
LINE 12b: A, a (2 reals)
to define the form factors. The set have to be terminated by a line with the number “-100” in first position.

Note: Scattering length, X-rays and magnetic form factors are stored in internal tables. To use them you must give the “name” of the scatterer using UPPER CASE chemical symbols (for neutron scattering length b), chemical species (e.g. CU+2, for the X-rays form factor of Cu^{2+}) or M followed by the chemical symbol and formal charge state (for magnetic form factor, e.g. MNI2 for the magnetic Ni^{2+}). These names will have to be given in **Typ**, in the lines **25a-1** or **25b-1** behind the atom name (see below).

For X-ray diffraction, in the case of giving user supplied Df' and Df'' , the chemical symbol is converted to LOWER CASE: the form factors symbols **Typ** behind the atom name can then be given either in LOWER or in UPPER case. This will force the program to use the user-defined values.

If tabulated values of the magnetic form factors of the rare earths are to be used, two options exist:

Example:

MHO3: magnetic form factor of Ho^{3+} as $\langle j_0 \rangle$

JHO3: magnetic form factor of Ho^{3+} as $\langle j_0 \rangle + c_2 \langle j_2 \rangle$

where c_2 has been calculated using the dipolar approximation. Seven coefficients A, a, B, b, C, c, D are used for approximating $\langle j_0 \rangle + c_2 \langle j_2 \rangle$.

Number of refined parameters

(LINE 13 is compulsory)

<i>Input format</i>

LINE 13: Maxs (one integer) !Number of refined parameters (Comment on the same line)

<i>Variable</i>

Maxs - Number of refined parameters

Number of parameters varied. In case of using Aut = 1 the program determines automatically the number of parameters to be refined. In this case, Maxs is updated to the total number of parameters that are automatically varied.

Powder data experimental set-up II (refinable parameters)

(LINE 14-17, are **optional** lines, looped over the patterns; they are read only if powder patterns are used, hence, if Cry=0-Line 4). They include:

Some general parameters:

(LINE 14 is compulsory, its format depend on Uni-Line 4)

Micro absorption:

(LINE 15, is an **optional** line, read only if Ilo(n_pat)=0 and Job(n_pat)=0 or 2, and Uni=0 on LINE 4)
The meaning of the parameters of this line is detailed separately in the LINE 15

Jason Hodges formulation of T.O.F vs. D spacing

(LINE 14-17, are **optional** and read only if powder patterns are used, hence, if Cry=0-Line 4)
The meaning of the parameters of this line is detailed separately in the LINE 16

Analytic background models:

(LINE 17, is **optional** and read only for special values of Nba(n_pat) in LINE 4)
The meaning of the parameters of this line is detailed separately in the LINE 17

Input format depend on Uni

Single and Multi-Pattern:

A specific set of lines (14-17) should be given for each pattern. For each value of Uni(n_patt), there is a compulsory Line 14 to give, and optional lines 15-17

For 2θ patterns:

Comment line:! Zero Code Sycos Code Sysin Code Lambda Code More -> Patt #N
LINE 14: [2θ] Zero, Code, Sycos, Code, Sysin, Code, Lambda, Code, More (8 reals - 1 integer)

(Optional LINE 15 read if More≠0)

Comment line:! P0 Cod_P0 Cp Cod_Cp Tau Cod_Tau
LINE 15 : Micro-absorption parameters format: see below

(Optional LINE 17 read if $-2 \leq \text{Nba}(n_pat) \leq 1$)

Comment line:! Background coefficients/Codes for Pattern #N or,
Comment line:! Window for Fourier Filter for Pattern #N, or
Comment line:! Additional Background coefficients/Codes for Pattern #N
LINE 17 : Background parameters format: see below

For T.O.F patterns :

Comment line:! Zero Code Dtt1 Code Dtt2 Code 2SinTh -> Patt #N
LINE 14: [T.O.F] Zero, Code, Dtt1, Code, Dtt2, Code, 2SinTh (7 reals)

(Optional LINE 16 read if Npr(n_patt)=10)

Comment line:! Zerot/Code Dtt1t/Code Dtt2t/Code x-cross/Code Width/Code
LINE 16 : Jason Hodges formulation: see below

(Optional LINE 17 read if $-2 \leq \text{Nba}(n_pat) \leq 1$)

Comment line:! Background coefficients/Codes for Pattern #N or,
Comment line:! Window for Fourier Filter for Pattern #N, or
Comment line:! Additional Background coefficients/Codes for Pattern #N
LINE 17 : Background parameters format: see below

For Energy dispersive patterns:

Comment line:! Zero Code StE1 Code StE2 Code 2SinTh -> Patt #N
LINE 14: [Energy dispersive] Zero, Code, STE1, FSTE1, STE2, FSTE2, 2SinTh (7 reals)

(Optional LINE 17 read if $-2 \leq \text{Nba}(n_pat) \leq 1$)

Comment line:! Background coefficients/Codes for Pattern #N or,
Comment line:! Window for Fourier Filter for Pattern #N, or
Comment line:! Additional Background coefficients/Codes for Pattern #N
LINE 17 : Background parameters format: see below

Variables

Zero – Zero point

Zero point for T (in degrees/microseconds/keV): $T_{True} = T_{Exp} - ZER$

Code - Codeword

Codeword for zero-shift

Sycos – Systematic shift (cos dependance)

Systematic 2θ shift with $\cos\theta$ dependence. Sample displacement in $\theta - 2\theta$ diffractometers

Code - Codeword

Codeword for [Sycos](#)

Sysin - Systematic shift (sin dependance)

Systematic 2θ shift with $\sin 2\theta$ dependence. Sample transparency coefficient in θ - 2θ diffractometers

Code- Codeword

Codeword for [Sysin](#)

Lambda – Wavelength (refinable)

Wavelength to be refined (only a single wavelength can be refined)

Code- Codeword

Codeword for [Lambda](#).

Cell parameters should be fixed if wavelength is to be refined.

Dtt1, Dtt2 – Reflexion positions parameters in T.O.F patterns

The TOF position of a reflection, for [Npr](#)(*n_pat*)= 9, with d-spacing d is calculated using the formula:

$$\text{TOF} = \text{Zero} + \text{Dtt1} d + \text{Dtt2} d^2$$

The component of the TOF position of a reflection, for [Npr](#)(*n_pat*)= 10, with d-spacing d for the region of epithermal neutrons ([Dtt2](#) is not used) is calculated using the formula:

$$\text{TOF}_e = \text{Zero} + \text{Dtt1} d$$

See next [line](#) to see the calculation of the TOF position when [Npr](#)(*n_pat*)= 10.

Code, Code – Refinement codewords

Codewords for [Dtt1](#), [Dtt2](#)

2SinTh – Angle of the detector bank in T.O.F patterns

Value of $2\sin\theta$ for the detector bank. Used for obtaining the wavelengths and for Lorentz factor correction.

More – Flag to read micro-absorption coefficients

If different from zero, and the scattering variable is $2j$ (see [Uni](#)), the following [line](#) [LINE 15](#) (OPTIONAL) is read to define the microabsorption coefficient

Micro-absorption coefficients

(LINE 15 is **OPTIONAL**, and read for those 2 θ patterns ($Uni(n_pat)=0$) for which $Ilo(n_pat)=0$ and $Job(n_pat)=0$ or 2 on [LINE 4](#), and $More\neq 0$ on [LINE 14](#))

Input format

Single and Multi-Pattern:

The right place to input this line is detailed in the [LINE 14](#)

Line 15: P0, CP0, CP, CCP, TAU, CTAU (5 reals, to be given)

Comment line: ! P0 Cod_P0 Cp Cod_Cp Tau Cod_Tau

Variables

Micro-absorption coefficients and codes. Only valid for Bragg-Brentano geometry.
The value of the parameters correspond to the variables in the following formula:

$$P(\mathbf{h}) = P_0 + C_p \frac{\tau}{\sin \theta_h} \left(1 - \frac{\tau}{\sin \theta_h} \right)$$

See Mathematical section for details.

P0, value of P_0 ; **CP0**, codeword of P_0

CP, value of C_p ; **CCP**, codeword of C_p

TAU, value of τ ; **CTAU**, codeword of τ

Jason Hodges formulation of the TOF versus d-spacing

(LINE 16 is **OPTIONAL**, and read for those T.O.F patterns ($Uni(n_pat)=1$) for which $Npr(n_pat)=10$)

Input format

Single and Multi-Pattern:

The right place to input this line is detailed in the [LINE 14](#)

Comment line: ! Zerot/Code Dtt1t/Code Dtt2t/Code x-cross/Code Width/Code

Line 16 [Num_TOF_Patt_Lines]: Zerot, Dtt1t, Dtt2t, x-cross, Width (5 reals)

LINE 16-1: [Num_TOF_Patt_Lines] CZERT, CDTT1t, CDTT2t, Cx-cross, CWIDTH (Codewords 5 reals)

Variables

This line corresponds to the formulation, by Jason Hodges, of the TOF versus d-spacing that divides the spectrum in three parts depending on the neutron energy. The epithermal (fastest neutrons), crossover (intermediate) and thermal regions. It is supposed that in the previous line the parameters allowing the calculation of TOF as a function of d-spacing correspond to epithermal neutrons. In this line the parameters controlling the thermal and crossover regions. See Mathematical section for details.

Zerot

Zero shift for thermal neutrons

Dtt1t

Coefficient 1 for d-spacing calculation

Dtt2t

Coefficient 2 for d-spacing calculation

x-cross

Position of the centre of the crossover region

Width

Width of the crossover region

The component of the TOF position of a reflection, for Npr(*n_pat*)= 10, with d-spacing *d* for the region of thermal neutrons is calculated using the formula:

$$TOF_t = \text{Zerot} + \text{Dtt1t} d - \text{Dtt2t} d^1$$

The position of a Bragg reflection (in microseconds TOF) is finally calculated using the expressions:

$$n_{\text{cross}} = 0.5 \operatorname{erfc}(\text{Width} (\text{x-cross}-d^1))$$

$$TOF = n_{\text{cross}} TOF_e + (1-n_{\text{cross}}) TOF_t$$

Where *erfc* is the complementary error function.

CZERt, CDTT1t, CDTT2t, Cx-cross, CWIDTH

are the codewords of the previous parameters.

Annalytic background model

(LINE 17, is read if for some patterns, Nba(*n_pat*) takes some special values)

Input format will depend on Nba(*n_pat*)

*For each pattern for which $-2 \leq \text{Nba}(n_pat) \leq 1$, a list of lines defining the background coefficients has to be given at . The number of lines depends Nba(*n_pat*).*

The right place to input this line is detailed in the LINE 14

The coefficients correspond to the following models:

*12 coefficients-2θ Polynome function for Nba (*n_pat*)=-3*

Line 17 [4 lines for each corresponding pattern]:

<u>BACK1, BACK2, BACK3, BACK4, BACK5, BACK6</u>	(6 reals)
<u>FBACK1.FBACK2.FBACK3.FBACK4.FBACK5.FBACK6</u>	(Codewords 6 reals)
<u>BACK7, BACK8, BACK9, BACK10, BACK11, BACK12</u>	(6 reals)
<u>FBACK7.FBACK8.FBACK9.FBACK10.FBACK11.FBACK12</u>	(Codewords 6 reals)
Comment line:! Background coefficients/Codes for Pattern #N	

*Fourrier filtering wondow, for Nba (*n_pat*)=-2*

Line 17 [1 line for each corresponding pattern]:

FWINDOW (1 integer)

Comment line:!
Window for Fourier Filter for Pattern #N

2θ-6 coefficients polynome function + Debye-like function for Nba (n_pat)=-1

Line 17: [6 lines for each corresponding pattern]

BACK1, BACK2, BACK3, BACK4, BACK5, BACK6 (6 reals)
FBACK1,FBACK2,FBACK3,FBACK4,FBACK5,FBACK6 (Codewords 6 reals)
Bc1, Bc2, Bc3, Bc4, Bc5, Bc6 (6 reals)
CBc1, CBc2, CBc3, CBc4, CBc5, CBc6 (Codewords 6 reals)
D1, D2, D3, D4, D5, D6 (6 reals)
CD1, CD2, CD3, CD4, CD5, CD6 (Codewords 6 reals)
Comment line:! Additional Background coefficients/Codes for Pattern #N

6 coefficients 2θ Polynome for Nba (n_pat)=0

Line 17: [2 lines for each corresponding pattern]

BACK1, BACK2, BACK3, BACK4, BACK5, BACK6 (6 reals)
FBACK1,FBACK2,FBACK3,FBACK4,FBACK5,FBACK6 (Codewords 6 reals)
Comment line:! Background coefficients/Codes for Pattern #N

4 additional coefficient-2θ polynome for Nba (n_pat)=1

Line 17: [2 lines for each corresponding pattern]

BACK1, BACK2, BACK3, BACK4 (6 reals)
FBACK1,FBACK2,FBACK3,FBACK4 (Codewords 6 reals)
Comment line:! Background Trans_coefficients/Codes for Pattern #N

Variables

BACK(i, n_pat) – Background polynomial coefficients

Background coefficients (see [Mathematical section](#))

Bc(i, n_pat), D(I, n_pat) – Background Debye-like function coefficients

Background coefficients (see [Mathematical section](#))

FBACK(i, n_pat) – Codewords

Codewords for background coefficients

CBc(i, n_pat), CD(i, n_pat) – Codewords

Codewords for background coefficients

FWINDOW- Number of points of the window of fourier filtering

Window for Fourier filtering. The value of **FWINDOW** must be much greater than the number of points subtended by the base of a single Bragg reflections in the widest region (a factor greater than five, at least!).

The starting background is read from file FILE.bac as in the case **Nba**=1. But, at variance with the case **Nba** =1, the file FILE.bac is re-written at the end of the session.

Note: If **Nba**=1 (background read from file), **BACK1** cannot be zero. Only four coefficients are needed if such a case.

LOOP OVER PHASES

At this point of the CODEFIL.pcr file, **Nph** (defined on [line 4](#)) blocks of lines (Line 18 to 46) should be given in order to describe the different phases to be included. These phases are used model the single/multi-pattern/integrated intensities if the users give input data (for **Job**=0,1,-1 on [line 4](#)), or to calculate simulated powder patterns (for **Job**=2,3,-3 on [line 4](#)). Each block defines several control codes, parameters and variables that are associated to the phase. The beginning of each block is identified with a heading made of several comment lines (optional on the reading, but re-written automatically after the first refinement). These comments have the form:

Comment line:

```
-----  
! Data for PHASE number:n_phas ==> Current R_Bragg for Pattern# n_pat1: 6.79  
! Data for PHASE number:n_phas ==> Current R_Bragg for Pattern# n_pat2: 5.46  
...  
!-----
```

indicating the R_Bragg of the different patterns to which the phase contributes. The selection of such patterns is made through **JCONTR(n_pat(i))**- [Line 19](#)

The specific job and parameters to be applied to each phase and the specific outputs are selected in the [line 19](#). The other parameters and variables for this phase are given in the following lines, and their format will depend on the control codes given in [Line 19](#). These parameters include:

- The phase name -[Line 18](#),
- The phase crystallographic/magnetic/mechanic symmetry properties, [Line 22](#) and [Line 23a](#) or [Line 23b](#).
- The atomic and/or magnetic parameters in several available formats on in the [Line 25a](#) or the [Line 25b](#)

The input format then differs slightly, whether the phase is intended to contribute to patterns (controlled through **Cry** in [Line 4](#) and **JCONTR** in [Line 19](#)) or if it contributes to a set of integrated intensities as observations (in this case, **Cry**≠0, **Irf** on [Line 19](#) should be set to 4, and all **JCONTR** set to 0 on the [Line 19](#)):

- For those phases participating to a (some) patterns(s), (then **JCONTR(n_pat)** should be ≠0 to indicate the phase contribute to the pattern n_pat), the user have to give, for each pattern, a set of lines in the following order:
Profile I parameters (including some general parameters and strain parameters in the [Line 26-27](#)) / Cell parameters ([Line 29](#)) / profile II (preferred orientation and asymmetry parameters- [Line 30](#)) /hkl shifts ([Line 34-35](#))/ size parameters ([Line 36](#))/ additional size-strain parameters ([Line 37-38](#))
Lines 34 to 38 are optional and their reading is controlled by the codes **Sol** in the line 19 for hkl-shifts, **Str** on line 19 for strain together with **Strain-Model** on Line 26, and **Size-Model** in the Line 27 for size.
- In a case of a phase contributing to a single crystal integrated intensity reflection list, these lines have a different format and correspond to
Scale parameters ([Line 26](#))/Extinction parameters ([Line 29](#))/λ/2 contamination ([Line 30](#))
- After, the reading of the CODEFIL.pcr file is common for both modes, and continue at the line [Line 43](#) up to the end.

Phase name

(LINE 18 is a compulsory line, for information purpose only)

Input format

Line 18: PHSNM (Character)

Variable

PHSNM

Name of phase

Main control codes line for the phase

(LINE 19 is a compulsory line)

Input format

Multi-Pattern:

The selection of the patterns to which the phase contributes is made in line 19-2 and 19-3. More≠0 in the line 19-1 is used to read optional codes and variables to define specific outputs and/or jobs related to the phase

Comment line: ! Nat Dis {Mom or Ang} Pr1 Pr2 Pr3 Jbt Irf Isy Str Furth ATZ Nvk Npr More

LINE 19: Nat, Dis, {Mom(Moment) or Mom(Angles)}, Jbt, Isy, Str, Furth, ATZ, Nvk, More

(7 integers - 1 real - 2 integers)

(Optional line to be given if More≠0)

Comment line: !Jvi Jdi Hel Sol Mom Ter

LINE 19-1: Jvi, Jdi, Hel, Sol, Mom, Ter

(6 integers)

(Read one compulsory line)

Comment line: ! Contributions (0/1) of this phase to the #N Pattern

LINE 19-2: JCONTR(1:NPATT) (NPATT integers)

(For each pattern, two optional lines 19-3n and 19-4n are read if JCONTR(n_pat)≠0)

Comment line: ! Irf Npr Jtyp for Pattern #N

LINE 19-3: Irf, Npr, Jtyp (3 reals)

Comment line: ! Pr1 Pr2 Pr3 Brind Rmua Rmub Rmuc for Pattern #N

LINE 19-4: Pr1 Pr2 Pr3, Brind, Rmua, Rmub, Rmuc (7 reals)

[(Old) Single pattern format]

The items of the different multi-pattern lines are merged in two lines with a slightly different formatting in case of the use of the single pattern format (The multi-pattern Lines 19, 19-3n excepted [Jtyp](#) and [Pr1 Pr2 Pr3](#) in 19-4n are merged into Line 19; for [More≠0](#), the Line 19-1, [Jtyp](#) in the line 19-3n and items of the Line 19-4n-excepted [Pr1 Pr2 Pr3](#) are merged into line 19-1). As in the multi pattern format [More≠0](#) is used to read optional outputs and/or jobs defined in the line 19-1:

Comment line: ! Nat Dis {Mom or Ang} Pr1 Pr2 Pr3 Jbt Irf Isy Str Furth ATZ Nvk Npr More
LINE 19: [Nat](#), [Dis](#), {[Mom](#)(Moment) or [Mom](#)(Angles)}, [Pr1](#) [Pr2](#) [Pr3](#), [Jbt](#), [Irf](#), [Isy](#), [Str](#), [Furth](#), [ATZ](#), [Nvk](#),
[Npr](#), [More](#)
(3 integer, 3 reals, 5 integers, 1 real, 3 integers)

(Optional line to be given if [More≠0](#))

Comment line: ! Jvi Jdi Hel Sol Mom Ter Brind RMua RMub RMuc Jtyp
LINE 19-1: [Jvi](#), [Jdi](#), [Hel](#), [Sol](#), [Mom](#), [Ter](#), [Brind](#), [Rmua](#), [Rmub](#), [Rmuc](#), [Jtyp](#)
(6 integers, 4 reals, 1 integer)

Common flags for Single/Multi-Pattern

Nat- Number of atoms

Number of atoms in asymmetric unit. The total number of atoms for all phases cannot be greater than NATS

Dis- Number of distance constraints

Number of distance constraints

Mom(Angles)- Number of angle constraints

Number of angle constraints

Mom(Moment) - Number of magnetic moment constraints

Number of magnetic moment constraints

Jbt –Structure factor model and refinement method for the phase

- = 0 The phase is treated with the Rietveld Method, then refining a given structural model.
- = 1 The phase is treated with the Rietveld Method and it is considered as pure magnetic. Only magnetic atoms are required. In order to obtain the correct values of the magnetic moments the scale factor and structural parameters must be constrained to have the same values (except a multiplying factor defined by the user) that their crystallographic counterpart. See note on magnetic refinements. The three extra parameters characterising the atomic magnetic moments corresponds to components (in Bohr magnetons) along the crystallographic axes.
- =-1 As 1 but the three extra parameters characterising the atomic magnetic moments corresponds to the value of M (in Bohr magnetons) the spherical Φ angle with X axis and the spherical Θ angle with Z axis. This mode works only if the Z axis is perpendicular to the XY plane. (for monoclinic space groups the Laue Class $1\ 1\ 2/m$ is required).
- = 2 Profile Matching mode with constant scale factor.
- =-2 As 2 but instead of intensity the modulus of the structure factor is given in the CODFILn.hkl file
- = 3 Profile Matching mode with constant relative intensities for the current phase. The scale factor can be refined. In this case [Irf](#)(*n_pat*) must be equal to 2, see below.
- =-3 As 3 but instead of intensity the modulus of the structure factor in absolute units (effective number of electrons for X-rays/ units of 10^{-12} cm for neutrons) is given in the CODFILn.hkl file. This structure factor is given for the non-centrosymmetric part of the

primitive cell, so for a centrosymmetric space group with a centred lattice the structure factor to be read is:

$$F_{reduced} = F_{conventional} / (Nlat \cdot Icen)$$

where $Nlat$ is the multiplicity of the conventional cell and $Icen=1$ for non-centrosymmetric space groups and $Icen=2$ for centrosymmetric space groups.

- = 4 The intensities of nuclear reflections are calculated from a routine handling Rigid body groups.
- = 5 The intensities of magnetic reflections are calculated from a routine handling conical magnetic structures in real space.
- =+10/-10 The phase can contain nuclear and magnetic contributions STFAC is called for reflections with no propagation vector associated and CALMAG is called for satellite reflections. CALMAG is also called for fundamental reflections if there is no propagation vector given but the number of magnetic symmetry matrices ([MagMat](#), see below) is greater than 0. The negative value indicates spherical components for magnetic parameters. For this case the atom parameters are input in a slightly different format.
- =+15/-15 The phase is treated as a commensurate modulated crystal structure. All the input propagation vectors and also $\mathbf{k}=(0,0,0)$ are identified to be magnetic and/or structural by the reading subroutine. All nuclear contributions at reflections without propagation vectors (fundamental reflections of the basic structure) and all the reflections associated to a modulation propagation vector (superstructure reflections), are calculated by MOD_STFAC. Magnetic contributions are added, if necessary, calling the subroutine CALMAG as in the case of [Jbt](#)=+10/-10. The negative value indicates spherical components for magnetic parameters. This value of [Jbt](#) implies the use of a specific format for atom parameters.

[Isy](#) – Symmetry operators reading control code

- =0 The symmetry operators are generated automatically from the space group symbol.
- =+/-1 The symmetry operators are read below. In the case of a pure magnetic phase [Isy](#) must be always equal to 1 or 2.
- =2 The basis functions of the irreducible representations of the propagation vector group are read instead of symmetry operators. At present this works only for a pure magnetic phase.

For [Jbt](#)=10 with magnetic contribution [Isy](#) could be 0 but a comment starting with “Mag” should be given after the space group symbol.

Note: For Profile Matching mode 2, [Irf](#) can be 0 in the first run. In that case, a CODFILn.hkl file is generated and [Irf](#) is set to 2 in the new CODFIL.pcr file. The file is updated at each run in the case of [Jbt](#) =2. Of course [Isy](#) must be 0.

If for a phase [Irf](#)≤0 and [Isy](#)=1, the reflections are generated from the symbol given in the place reserved for the space group. In that case, a file CODFILn.hkl is generated with the relevant (non-zero) reflections and proper multiplicities for the particular model described by user-given symmetry operators. In addition the calculated intensities are F^2 (corrected for multiplicity, scale and LP-factor) in absolute units. This file can be used as an input file to accelerate the calculations. The program does not use the intensities in new runs reading this generated file.

The contain of this generated file, apart from the features described above, is:

No k-vectors $\Rightarrow h \ k \ l \quad m \ F^2(\text{calc}) \ F^2(\text{obs})$
 k-vectors $\Rightarrow h \ k \ l \ nv \ m \ F^2(\text{calc}) \ F^2(\text{obs}) \ hr \ kr \ lr$
 with obvious meaning.

[Str](#) – Size-strain reading control code

- =0 If strain or/and size parameters are used, they are those corresponding to selected models
- =1 The generalised formulation of strains parameters will be used for this phase.
If **Strain-Model**≠0 a quartic form in reciprocal space is used (see below)
- =-1 Options **1** and **2** simultaneously. The size parameters of the quadratic form are read before the strain parameters.
- =2 The generalised formulation of size parameters will be used for this phase.
Quadratic form in reciprocal space. Only special options of strains with **Strain-Model**≠0 can be used together with this size option.
- =3 The generalised formulation of strain *and* size parameters will be used for this phase.

Furth- Number of user defined paramters

Number of further parameters defined by user, to be used with user supplied subroutines. The default is the number of parameters defining the TLS for rigid body groups. It should be used only when **Jbt**=4.

ATZ – Quantitative phase analysis

Coefficient to calculate the weight percentage of the phase.

$$ATZ = Z M_w f^2 / t$$

- Z: Number of formula units per cell, M_w = molecular weight
- f: Used to transform the site multiplicities used on line 11-41 to their true values. For a stoichiometric phase $f=1$ if these multiplicities are calculated by dividing the Wyckoff multiplicity m of the site by the general multiplicity M . Otherwise $f=Occ.M/m$, where Occ . is the occupation number given in **LINE 25**.
- t: Is the Brindley coefficient that accounts for microabsorption effects. It is required for quantitative phase analysis only. When different phases have similar absorption (in most neutron uses), this factor is nearly 1 (in such case $ATZ=Z.M_w.f^2$). The Brindley coefficient is directly read in one of the following lines.

Nvk - Number of propagation vectors

Number of propagation vectors. If **Nvk** < 0 the vector **-k** is added to the list.

More - Options

If different from 0 the **LINE 19-1** is read

Optionnal parameters common for Single/Multi-Pattern (More≠0)

Jvi – Optional outputs

- =1 A file suitable for SCHAKAL is generated
- =2 A file suitable for STRUPLO is generated (The extension of the file is in both cases “.sch”)
- =11 If **Jbt**=2 a file CODFIL*n*.int with a list of overlapped peak clusters is output. Useful as input file for working with integrated intensities in further processing using **Irf**=4 and/or **Cry**=1, 2, 3 (Least Squares, Montecarlo or Simulated Annealing optimisation).

Jdi – Optional crystallographic output

- =1 Creates a file called CODFIL*n*.atm with all atoms within a primitive unit cell for a magnetic phase. The number n corresponds to the number of the current phase. If **Jbt**=10 only the list of magnetic atoms is generated.
- =-1 For a magnetic phase creates a file called CODFIL*n*.atm with a format suitable for further processing with the program **MOMENT**.
- =2 As 1 but, for a crystal structure, all atoms inside the conventional cell are generated.

- If **Jbt**=15 then the output is slightly different. It gives the co-ordinates of all the atoms calculated from the average structure, displacement parameters, and symmetry relations in a multiple cell defined by the user, after all the atom parameters and before scale factors. The user has to provide the rotational part transforming the average structure basis vectors into the multiple cell basis and a translation part giving the shift of the origin of the multiple cell in the conventional cell setting (see below, **LINE 25b**).
- =3 Distance and angle calculations will be performed for the current phase. Bond Valence calculations may also be performed. The output is in the file CODFIL*n*.dis. An additional file helping to create strings for soft constraints is output. This file has a fixed name: "dconstrn.hlp".
 - =4 Only Bond Valence calculations are output to the file CODFIL*n*.dis for the current phase.

Hel – Control code to constrain a magnetic structure to be helicoïdal

- =0 No action
- =1 The real and imaginary components of the Fourier coefficient of a magnetic atom are constrained to be orthogonal. The factor 1/2 is also included. This constraint may be unstable because is applied *a posteriori*. (see Mathematical section).

Sol - Additional *hkl*-dependent shifts reading code control

- =0 No action
- =1 Additional *hkl*-dependent shifts parameters are read.

Mom - Unused

Unused at present

Ter - Unused

Unused at present

Specific multi-pattern or pattern dependant parameters

JCONTR – Phase contribution to patterns flags

- =0 The current phase *does not* contributes to the pattern
- =1 The current phase contributes to the pattern

Example (it is supposed that we are dealing with 5 diffraction patterns simultaneously):

0 1 1 0 1

The meaning of the list is that the current phase only contributes to patterns number 2, 3 and 5.

Irf –Control the reflexion generation or the use of a reflexion file

- =0 The list of reflections for this phase is automatically generated from the space group symbol
- =1 The list *h, k, l, Mult* is read from file CODFIL*n*.hkl (where *n* is the ordinal number of the current phase)
- =-1 The satellite reflections are generated automatically from the given space group symbol
- =2 The list *h, k, l, Mult, Intensity* (or Structure Factor if **Jbt**=-3) is read from file CODFIL*n*.hkl.
- =3 The list *h, k, l, Mult, F_{real}, F_{imag}* is read from file CODFIL*n*.hkl. In this case, the structure factor read is added to that calculated from the supplied atoms. This is useful for simplifying the calculation of structure factors for intercalated compounds (rigid host).
- =4,-4 A list of integrated intensities is given as observations for the current phase (In the case of **Cry**≠0 this is mandatory)

The file CODFIL*n*.hkl can also be named as HKL*n*.hkl, or CODFIL.int in the case **Cry**≠0.

The format of CODFIL n .hkl files is described in this appendix in the paragraph

Npr – Specific profile function for the phase

Integer indicating the peak shape function for the present phase (see LINE 4n). If **Npr**=0, the default value of **Npr** is taken.

Jtyp – Job type for the current phase

Job type for the current phase. Allows the refinement of heterogeneous data (Same values as the global variable **Job** in LINE 4n). For the moment is only useful for **Irf**=4.

Pr1 Pr2 Pr3 - Preferred orientation direction

Preferred orientation direction in reciprocal space. Three components.

Brind - Brindley coefficient

Brindley coefficient

Rmua – Weight of integrated intensity data sets

Used when **Irf**=4. If **Rmua**=0.0 the program puts **Rmua**=1.0 internally. The value of this variable corresponds to the global weight of the integrated intensity observations with respect to the global profile. The contribution to the normal equations of the integrated intensity part is multiplied by **Rmua**.

Rmub – Exclusion of low statistic reflexions in integrated intensity data sets

If **Irf**=4, **Rmub** is a factor for excluding reflections: only the reflections verifying the constraint: $G_{\text{obs}} \geq \text{Rmub} \times \sigma(G_{\text{obs}})$, are considered in the refinement. G_{obs} is the integrated intensity, structure factor or structure factor squared of the current reflection.

If **Jvi**=11 and **Jbt**=2 and **Irf**≠4 see note below.

Rmuc – Chi2 dependent weighting of integrated intensity data sets

If **Irf**=4 and **Rmuc**>0.9 the weights are divided by the reduced χ^2 of the precedent cycle (not tested!) for integrated intensity refinements (**Irf**=4).

If **Jvi**=11 and **Jbt**=2 and **Irf**≠4 see note below.

Note: If **Jvi**=11 and **Jbt**=2 the parameters **Rmub** and **Rmuc** are used to control whether two consecutive reflections belongs to a same cluster. This is only for **Irf** different from 4/-4. The rule is the following:

The reflections i and $i+1$ belong to the same cluster if

$$T(i+1)-T(i) < 0.5 \times (H(i)+H(i+1)) \times \text{Rmub}$$

or

$$T(i+1)-T(i) < 0.5 \times (H(i)+H(i+1)) \text{ and } G(i+1) < G_{\text{sum}} \times \text{Rmuc}$$

$G(i)$ is the integrated intensity, $T(i)$ is the Bragg position, $H(i)$ is the FWHM of reflection i , G_{sum} is the cumulated integrated intensity of the current cluster.

If **Rmub** and **Rmuc** are given as zeroes, the program uses the values **Rmub**=1.0 and **Rmuc**=0.2.

From here up to the end of the PCR-file the old single pattern format coincides with the format for multiple patterns. The only prescription to take into account is that **NPATT**=1 in the loops depending on the number of patterns.

Optional Bond-Valence or distance and angles calculations

(LINE 20 and 21 are **Optional** lines, read if $J_{di}=3$ or 4)

Input format

Single/multi Pattern common format:

Only the line 20 is needed for distance and angles calculation. Optional line 21, 21-1 and 21-2 are read in case the user wants to perform BVS calculations (the character variable **BVS** have in this case to be set to "BVS"). Outputs of the results is made in a file named CODFILn.dis (where n is the ordinal number of the phase)

LINE 20: DIS_MAX, ANG_MAX, BVS
(2 reals - Character)

Comment line: ! Max_dst(dist) (angles) Bond-Valence Calc

(Optional lines, read in case of BVS calculations)

LINE 21: N_CATIONS, N_ANIONS, TOLERANCE
(2 integers - 1 real)

Comment line: !N_cations N_anions Tolerance(%) / Name cations/ and Anions

LINE 21-1: CATIONS (Character) (N_CATIONS)

LINE 21-2: ANIONS (Character) (N_ANIONS)

Variables

DIS_MAX – Control of the number of distances outputed

Maximum distance between atoms to output in file CODFILn.dis.

ANG_MAX - Control of the number of angles outputed

Maximum distance between atoms to output angles in file CODFILn.dis. If ANG_MAX=0 no angle calculations are performed.

BVS – Flag for BVS calculations

If this character variable is equal to BVS then Bond Valence calculations are performed and the results output to file CODFILn.dis. The LINE 21 is then read.

N_CATIONS - Number of cations

Number of cations

N_ANIONS- Number of anions

Number of anions

TOLERANCE - Tolerance for the ionic radius in percentage

Tolerance for the ionic radius in percentage. Two atoms are considered as bonded if their distance is less than the sum of their respective ionic radius augmented by the value of TOLERANCE. The explicit expression for considering two atoms as bonded is:

$$Distance(Atom_1, Atom_2) \leq (R(Atom_1)+R(Atom_2)) \times (1+0.01 \times \underline{TOLERANCE})$$

If TOLERANCE=0 the program takes TOLERANCE=20.

CATIONS- Symbols of the cations

Symbols of the cations in uppercase and putting the sign of the charge before the valence.

Example for three cations:

CU+2 Y+3 BA+2

The chemical species are numbered sequentially, so: Cu^{2+} is the species number 1, Y^{3+} is the species number 2 and Ba^{2+} is the species number 3. This numbering is important to identify the chemical nature of the atoms in the asymmetric unit.

ANIONS - Symbols of the anions

Symbols of the anions in uppercase and putting the sign of the charge before the valence.

Example for two anions:

O⁻² CL⁻¹

O²⁻ is the species number 4 and Cl⁻ is the species number 5.

Space group symbol

(LINE 22, compulsory line)

Input format

LINE 22: SYMB, Comment (Character A20, A60)

Variables

SYMB - Space group symbol

Space group symbol must be given in the first 20 positions of the line.

Examples:

```
P 63/m          <- Space Group Symbol
P 21 21 21      Magnetic symmetry follows
```

If SYMB starts with the keyword HALL, the program expects to read the Hall symbol of the space group instead of the conventional Hermann-Mauguin symbol.

Example:

```
HALL -P 4 2a
```

Comment – Allowed options

Only needed for Jbt=+10/-10. The Comment must start after the 20th column in the line.

Note: At present rhombohedral space groups must be given in the hexagonal description. Do not forget blanks between symmetry operators corresponding to different directions. It is advisable to check the Laue symmetry and symmetry operators in the output file especially for those space groups for which alternative origins are shown in the International Tables. By default the program uses the setting with the inversion centre at the origin. Upper and/or lower case characters can be used. If you find that for a non standard space group symbol the operators of the space group do not correspond to what you expect (the program has selected another origin!), you have to change the setting (another symbol) or give your own symmetry operators (see above Isy). For cubic space groups you may use either the old notation, e.g. $F\bar{d}3m$ or the new notation of the International Tables: $F\bar{d}3m$. The space group symbol must be given even in the case that you are giving your own symmetry operators. The reflections (if they are not read from file) will be generated according to the space group symbol.

A Comment can be put after column 20. If this Comment starts with the keyword “Mag” (without quotes) then the following line is read if Jbt=10 or Jbt=15.

Specific user defined mag symmetry with time reversal operations

(LINE 23a is optional, only read if $Jbt=\pm 10$ AND Comment = Mag)

Input format

(Specific input if $Jbt=\pm 10$ and Comment = Mag on LINE 22)

LINE 23: TIME_REV (NS+1 integers)

Comment line: ! Time Reversal Operations on Crystal Space Group

Example (NS=12 for the P 6/m m m space group):

```
P 6/m m m           Magnetic symmetry below
! Time Reversal Operations on Crystal Space Group
1  1  1  1  1  1 -1 -1 -1 -1 -1 -1  1
```

Meaning

NS is the number of independent symmetry operators given in file CODFIL.out for the crystallographic space group. TIME_REV(*i*)=-1 if time reversal is associated to operator “*i*” for magnetic symmetry, otherwise is equal to 1. The order of operators is the same as in CODFIL.out, so a first run is needed for knowing the list of crystallographic symmetry operators. For centrosymmetric groups TIME_REV(NS+1) tells the program if time reversal is associated (-1) or not (1) to the inversion operator. This last item should be given only for centrosymmetric space groups.

This approach assumes that the magnetic symmetry belongs to the family of the crystallographic space group. However the user can treat the problem using subgroups of the space group (making the appropriate constraints in the atomic positions) when needed.

Controls codes for symmetry operators AND/OR irreducible representations and basis functions

(LINE 23b, optional, read if $Isy\neq 0$)

Input format

[If $Isy\neq 0$, $Jbt\neq 15$]

Comment line: ! Nsym Cen Laue MagMat

LINE 23: Nsym, Cen, Laue, MagMat (4 integers)

[If $Isy\neq 0$, $Jbt=15$]

Comment line: ! Nsym Cen Laue DepMat MagMat

LINE 23: Nsym, Cen, Laue, DepMat, MagMat (5 integers)

[If $Isy=-2$, $Jbt=\pm 1$]

Comment line: ! Nsym Cen Laue Ireps N_Bas

LINE 23: Nsym, Cen, Laue, Ireps, N_Bas (5 integers)

(An OPTIONAL line is read if $Isy=-2$)

Comment line: ! Real(0)-Imaginary(1) indicator for Ci

LINE 23-1: ICOMPL (up to 9 integers)

Variables

Nsym - Number of crystallographic symmetry operators

Number of symmetry operators given below.

Cen – Centrosymmetry flag

= **1** Non centrosymmetric structure

= **2** Centrosymmetric structure

Laue – Laue class

Integer corresponding to the following Laue classes:

1:	-1
2:	2/m
3:	mmm
4:	4/m
5:	4/mmm
6:	-3 R
7:	-3m R
8:	-3
9:	-3m1
10:	-31m
11:	6/m
12:	6/mmm
13:	m3
14:	m3m

This number is only used for checking the symmetry operators given by users. For a phase described in a hexagonal basis one should put Laue=6,7...12, even if the space group symbol used for generating the reflections is of different symmetry.

MagMat - Number of magnetic rotation matrices

Number of magnetic rotation matrices for each symmetry operator.

DepMat - Number of atomic displacement rotation matrices

Number of atomic displacement rotation matrices for each symmetry operator. This item is given only if Jbt=15.

Ireps - Number of irreducible representations

Number of irreducible representations. The representations themselves must not be given. The user must provide the components of atomic basis functions (constant vectors) corresponding to the irreducible representations of the propagation vector group. Given only if Isy=-2.

If Ireps is given a negative value, complex basis functions will be provided, that is the real and imaginary components of the atomic basis functions.

N_Bas - Number of atomic basis functions- BSF

Number of atomic basis functions: constant vectors of three components referred to the conventional unit cell. This number corresponds to the maximum number of free coefficients that can be refined. At present N_Bas ≤ 9

(ICOMPL(i), i=1, N_Bas) – Real or pure imaginary BSF coefficients flags

Flags indicating if the coefficient of the linear combination of atomic basis functions is real or pure imaginary. See Mathematical section for details.

Input format for symmetry operators or basis functions

(LINE 24, optional, read if [Isy](#)≠0)

Input format will depend on [Isy](#)

User defined symmetry operators [[Isy](#)≠-2]

[Isy](#)≠-2

LINE 24: [Nsym](#) × (1+[MagMat](#)+[DepMat](#)) Lines

User defined basis functions [[Isy](#)=-2]

or

[Isy](#)=-2

LINE 24: [Nsym](#) × (1+ABS([Ireps](#))) or [Nsym](#) × (1+2 × ABS([Ireps](#))) Lines

The different lines have different formats:

User defined symmetry operators ([Isy](#) other than -2)

If [Isy](#)=1 the symmetry operators are given as 3x3 matrix in numeric form:

S_{ij} for symmetries relating atomic position, M_{ij} for matrix relating Fourier components of the magnetic moments, and D_{ij} for the matrix relating the Fourier components of the displacement parameters ([Jbt](#)=15). The numerical Ph value following the M_{ij} and D_{ij} symmetry operators are the magnetic/displacement phase in units of 2π .

Remark: [DepMat](#) is always 0 if [Jbt](#) ≠ 15 (see line 23b)

Comment line :

```
!S11 S12 S13 T1 S21 S22 S23 T2 S31 S32 S33 T3
!M11 M12 M13 M21 M22 M23 M31 M32 M33 Ph (If MagMat≠0)
!D11 D12 D13 D21 D22 D23 D31 D32 D33 Ph (If Jbt=±15 and DepMat≠0)
```

Followed by [Nsym](#) blocks of the form:

```
S11 S12 S13 T1 S21 S22 S23 T2 S31 S32 S33 T3 (3(3Int,1real))
D11 D12 D13 D21 D22 D23 D31 D32 D33 .Phase (9Int,1real)
R11 R12 R13 R21 R22 R23 R31 R32 R33 .Phase (9Int,1real)
DepMat lines
MagMat lines
```

- If [Isy](#)=-1 the symmetry operators are given in alpha-numeric form:

Example (the displacement operator is given only if [Jbt](#)=±15):

```
!
SYMM X, Y, Z
DSYM U, V, W, 0.0
MSYM U, V, W, 0.0
!
SYMM X+1/2, -Y, Z
DSYM u, -v, w, 0.5
MSYM -U, V, -W, 0.0
!
SYMM -x, -y, -z
```

```

DSYM   U, -V, W, 0.0
MSYM   U, V, W, 0.0
. . . . .

```

The symbols U,V,W (lower or capital case) are used for the Fourier components of the magnetic moments/displacement vectors along X,Y,Z. The numerical value following the MSYM/DSYM operator is the magnetic/displacement phase in units of 2π .

User defined basis functions ($I_{sy}=-2$)

- **If $I_{sy}=-2$ the basis functions of the irreducible representations of the propagation vector group are read in the following form:**

The keyword BASR precedes the list of 3-components vectors corresponding to the real part of the atomic basis functions. A list of $3 \times N_{Bas}$ real numbers should be given in the same line. In the case of negative I_{reps} , the keyword BASI precedes the list of $3 \times N_{Bas}$ real numbers corresponding to the imaginary components of the atomic basis functions.

Example for $N_{sym}=4$, $I_{reps}=-1$ and $N_{Bas}=3$:

```

!
SYMM   x, y, z
BASR   1 0 0  0 1 0  0 0 1
BASI   0 0 0  0 0 0  0 0 0
!
SYMM   x+1/2, -y, z
BASR   0 0 0  0 0 0  0 0 0
BASI  -1 0 0  0 1 0  0 0 -1
!
SYMM   -x, -y, -z
BASR   1 0 0  0 1 0  0 0 1
BASI   0 0 0  0 0 0  0 0 0
!
SYMM   -x+1/2, y, -z
BASR   0 0 0  0 0 0  0 0 0
BASI  -1 0 0  0 1 0  0 0 -1

```

INPUT FORMAT OF ATOM PARAMETERS FOR CONVENTIONAL JOBS

(LINE 25a $Jbt \neq \pm 10, \pm 15$)

The input format will depend mostly on Jbt

Single/multi Pattern have a common format:

Several lines are read for each of the Nat atoms, depending on the value of the variables Jbt in one hand, N_t and/or Isy on the other hand.

In the following the refinement codewords for a variable adopts the same name as the variable preceded by the character C.

For X-ray or nuclear Neutron scattering [$Jbt=0$]

Comment line : !Atom Typ X Y Z Biso Occ In Fin N_t Spc / Codes

LINE 25-1: Atom, Typ, X, Y, Z, Biso, Occ, In, Fin, N_t , Spc
(2A4, 5 reals, 4 integers)

LINE 25-2: CX, CY, CZ, CB, CN (Codewords 5 reals)

(Optional anisotropic temperature factors are read if $N_t=2$)

Comment line : ! beta11 beta22 beta33 beta12 beta13 beta23 /Codes

LINE 25-3: B11, B22, B33, B12, B13, B23 (7 reals)

LINE 25-3: CB11, CB22, CB33, CB12, CB13, CB23 (Codewords 8 reals)

(Optional Form factors coefficients are read if $N_t=4$ - 4 lines)

Comment line : ! Form-factor refinable parameters

LINE 25-3 f1 f2 f3 f4 f5 f6 f7 (7 reals)

LINE 25-4 Cf1 Cf2 Cf3 Cf4 Cf5 Cf6 Cf7 (Codewords 7 reals)

LINE 25-5 f8 f9 f10 f11 f12 f13 f14 (7 reals)

LINE 25-6 Cf8 Cf9 Cf10 Cf11 Cf12 Cf13 Cf14 (Codewords 7 reals)

(Optional special form factors are read if $Typ=SASH$ - 2 lines)

LINE 25-7: SASH-type, Ncoeff, Matrix (1 character, 1 integer, 9 reals)

LINE 25-8: Imp(3, 12) (3× Ncoeff, or 2× Ncoeff, integers)

For X-ray or nuclear Neutron scattering:

Structural model supplied by user (Rigid body refinement) [$Jbt=4,-4$]

Parameters defined by user in STRMOD. At present for Rigid Body Groups.

Comment line : !Atom Typ p1 p2 p3 p4 p5 p6 p7 p8

LINE 25-1: Atom, Typ, P1, P2, P3, P4, P5, P6, P7, P8 (2A4, 8 reals)

LINE 25-2: CP1, CP2, CP3, CP4, CP5, CP6, CP7, CP8 (Codewords 8 reals)

Comment line : ! p9 p10 p11 p12 p13 p14 p15 p16

LINE 25-3: P9, P10, P11, P12, P13, P14, P15 (7 reals)

LINE 25-4: CP9, CP10, CP11, CP12, CP13, CP14, CP15 (Codewords 7 reals)

For magnetic Neutron scattering [Jbt=1]

Parameters and codewords for fractional atomic co-ordinates, isotropic displacement (temperature) parameter, occupation number and magnetic moment components (LINE 25-1), imaginary components of the magnetic moments, and diagonal terms β_{ii} of the ATF tensor.

Comment line : !Atom Typ Mag Vek X Y Z Biso Occ Rx Ry Rz
LINE 25-1: Atom, Typ, Mag, Vek, X, Y, Z, Biso, Occ, RX, RY, RZ (2A4, 2 integers, 8 reals)
LINE 25-2: CX, CY, CZ, CB, CN, CRX, CRY, CRZ (Codewords 8 reals)
Comment line : ! Ix Iy Iz beta11 beta22 beta33 MagPh
LINE 25-3: IX, IY, IZ, B11, B22, B33, MagPh (7 reals)
LINE 25-4: CIX, CIY, CIZ, CB11, CB22, CB23, CMagPh (Codewords 7 reals)

For magnetic Neutron scattering [Jbt=-1]

Same as previous ($Jbt=1$) but the moment components are given in spherical coordinates

Comment line : ! Atom Typ Mag Vek X Y Z Biso Occ Rm Rphi Rtheta'
LINE 25-1: Atom, Typ, Mag, Vek, X, Y, Z, Biso, Occ, RM, Rphi, Rthet
(2A4, 2 integers, 8 reals)
LINE 25-2: CX, CY, CZ, CB, CN, CRM, CRPHI, CRTHET (Codewords 8reals)
Comment line : ! Im Iphi Itheta beta11 beta22 beta33 MagPh
LINE 25-3: Im, Iphi, Ithet, B11, B22, B33, MagPh (7 reals)
LINE 25-4: CIX, CIY, CIZ, CB11, CB22, CB23, CMphas (Codewords 7 reals)

For magnetic Neutron scattering [Jbt=-1 and Isy=-2]

Same as previous ($Jbt=1$) but the moment components are given by a linear combination of the basis functions defined in the [Line 24](#) affected to the coefficients C_i

(Coefficients for basis functions, C_i)

Comment line : ! Atom Typ Mag Vek X Y Z Biso Occ C1 C2 C3
LINE 25-1: Atom, Typ, Mag, Vek, X, Y, Z, Biso, Occ, C1, C2, C3 (2A4, 2 integers, 8 reals)
LINE 25-2: CX, CY, CZ, CB, CN, CC1, CC2, CC3 (Codewords 8 reals),
Comment line : ! C4 C5 C6 C7 C8 C9 MagPh
LINE 25-3: C4, C5, C6, C7, C8, C9, MagPh (7 reals)
LINE 25-4: CC4, CC5, CC6, CC7, CC8, CC9, CMPhas (Codewords 7 reals)

Magnetic model supplied by user. At present conical structures in real space **[Jbt=5,-5]**

LINE 25-1: Atom, Typ, Mag, Vek, P1, P2, P3, P4, P5, P6, P7, P8 (2A4, 2 integers, 8 reals)
Comment line : !Atom Typ Mag Vek X Y Z Biso Occ Mom beta Phase
Line 25-2: CP1, CP2, CP3, CP4, CP5, CP6, CP7, CP8 (Codewords 8reals)
Line 25-3: P9, P10, P11, P13, P14, P15, P12 (
Comment line : ! Phi & Theta of Cone-axis + unused params
Line 25-4: CP9, CP10, CP11, CP13, CP14, CP15, CP12 (Codewords 8reals)

Common variables to all models

Atom – Identification name

Identification characters for atom or object.

Typ - Link to scattering data

Link to scattering data of the atom: either [NAM](#) from [LINE 12](#) or chemical symbol and valence to access internal table (use only upper case letters). See notes given in [LINE 12](#). Also a series of special form factors are available with refinable parameters. For using this option [Typ](#) should be

equal to one of the following words: SPHS, SPHE, SASH, ELLI, DISK, TORE (not available yet), FUD1, FUD2, FUD3, FUD4 are [Dum](#) my symbols that may be introduced for special form-factor refinements, the code calculating the form-factor must be included in the subroutine Form_Factor. In case of special form-factor the content of the variable [Atom](#) must start with the chemical symbol to normalise the scattering density. See Mathematical section for details.

Variables describing the nuclear structure

[X, Y, Z](#) - Fractional atomic coordinates

Fractional atomic coordinates

[Biso](#)- Isotropic displacement (temperature) parameter

Isotropic displacement (temperature) parameter in \AA^2

[Occ](#) - Occupation number

Occupation number i.e. chemical occupancy \times site multiplicity (can be normalised to the multiplicity of the general position of the group).

[In](#), [Fin](#) – Subset of symmetry operator

Ordinal number of first and last symmetry operator applied to the atom, apart from the identity which must always be the first one.

Useful to describe pseudo-symmetries. This option is normally used when the user supply their own list of symmetry operators ([Isy](#)=1). Be careful with multiplicity of reflections!. It is suggested that the users supply also their list of reflections. If [In](#)=[Fin](#)=0 all the symmetry operators are applied. Only used for crystallographic structures.

[N t](#) – Atom type

=0 Isotropic atom. No anisotropic temperature factors are given.

=2 Anisotropic atom. The anisotropic temperature factors should be given below.

=4 The form-factor of this atom is calculated using a special subroutine and refinable parameters should be given below (under test!)

[Spc](#) - Number of the chemical specie

Number of the chemical specie. Used for Bond Valence calculations, see [LINE 21](#) for details.

Variables specific to magnetic structures

[Mag](#) - Magnetic rotation matrix identifier

Ordinal number of the magnetic rotation matrices applied to the magnetic moment of the atom. To be given only in the case of a magnetic phase

[Vek](#) - Propagation vector identifier

Number of the propagation vector to which the atom contributes. If [Vek](#)=0 the atom is used for all the propagation vectors in the calculation of structure factor. If [Vek](#)<0 the atom contributes to $\mathbf{VK}(\text{abs}(\mathbf{Vek}))$ and to the vector $\mathbf{VK}(\text{abs}(\mathbf{Vek})+\mathbf{Nvk}/2)$

[RX](#), [RY](#), [RZ](#) – Cartesian magnetic moments components

Components along the crystallographic axis of the magnetic moments, in units of Bohr magnetons.

[RM](#), [Rphi](#), [Rthet](#)– Spherical magnetic moments components

In the case [Jbt](#)=-1 these three parameters correspond to the spherical components of the magnetic moment \mathbf{M} , in the following order: (μ , ϕ , θ). μ : magnitude of the Fourier component of magnetic moment, ϕ and θ are spherical angles of vector \mathbf{M} (see note on magnetic refinements).

If the magnetic phase is incommensurate or described in the crystallographic cell with the help of a propagation vector, these components ([RX](#), [RY](#), [RZ](#) or [RM](#), [Rphi](#), [Rthet](#)) are actually the real part of the Fourier component of the magnetic moment of the atom (S_k).

[IX](#), [IY](#), [IZ](#) – Imaginary Cartesian magnetic moment components

Imaginary components of the Fourier coefficient of magnetic moment:

[Im](#), [Iphi](#), [Ithet](#) – Imaginary spherical magnetic moment components

Spherical components as for real components.

[C1](#), [C2](#), [C3](#), [C4](#), [C5](#), [C6](#), [C7](#), [C8](#), [C9](#) – Coefficients of the basis functions

Coefficients of the linear combinations of basis functions.

[MagPh](#) - Magnetic phase

Magnetic phase of the atom in units of 2π (see [Mathematical section](#))

The units of [IX](#), [IY](#), [IZ](#) and [Im](#) are in Bohr magnetons. The angles [Iphi](#), [Ithet](#) are in degrees, the vector corresponding to the Fourier component of the magnetic moments is defined as (see [Mathematical section](#)):

$$S_k = 1/2 (\mathbf{R}_k + i \mathbf{I}_k) \exp(-2\pi i \text{MagPh})$$

The components of the real, \mathbf{R}_k , and imaginary, \mathbf{I}_k , parts are given with respect to a basis of unit vectors along the crystallographic unit cell.

If [Hel](#)=1 (see [LINE 19](#)) the third component [IZ](#) is calculated by the program in order to have an imaginary vector orthogonal to the real vector ($\mathbf{R}_k \cdot \mathbf{I}_k = 0$). If [Jbt](#)<0, then the ϕ -angle of the imaginary part is calculated by the program for keeping the orthogonal constraint.

Options-I: anisotropic temperature factors and form factors

[B11](#), [B22](#), [B33](#), [B12](#), [B13](#), [B23](#) – Anisotropic temperature factors

Anisotropic displacement (temperature) parameters (β_{ij}) are all refinable for X-ray or nuclear Neutron scattering, if [Jbt](#)=0. For [Jbt](#)=1,-1, only the diagonal elements [B11](#), [B22](#), [B33](#) of the anisotropic temperature factors tensor are refinable.

[f1](#) to [f14](#) – Refinable form-factor coefficients

The parameters [f1](#) to [f14](#) are used for X-ray or nuclear Neutron scattering ([Jbt](#)=0) if [N_t](#) is set to 4: they allow describing the form-factor of the current object and the corresponding codewords are given by [Cf1-Cf14](#) ([Line 25-4/6](#)). The meaning of the coefficients is explained in [Mathematical section](#).

Options-II: Special form factors

[SASH-type](#) – Special form factor

Label indicating if real spherical harmonics ([SASH-type](#) =ylmp) or cubic harmonics ([SASH-type](#) =klj) are used to describe the scattering density.

[Ncoeff](#) – Order of the SH development

Number of coefficients used in the description of the scattering density.

Matrix – Rotation matrix from crystallographic to local axes

Matrix transforming the crystallographic Cartesian axes to the local Cartesian axes for the representative position of the molecular centre. If not given, the program uses the identity matrix.

Imp(3, 12) – Spherical harmonics-SH indices

List of real spherical harmonics indices: **Ncoeff** triplets (l, m, p) for **SASH-type** =yImp, or

List of cubic harmonics indices: **Ncoeff** pairs (l, j) for **SASH-type** = klj.

The form-factor corresponding to **Tvp** =SASH is well adapted to nearly free molecular rotations. For a molecule with N atoms rotating around its centre of mass (supposed to be in a particular crystallographic site) the molecular form factor is given by:

$$f(\mathbf{Q}) = 4\pi \sum_{slmp} c_{lmp}^s i^l j_l(Qr_s) b_s y_{lmp}(\theta, \varphi)$$

or

$$f(\mathbf{Q}) = 4\pi \sum_{slj} c_{slj}^s i^l j_l(Qr_s) b_s K_{lj}(\theta, \varphi)$$

Where the index *s* runs from 1 to N and corresponds to N spherical shells. $j_l(x)$ is the spherical Bessel function of order *l*. $K_{lj}(\theta, \varphi)$ and $y_{lmp}(\theta, \varphi)$ are the cubic harmonics and the real spherical harmonics, respectively, as defined in M. Kara and K. Kurki-Suonio, *Acta Cryst* **37**, 201 (1981). b_s is the scattering length (or X-ray scattering factor). The coefficients c_{lj}^s and c_{lmp}^s are free parameters.

In ***FullProf*** the form-factor **Tvp** =SASH corresponds to just *one shell* (a particular value of *s*). For a complete molecule the user must provide a number of SASH objects equal to the number of spherical shells characterising the molecule. The first parameter in the list (**f1**, **f2**, **f3**, ...) corresponds to the value of the spherical shell radius: **f1**=*r*, the second parameter is the number of atoms of the chemical species given by **Atom**, within the spherical shell. The coefficients (**f3**, **f4**, **f5**, ...**f14**) correspond to the free parameters c_{lj}^s or c_{lmp}^s of the current shell in the order specified by the list **Imp(3, 12)**. The first coefficient **f3** corresponds to the free rotator term $l=m=0$ (or $l=j=0$) and should, normally, be fixed to 1. The user must provide the list (l, m, p)/(l, j) in ascending order on *l* according to the local site symmetry. See Table 2 in the above reference.

Example:

User defined structural or magnetic models

P1 to P15

For **Jbt**=4,-4, the parameters are defined by user in the subroutine STRMOD. At present the parameters correspond to the Rigid Body constraints. The meaning of the parameters is explained in the Rigid Body Refinements section.

For **Jbt**=5,-5 the parameters are defined by user in the subroutine MAGMOD. At present the parameters correspond to the treatment of conical structures in real space.

INPUT FORMAT OF ATOM PARAMETERS FOR COMBINED JOBS

(LINE 25b, FOR $J_{bt}=\pm 10, \pm 15$)

(For X-rays or nuclear + magnetic Neutron scattering $J_{bt}=\pm 10$)

Comment lines : !Atom Typ Mag Vek X Y Z Biso Occ N_type Spc
 !Line below:Codes
 or
 ! Rx Ry Rz Ix Iy Iz MagPh
 !Line below:Codes
 or
 ! Rm Rphi Rtheta Im Rphi Rtheta MagPh
 !Line below:Codes
 or
 ! C1 C2 C3 C4 C5 C6 MagPh
 !Line below:Codes
 or
 ! beta11 beta22 beta33 beta12 beta13 beta23 /Line below:Codes
 or
 ! Form-factor refinable parameters

LINE 25-1: Atom, Typ, Mag, Vek, X, Y, Z, Biso, Occ N_t Spc (a)
 (2a4,2 integers,5 reals,2 integers)
 LINE 25-2: CX, CY, CZ, CB, COC (b) (5 reals)
 (Optional magnetic moment components in cartesian coordinates, read if $N_t=1,3$)
 LINE 25-3: RX, RY, RZ IX, IY, IZ MagPh (c) (7 reals)
 LINE 25-4: CRX CRY CRZ CIX CIY CIZ CMPhas (d) (7 reals)
 (Optional magnetic moment components in spherical coordinates, read if $N_t=-1,-3$)
 LINE 25-3: RM Rphi Rthet Im Iphi Ithet MagPh (c) (7 reals)
 LINE 25-4: CRM CRPHI CRTHET CIM CIPHI CITHET CMPhas (d) (7 reals)
 (Optional coefficient for the linear combination of basis functions, read if $N_t=1,3$ and $Isy=-2$)
 LINE 25-3: C1 C2 C3 C4 C5 C6 MagPh (c) (7 reals)
 LINE 25-4: CC1 CC2 CC3 CC4 CC5 CC6 CMPhas (d) (7 reals)
 (Optional Anisotropic temperature factor parameters, read if $N_t=2,3,-3$, and $Isy=-2$)
 LINE 25-5: B11 B22 B33 B12 B13 B23 (e) (6 reals)
 LINE 25-6: CB11 CB22 CB33 CB12 CB13 CB23 (f) (6 reals)
 (Optional Form factors coefficients and codes (4 lines) are read if $N_t=4$)
 LINE 25-7: f1 f2 f3 f4 f5 f6 f7 (g) (7 reals)
 LINE 25-8: Cf1 Cf2 Cf3 Cf4 Cf5 Cf6 Cf7 (h) (7 reals)
 LINE 25-9: f8 f9 f10 f11 f12 f13 f14 (i) (7 reals)
 LINE 25-10: Cf8 Cf9 Cf10 Cf11 Cf12 Cf13 Cf14 (j) (7 reals)

Note that:

If $N_t = 0$ Only lines (a) and (b) need to be given
 If $N_t = 1$ give the lines (a), (b), (c) and (d)
 If $N_t = 2$ give the lines (a), (b), (e) and (f)
 If $N_t = 3$ give the lines from (a) to (f)
 if $N_t = 4$ give the lines (a), (b), (g),(h),(i) and (j) (special form-factor)

This input could be also used for X-rays, in such case Mag and Vek should be zero for all the atoms and Job or Jtyp(n)=0. In such case the space group symbol can be used for generation of reflections and symmetry operators (with Irf set to 0).

For a phase with magnetic contributions Typ should be equal to the magnetic form factor symbol. The program extracts internally the Fermi length symbol from Typ. If there are magnetic contributions the symmetry should be controlled by the user (Isy= $\pm 1,-2$) and the magnetic part should be described with the

formalism of propagation vectors, the magnetic contribution is calculated only for the satellite reflections. If fundamental reflections have magnetic contribution the propagation vector $\mathbf{k}=(0,0,0)$ must be included explicitly if there are other propagation vectors. If the magnetic cell is the same as the chemical cell propagation vectors are not needed.

The symmetry operators must belong to the group of the propagation vector $G_{\mathbf{k}}$, so some atoms need, in general, to be repeated for the rest of positions not generated by $G_{\mathbf{k}}$.

For X-rays or nuclear + magnetic Neutron scattering (Modulated structures)

Jbt=±15:

At present only rational components of the propagation vectors are allowed (Commensurate modulated structures). The calculation of the structure factor is made exactly

Comment lines : !Atom Typ Mag Vek X Y Z Biso Occ N_type Nki
 !Line below:Codes
 or
 ! Rx Ry Rz Ix Iy Iz MagPh Dep Dvek
 !Line below:Codes
 or
 ! Rm Rphi Rtheta Im Rphi Rtheta MagPh Dep Dvek
 !Line below:Codes
 or
 ! beta11 beta22 beta33 beta12 beta13 beta23 /Line below:Codes
 or
 ! Form-factor refinable parameters

LINE 25-1: Atom, Typ, Mag, Vek, X, Y, Z, Biso, Occ N_t Ndvk (a)
 (2a4,2 integers,5 reals,2 integers)

LINE 25-2: CX, CY, CZ, CB, COC (b) (5 reals)

(Optional magnetic moment components in cartesian coordinates, read if N_t=1,3)

LINE 25-3: RX, RY, RZ IX, IY, IZ, MagPh (c) (7 reals)

LINE 25-4: CRX CRY CRZ CIX CIY CIZ CMPhas (d) (7 reals)

(Optional magnetic moment components in spherical coordinates, read if N_t=-1,-3)

LINE 25-3: RM Rphi Rthet Im Iphi Ithet MagPh (c) (7 reals)

LINE 25-4: CRM CRPHI CRTHET CIM CIPHI CITHET CMPhas (d) (7 reals)

(Optional real and imaginary components of the modulated displacement parameters are read if Ndvk≠0, in Ndvk pairs of lines)

LINE 25-3: Dx Dy Dz Dxi Dyi Dzi DPhas Dep Dvek (c) (7 reals)

LINE 25-4: CDx CDy CDz CDxi CDyi CDzi CDPhas (d) (7 reals)

(Optional Anisotropic temperature factor parameters, read if N_t=2,3,-3, and Isy=-2)

LINE 25-5: B11 B22 B33 B12 B13 B23 (e) (6 reals)

LINE 25-6: CB11 CB22 CB33 CB12 CB13 CB23 (f) (6 reals)

(Optional Form factors coefficients and codes (4 lines) are read if N_t=4)

LINE 25-7: f1 f2 f3 f4 f5 f6 f7 (g) (7 reals)

LINE 25-8: Cf1 Cf2 Cf3 Cf4 Cf5 Cf6 Cf7 (h) (7 reals)

LINE 25-9: f8 f9 f10 f11 f12 f13 f14 (i) (7 reals)

LINE 25-10: Cf8 Cf9 Cf10 Cf11 Cf12 Cf13 Cf14 (j) (7 reals)

[If Jdi = 2] The user inputs after all the atoms, a cell transformation matrix T_{ij} , and an origin shift vector Or_sh, as explain above (see **LINE 19-1**)

Comment line : ! Multiple Cell Transformation

Line 25-11-1 T11 T12 T13 Or_Sh1 (4 reals)

Line 25-11-2 T21 T22 T23 Or_Sh2

Line 25-11-3 T31 T32 T33 Or_Sh3

Note that

If N_t = 0 Only lines (a) and (b) need to be given

If N_t = 1 give the lines (a), (b), (c) and (d)

If N_t = 2 give the lines (a), (b), (e) and (f)

If N_t = 3 give the lines from (a) to (f)

if $N_t = 4$ give the lines (a), (b), (g),(h),(i) and (j) (special form-factor)

The symmetry operators must belong to the group of the propagation vector G_k , so some atoms need, in general, to be repeated for the rest of positions not generated by G_k .

Dx, Dy, Dz, Dxi, Dyi, Dzi, Phas, Dep, Dvek – Displacements parameters

The displacement \mathbf{u}_{ni} of an atom i in the n^{th} cell is calculated as:

$$\mathbf{D}_{nk} = \frac{1}{2} (\mathbf{RD}_{nk} + i \mathbf{ID}_{nk}) e^{-2i\pi D_{phas}}, \text{ and } \mathbf{u}_{ni} = \sum_{nk=1}^{N_{dvk}} \mathbf{D}_{nk} e^{-2i\pi \mathbf{PVK}(D_{vek}(nk)) \cdot \mathbf{R}_n}$$

Where the components of the vectors \mathbf{RD}_{nk} and \mathbf{ID}_{nk} ($nk=1 \dots N_{dvk}$) are defined in each line where **Dx, Dy, Dz, Dxi, Dyi, Dzi** are inputed. The propagation vector **PVK** is one of the propagation vectors of the phase. It is identified by the code (**Dvek**(nk)) that gives the ordinal number of **PVK** in the list of propagation vectors. This list is either read in the $(k_1, k_2, \dots, k_{Nvk})$ list inputed in the **LINE 44** (if $Nvk > 0$) or found in the list of $(k_1, k_2, \dots, k_{Nvk}, -k_1, -k_2, \dots, -k_{Nvk})$ of propagation vectors, generated from the given list of **LINE 44** in case $Nvk < 0$.

SINGLE CRYSTAL DATA

(LINE 26-30 have the following format, which is read if $lrf=4$ on **LINE 19**, the next line to be read after begins at least at **Line 43**)

Input format

Common Single/Multi-Pattern format:

*This specific block lines (see, [loop over phases](#)) should be given for a phase contributing to a set of integrated intensities if $lrf=4$ on **LINE 19**:*

Comment line: **! Scale Factors**

Comment line: **! Sc1 Sc2 Sc3 Sc4 Sc5 Sc6**

LINE 26: SC1, SC2, SC3, SC4, SC5, SC6 (6 reals)

LINE 26-1: CSC1, CSC2, CSC3, CSC4, CSC5, CSC6 (Codewords 6 reals)

Comment line: **! Extinction Parameters**

Comment line: **! Ext1 Ext2 Ext3 Ext4 Ext5 Ext6 Ext7**

LINE 27: EXT1, EXT2, EXT3, EXT4, EXT5, EXT6, EXT7 (7 reals)

LINE 27-1: CEXT1, CEXT2, CEXT3, CEXT4, CEXT5, CEXT6, CEXT7 (Codewords 7 reals)

Comment line: **! a b c alpha beta gamma**

LINE 29: a, b, c, alpha, beta, gamma (6 reals)

LINE 29-1: ca, cb, cc, cd, ce, cf (6 reals)

Comment line: **! Not yet used parameters**

LINE 30: Par1...Par5 (5 reals)

LINE 30-1: cpar1,...,cpar5 (5 reals)

REMARK: the next line to be read after in this mode begins at least at **Line 43**

Variables

SC1, SC2, SC3, SC4, SC5, SC6 – Scale factors, 6 domains allowed

Scale factor for different domains or for heterogeneous integrated intensity data. For powder data only the first scale factor is used

EXT1, EXT2, EXT3, EXT4, EXT5, EXT6, EXT7 – Extinction parameters

Extinction parameters

At present only the first extinction parameter is used for single crystal work.

a, b, c, alpha, beta, gamma – Cell parameters

Cell parameters in Å an angles in degrees

The codewords for cell constants must be equal to zero if there is no powder profile to be refined simultaneously with the integrated intensity data (Cry≠0).

Par1 – Second order reflexion contamination

The structure factor can be corrected for $\lambda/2$ contamination :

$$F_{calc} = F_{calc}(\mathbf{H}) + Par1.F_{calc}(2\mathbf{H})$$

POWDER DATA-I: PROFILE PARAMETERS FOR EACH PATTERN

(LINE 26-33 have the following format if Cry=0 and Irf≠4)

Input format depend on Uni and Npr

Common Single/Multi-Pattern format:

This specific sub-block of the block of lines 26 to 38 (see, [loop over phases](#)) should be given for each pattern to which the phase contributes (controlled by JCONTR(n_pat) in [line 19](#). These lines may be grouped as:

Sample parameters (excepted the Shape1 profile parameter)

The format of this line will depend on Uni

(If Uni=0: [2 θ] case:)

Comment line:! Scale Shape1 Bov Str1 Str2 Str3 Strain-Model

LINE 26 [2 θ]: Scale, Shape1, Bov, Str1, Str2, Str3, Strain-Model (6 reals – 1 integer)

LINE 26-1 [2 θ]: CSCALE, CSHAPE1, CBOV, CSTR1, CSTR2, CSTR3 (Codewords 6 reals)

(If Uni=1: [T.O.F] case:)

Comment line:! Scale Extinc Bov Str1 Str2 Str3 Strain-Model

LINE 26 [T.O.F]: Scale, Extinc, Bov, Str1, Str2, Str3, Strain-Model(6 reals)

LINE 26-1 [T.O.F]: CSCALE, CEXTI, CBOV, CSTR1, CSTR2, CSTR3 (Codewords 6 reals)

Profile parameters I:

A specific line 27 should be given for each pattern to which the phase contributes (controlled by JCONTR(n_pat) in [line 19](#)). The format will depend on Uni and Npr

(If Uni=0: [2 θ] case:)

(For Npr≠11, Line 27 have the format):

Comment line:! U V W X Y GauSiz LorSiz Size-Model

LINE 27 [2 θ]: U, V, W, X, Y, GausSiz, LorSiz, Size-Model (7 reals – 1 integer)

LINE 27-1 [2 θ]: CU, CV, CW, CX, CY, CIG, CSZ (Codewords 7 reals)

[For Npr=11, Line 27 have a different format]

Comment line:! U1 V1 W1 X1 Y GauSiz LorSiz Size-Model

LINE 27 [2 θ]: UL, VL, WL, XL, Y, GausSiz, LorSiz, Size-Model (7 reals – 1 integer)

LINE 27-1 [2 θ]: CUL, CVL, CWL, CY, CIG, CSZ (Codewords 7 reals)

Comment line:! Ur Vr Wr Eta0r Xr
 LINE 27-2 [20]: Ur, Vr, Wr, Eta0r, Xr (5 reals)
 LINE 27-3 [20]: CUr, CVr, CWr, CEta0r, CXr (Codewords 5 reals)

(If Uni=1: [T.O.F] case:)

Comment line:! Sig-2 Sig-1 Sig-0 Xt Yt Z1 Zo Size-Model
 LINE 27: [T.O.F]: Sig-2, Sig-1, Sig-0, Xt, Yt, Z1, Z0, Size-Model (7 reals)
 LINE 27-1 [T.O.F]: CSIG2, CSIG1, CSIG0, CXt, CYt, CZ1, CZ0 (Codewords 7 reals)
 Comment line:! Gam-2 Gam-1 Gam-0 LStr Lsiz
 LINE 27-2 [T.O.F]: Gam-2, Gam-1, Gam-0, LStr, Lsiz (5 reals)
 LINE 27-3 [T.O.F]: CGAM2, CGAM1, CGAM0, CLStr, CLsiz (Codewords 5 reals)

Sample cell parameters:

Comment line:! a b c alpha beta gamma
 LINE 29 [20]: a, b, c, alpha, beta, gamma (6 reals)
 LINE 29-1 [20]: ca, cb, cc, cd, ce, cf (Codewords 6 reals)

Comment line:! a b c alpha beta gamma
 LINE 29 [T.O.F]: a, b, c, alpha, beta, gamma (6 reals)
 LINE 29-1 [T.O.F]: ca, cb, cc, cd, ce, cf (Codewords 6 reals)

Preferred orientation, asymmetry, profile-II:

A specific lines 30-33 should be given for each pattern to which the phase contributes (controlled by JCONTR(n_pat) in line 19). The format will depend on Uni and Npr

(If Uni=0: [20] case:)

(If Npr≠7 LINE 30 has the format)

Comment line:! Pref1 Pref2 Asy1 Asy2 Asy3 Asy4
 LINE 30: Pref1, Pref2, PA1, PA2, PA3, PA4 (6 reals)
 LINE 30-1: CG1, CG2, CPA1, CPA2, CPA3, CPA4 (6 reals)
 (If Npr=7, two extra parameters are read in LINE 30)
 Comment line:! Pref1 Pref2 Asy1 Asy2 Asy3 Asy4 S_L D_L
 LINE 30: Pref1, Pref2, PA1, PA2, PA3, PA4, S_L, D_L (8 reals)
 LINE 30-1: CG1, CG2, CPA1, CPA2, CPA3, CPA4, CS_L, CD_L (8 reals) [For Npr=7]

(Optional line 32, read if If Ratio < 0)

Comment line:! Additional U,V,W parameters for Lambda2
 LINE 32: U2, V2, W2, V2, W2 (3 reals)
 LINE 32-1: CU2, CV2, CW2 (Codewords 3 reals)

(Optional line 33, read if Npr=4, or Npr>7)

Comment line:! Additional Shape Parameters
 LINE 33: SHP1, CSHP1, SHP2, CSHP2 (4 reals, 2 params, two codes)

(If Uni=1: [T.O.F] case:)

Comment line:! Pref1 Pref2 alph0 beta0 alpha1 beta1
 LINE 30 [T.O.F]: Pref1, Pref2, ALPH0, BETA0, ALPHA1, BETA1 (6 reals)
 LINE 30-1 [T.O.F]: CG1, CG2, CALPH0, CBETA0, CALPHA1, CBETA1 (Codewords 6 reals)

(Optional line, read if Npr=10)

Comment line:! alph0t beta0t alph1t beta1t
 LINE 31 [T.O.F]: ALPH0T, BETA0T, ALPHA1T, BETA1T (4 reals)
 LINE 31-1 [T.O.F]: CALPH0T, CBETA0T, CALPH1T, CBETA1T (Codewords 4 reals)

Comment line: Absorption Correction Parameters

LINE 33 [TOF]: ABS1, CABS1, ABS2, CABS2

(4 reals 2 params, two codes)

Sample parameters

Scale – Scale factor

Scale factor

Extinc - Extinction parameter for powders

Extinction parameter for powders

Bov - Overall isotropic displacement

Overall isotropic displacement (temperature) factor in \AA^2

Str1, Str2, Str3 - Strain parameters

Strain parameters, defined through the subroutine STRAIN (see Microstructure in *FullProf* section). If Str=1 set these values to 0.0

Anisotropic Gaussian contribution of micro-strain. It is calculated in subroutine STRAIN as a function of Strain-Model and/or Str. If Str=1 and Strain-Model $\neq 0$ then the notation of P. Stephens is used. DST depends on STR1, STR2,...parameters and *hkl*. See Mathematical section for details.

Strain-Model - Strain model selector

Integer to select a particular model for strains in subroutine STRAIN. This variable depends on the pattern, but, to be consistent it should normally be the same for all patterns to which the current phase is contributing.

CSCALE, CSHAPE1, CBOV, CEXTI

Codewords for scale factor, shape parameter, extinction, and overall temperature factor.

CSTR1, CSTR2, CSTR3

Remark on codewords for strain parameters, if Str=1 set these values to 0.0

Size-Model - Size model selector

Integer to select a particular model for LorSiz in subroutine SIZEF.

Profile parameters-I, 2theta

Shape1 - Profile shape parameter

Profile shape parameter

E.g.: η_0 for Npr=4,5 but not for Npr=7, in which case it is not used.

m_0 for Npr=6

U, V, W - Half-width parameters

Half-width parameters (normally characterising the instrumental resolution function).

UL, VL, WL, XL - Left Half-width parameters

Left Half-width parameters for split pseudo-Voigt profile Npr=11.

Ur, Vr, Wr, Eta0r, Xr - Right Half-width and shape parameters

FWHM and shape parameters for the right part of the split pseudo-Voigt function. This function is similar to [Npr=5](#) but the left ($x < 0$) and right ($x > 0$) parts of the profile have different U, V, W, η_0 and [X](#) parameters. Additional shape parameters are also read.

[X](#) - Lorentzian isotropic strain

Lorentzian isotropic strain parameter for [Npr=7](#).

[Y](#) - Lorentzian isotropic size

Lorentzian isotropic size parameter for [Npr=7](#).

[GausSiz](#) - Isotropic size parameter of Gaussian character

Isotropic size parameter of Gaussian character.

[LorSiz](#) - Anisotropic Lorentzian contribution of particle size

Anisotropic Lorentzian contribution of particle size. The function F is calculated in subroutine `SIZEF` and depend on parameter [LorSiz](#) and hkl . Different F -functions are selected by [Size-Model](#).

[LStr](#) - Lorentzian isotropic strain

Lorentzian isotropic strain

$DSIZ = F(Lsiz)$, F depends on [Lsiz](#) (and eventually on more size parameters) through the selected size model

Profile parameters-I, T.O.F

[Sig-2](#), [Sig-1](#), [Sig-0](#) – Variance of the gaussian component of the TOF peak profile

The variance of the Gaussian component of the peak shape in TOF neutrons is given by:

$$\sigma^2 = (Sig2 + GSIZ)d^4 + (Sig1 + DST)d^2 + Sig0$$

where d is the d-spacing in angstroms

Units: [Sig-2](#) (microsecs/Å²)²; [Sig-1](#) (microsecs/Å²); [Sig-0](#) (microsecs²)

[Xt](#), [Yt](#) - Unused

Not used at present

[Z1](#) - Gaussian isotropic size

$GSIZ$: Gaussian isotropic size component (microsecs/Å²)², this parameter cannot be refined simultaneously with [Sig-2](#).

[Z0](#) - Unused

Not used at present

[Gam-2](#), [Gam-1](#), [Gam-0](#) - FWHM parameters of the Lorentzian component of the TOF peak profile

The FWHM of the Lorentzian component of the peak shape in TOF neutrons is given by:

$$\gamma = (gam2 + DSIZ)d^2 + (gam1 + LStr)d + gam0$$

where d is the d-spacing in angstroms

Units: [Gam-2](#) (microsecs/Å²)

[Gam-1](#) (microsecs/Å)

[Gam-0](#) (microsecs)

Sample cell parameters:

a, b, c, alpha, beta, gamma – Cell parameters

Cell parameters in Å and angles in degrees. If all diffraction patterns are well calibrated the cell constants should be the same for all patterns, and therefore the cell parameters should be constrained to be the same for all patterns.

Codewords for cell constants A, B, C, D, E and F defined by :

$$\frac{1}{d^2} = Ah^2 + Bk^2 + Cl^2 + Dkl + Ehl + Fhk$$

Note that these codewords do not refer directly to the cell parameters; for instance, in the hexagonal system, the last codeword CF must be the same as CA and CB.

Preferred orientation, asymmetry, profile II:

Pref1, Pref2 - Preferred orientation parameters

Preferred orientation parameters (see Mathematical section)

- when Nor = 0, G1 = 0 means no preferred orientation
- when Nor = 1, G1 = 1 means no preferred orientation

PA1, PA2, PA3, PA4 - Asymmetry parameters

Asymmetry parameters applied to angles below AsymLim (given on LINE 8)

S L, D L

Asymmetry parameters corresponding to the L. Finger formulation of the axial divergence.

ALPH0, BETA0, ALPH1, BETA1 - Exponential decay parameters for TOF patterns

Parameters defining the variation of the exponential decay function with d-spacing.

For Npr=9

- Fast decay: $\alpha = \alpha_0 + \alpha_1/d$
- Slow decay: $\beta = \beta_0 + \beta_1/d^4$

For Npr=10, the parameters ALPH0, BETA0, ALPH1, BETA1 correspond to the epithermal component of the neutron spectrum. In this case the TOF peak positions and decay parameters versus d-spacing are calculated using the following expressions:

$$\begin{aligned} TOF_e &= \text{Zero} + \text{Dtt1} d \\ TOF_t &= \text{Zero} + \text{Dtt1} d - \text{Dtt2} d^1 \\ n_{\text{cross}} &= 0.5 \operatorname{erfc}(\text{Width} (\text{x-cross}-d^1)) \\ TOF &= n_{\text{cross}} TOF_e + (1-n_{\text{cross}}) TOF_t \end{aligned}$$

Where *erfc* is the complementary error function.

- Fast and slow decay for epithermal component: $\alpha_e = \alpha_{e0} + \alpha_{e1} d$, $\beta_e = \beta_{e0} + \beta_{e1} d$
- Fast and slow decay for thermal component: $\alpha_t = \alpha_{t0} - \alpha_{t1} d^1$, $\beta_t = \beta_{t0} - \beta_{t1} d^1$

The final fast and slow decay parameters are calculated like the case of TOF peak positions:

$$\begin{aligned} \alpha &= n_{\text{cross}} \alpha_e + (1-n_{\text{cross}}) \alpha_t \\ \beta &= n_{\text{cross}} \beta_e + (1-n_{\text{cross}}) \beta_t \end{aligned}$$

The parameters for calculating the thermal fast and slow decay functions are read in [LINE 31](#).

ALPH0T, BETA0T, ALPH1T, BETA1T

Parameters defining the variation of the exponential decay functions with d-spacing for the thermal component of the neutron spectrum. Only given for [Npr=10](#). See [LINE 30](#) for the expressions used to calculate the final α and β decay parameters.

U2, V2, W2 – 2nd wavelength FWHM parameters

Parameters for the second wavelength present in the diffraction pattern.

SHP1, SHP2 – Additional profile parameters

- For [Npr=11](#) (split pseudo-Voigt) the additional shape parameters correspond to the additional contribution to the FWHM for the Left (L) and Right (R) part of the profile for $2\theta < 90^\circ$ and $2\theta > 90^\circ$ respectively.

$$addFWHM^2(L) = Shp1 / \tan^2 2\theta$$

$$addFWHM^2(R) = Shp2 / \tan^2 2\theta$$

- For [Npr=12](#) (Convolved pseudo-Voigt with L. Finger axial divergence asymmetry)
 - [SHP1](#) = [S L](#) is source width/detector distance
 - [SHP2](#) = [D L](#) is detector width/detector distanceThe parameters [S L](#) and [D L](#) play the role of asymmetry parameters, they are used only for reflections below $2\theta =$ [AsymLim](#).

ABS1, ABS2 - Absorption correction for TOF

Absorption correction parameters

The physical meaning of these parameters depend on the function selected by [Iabsco](#) in [LINE 8](#).

POWDER DATA-II: hkl-DEPENDENT SHIFT PARAMETERS

(Optional LINE 34-35 are read if $Sol \neq 0$ in [LINE 19](#))

Input format

Common Single/Multi-Pattern format:

This specific optional sub-block of the block of lines 26 to 38 (see, [loop over phases](#)) should be given for each pattern to which the phase contributes (controlled by [JCONTR](#)(n_pat) in [line 19](#). These lines may be grouped as:

Shift parameters:

At the present, none of the user supplied subroutines for shifts use the SHF_i parameters, excepted [Shf_i](#) when [ModS](#) = ±1. Other built-in available models can be selected by [ModS](#) and are detailed afterwards.

Comment line: ! Shift1 Shift2 Shift3 ModS
LINE 34: [SHF1](#), [SHF2](#), [SHF3](#), [ModS](#) (3 reals- 1 integer)
LINE 34-1: [CSHF1](#), [CSHF2](#), [CSHF3](#) (Codewords 3 reals)

Shift models:

(1) Shift due to the deviation of the (hkl) direction with respect to a specific reciprocal direction by an angle φ : [ModS](#) = 1, $\cos\varphi$ dependance, [ModS](#) = -1, $\sin\varphi$ dependance of the shift
(If [ModS](#) = ±1, an optional line is read)

Comment line: ! Shift-cos(1) or Shift-sin(-1) axis
LINE 35: [Sh1](#), [Sh2](#), [Sh3](#) (3 reals)

(2) User defined selective shifts for specific (hkl) reflexions
(IF [ModS](#) < -1, and $ABS(\text{ModS}) \leq 10$, [ABS\(\text{ModS}\)](#) optional lines are read)

Comment line: ! Shift integers ($n1.h + n2.k + n3.l = n4.n + n5$) Shift-par Code
LINE 35: [n1](#), [n2](#), [n3](#), [n4](#), [n5](#), [Shift](#), [CShift](#) (5 integers, 2 reals)

(3) Generalized shift formulation up to quartic form in (hkl)
(IF [ModS](#) > 100, several optional lines are read)

([[ModS](#) = 101]: Laue class: -1)

Comment lines are directly the names of the variables as given below.
LINE 35: [D2_200](#), [D2_020](#), [D2_002](#), [D2_011](#), [D2_101](#), [D2_110](#) (6 reals)
LINE 35-1: [CD2_200](#), [CD2_020](#), [CD2_002](#), [CD2_011](#), [CD2_101](#), [CD2_110](#) (Codewords 6 reals)
LINE 35-2: [D4_400](#), [D4_040](#), [D4_004](#), [D4_220](#), [D4_202](#) (5 reals)
LINE 35-3: [CD4_400](#), [CD4_040](#), [CD4_004](#), [CD4_220](#), [CD4_202](#) (Codewords 5 reals)
LINE 35-4: [D4_022](#), [D4_211](#), [D4_121](#), [D4_112](#), [D4_301](#) (5 reals)
LINE 35-5: [CD4_022](#), [CD4_211](#), [CD4_121](#), [CD4_112](#), [CD4_301](#) (Codewords 5 reals)
LINE 35-6: [D4_301](#), [D4_130](#), [D4_103](#), [D4_013](#), [D4_031](#) (5 reals)
LINE 35-7: [CD4_301](#), [CD4_130](#), [CD4_103](#), [CD4_013](#), [CD4_031](#) (Codewords 5 reals)

([ModS]=102] Laue class: 1 2/m 1)

Comment lines are directly the names of the variables as given below.

LINE 35: D2_200, D2_020, D2_002, D2_101 (4 reals)
LINE 35-1: CD2_200, CD2_020, CD2_002, CD2_101 (Codewords 4 reals)
LINE 35-2: D4_400, D4_040, D4_004, D4_220, D4_202 (5 reals)
LINE 35-3: CD4_400, CD4_040, CD4_004, CD4_220, CD4_202 (Codewords 5 reals)
LINE 35-4: D4_022, D4_121, D4_301, D4_103 (5 reals)
LINE 35-5: CD4_022, CD4_121, CD4_301, CD4_103 (Codewords 5 reals)

([ModS]=-102] Laue class: 1 1 2/m)

Comment lines are directly the names of the variables as given below.

LINE 35: D2_200, D2_020, D2_002, D2_110 (4 reals)
LINE 35-1: CD2_200, CD2_020, CD2_002, CD2_110 (Codewords 4 reals)
LINE 35-2: D4_400, D4_040, D4_004, D4_220, D4_202 (5 reals)
LINE 35-3: CD4_400, CD4_040, CD4_004, CD4_220, CD4_202 (Codewords 5 reals)
LINE 35-4: D4_022, D4_112, D4_310, D4_130 (5 reals)
LINE 35-5: CD4_022, CD4_112, CD4_310, CD4_130 (Codewords 5 reals)

([ModS]=103] Laue class: mmm)

Comment lines are directly the names of the variables as given below.

LINE 35: D2_200, D2_020, D2_002 (4 reals)
LINE 35-1: CD2_200, CD2_020, CD2_002 (Codewords 4 reals)
LINE 35-2: D4_400, D4_040, D4_004, D4_220, D4_202, D4_022 (6 reals)
LINE 35-3: CD4_400, CD4_040, CD4_004, CD4_220, CD4_202, CD4_022 (Codewords 6 reals)

[ModS]=104, 105] Laue classes: 4/m, 4/mmm)

Comment lines are directly the names of the variables as given below.

LINE 35: D2_200, D2_002 (2 reals)
LINE 35-1: CD2_200, CD2_002 (Codewords 2 reals)
LINE 35-2: D4_400, D4_004, D4_220, D4_202 (4 reals)
LINE 35-3: CD4_400, CD4_004, CD4_220, CD4_202 (Codewords 4 reals)

[ModS]=106, 107] Laue classes: -3 R, -3m R)

Comment lines are directly the names of the variables as given below.

LINE 35: D2_200, D2_002 (2 reals)
LINE 35-1: CD2_200, CD2_002 (Codewords 2 reals)
LINE 35-2: D4_400, D4_004, D4_112, D4_211 (4 reals)
LINE 35-3: CD4_400, CD4_004, CD4_112, CD4_211 (Codewords 4 reals)

[ModS]=108, 109, 110, 111, 112] Laue classes: -3 H, -3m1, -31m, 6/m, 6/mmm)

Comment lines are directly the names of the variables as given below.

LINE 35: D2_200, D2_002 (2 reals)
LINE 35-1: CD2_200, CD2_002 (Codewords 2 reals)
LINE 35-2: D4_400, D4_004, D4_112 (3 reals)
LINE 35-3: CD4_400, CD4_004, CD4_112 (Codewords 3 reals)

[ModS]=113, 114] Laue classes: m3, m3m)

Comment lines are directly the names of the variables as given below.

LINE 35: D2_200 (1 real)
LINE 35-1: CD2_200 (Codewords 1 real)
LINE 35-2: D4_400, D4_220 (2 reals)
LINE 35-3: CD4_400, CD4_220 (Codewords 2 reals)

Shift parameters variables

Shf1, Shf2, Shf3 – Shift parameters

hkl-dependent shift parameters.

These parameters are defined by the user through the subroutines SHIFHKL, where a particular model for displacement of Bragg reflections is built for each value of **ModS**.

ModS – Select a model for shifts

Model for shifts.

[Sh1, Sh2, Sh3] - Vector defining the “shift axis”

Is the vector defining the “shift axis”. Should be the same for all patterns.

The position of the reflection is displaced according to the expressions:

$$T_S = T_B + \text{Shf1} \times \cos\phi \text{ (ModS=1)}$$

$$T_S = T_B + \text{Shf1} \times \sin\phi \text{ (ModS=-1)}$$

Where ϕ is the angle between the reciprocal vector $[hkl]^*$ and the direct vector **[Sh1, Sh2, Sh3]**

n1, n2, n3, n4, n5, SHIFT – User defined (hkl) shift rules

Set of ABS(**ModS**) (≤ 10) lines, defining rules to be satisfied by reflections undergoing shifts with respect to the theoretical Bragg position, due to some kind of defects: stacking and twin faults.

Are set of integers that should be the same for all patterns. If the Miller indices of the reflection satisfy the relation:

$$\mathbf{n1} H + \mathbf{n2} K + \mathbf{n3} L = \mathbf{n4} n + \mathbf{n5}$$

The position of the reflection is displaced according to the expressions:

$$2\theta_S = 2\theta_B + 2 \text{SHIFT} d^2 \tan\theta \times 10^{-2} \text{ (} 2\theta \text{ space)}$$

$$\text{TOF}_S = \text{TOF}_B - \text{SHIFT} d^3 \text{Dtt1} \times 10^{-2} \text{ (T.O.F. space)}$$

$$E_S = E_B - \text{SHIFT} / (2d) \text{Dtt1} \times 10^{-2} \text{ (Energy space)}$$

DN_HKL, (H+K+L=N; N=2,4) – Generalized shift

The position of the reflection is displaced according to the expressions:

$$2\theta_S = 2\theta_B + 2 S_h d^2 \tan\theta \times 10^{-2} \text{ (} 2\theta \text{ space)}$$

$$\text{TOF}_S = \text{TOF}_B - S_h d^3 \text{Dtt1} \times 10^{-2} \text{ (T.O.F. space)}$$

$$E_S = E_B - S_h / (2d) \text{Dtt1} \times 10^{-2} \text{ (Energy space)}$$

The expression used for calculating the scalar S_h for reflection \mathbf{h} is given by:

$$S_h = \sum_{\{H+K+L=2\}} D2_{HKL} h^H k^K l^L + \sum_{\{H+K+L=4\}} D4_{HKL} h^H k^K l^L$$

POWDER DATA III: ADDITIONAL ANISOTROPIC SIZE PARAMETERS

(Optional LINE 36 is read if Size-Model ≠ 0 on LINE 27)

Input format

Common Single/Multi-Pattern format:

This specific optional sub-block of the block of lines 26 to 38 (see, [loop over phases](#)) should be given for each pattern to which the phase contributes (controlled by JCONTR(n_pat) in line 19. These lines may be grouped as:

Size models:

(1) Cylindrical (platelet) shaped crystallites

[If Size-Model =±1]

Comment line: **!** Platelet-Needle vector (Size)

LINE 36: Sz1, Sz2, Sz3 (3 reals)

(2) User defined selective (hkl) size broadening due to defects

[If Size-Model <-1]

LINE 36: n1, n2, n3, n4, n5, SZ, CSZ (5 integers, 2 reals)

Comment line: **!** Size-Broadening (n1.h + n2.k + n3.l = n4 +/- n5) Size-par Code

(3) Spherical harmonics expansion of the crystallites shape

[Size-Model=15] Laue class: 2/m

LINE 36: Y00, Y22+, Y22-, Y20, Y44+, Y44- (6 reals)

LINE 36-1: CY00, CY22+, CY22-, CY20, CY44+, CY44- (Codewords 6 reals)

LINE 36-2: Y42+, Y42-, Y40 (3 reals)

LINE 36-3: CY42+, CY42-, CY40 (Codewords 3 reals)

Comment lines are directly the names of the variables as given below.

[Size-Model=16] Laue class: -3 m H

LINE 36: Y00, Y20, Y40, Y43-, Y60, Y63- (6 reals)

LINE 36-1: CY00, CY20, CY40, CY43-, CY60, CY63- (Codewords 6 reals)

LINE 36-2: Y66+ (1 real)

LINE 36-3: CY66+ (Codewords 1 real)

Comment lines are directly the names of the variables as given below.

[Size-Model=17] Laue classes: m3, m3m. Cubic harmonics, for m3m K62=0.

LINE 36: K00, K41, K61, K62, K81 (5 reals)

LINE 36-1: CK00, CK41, CK61, CK62, CK81 (Codewords 5 reals)

Comment lines are directly the names of the variables as given below.

[Size-Model=18] Laue class: mmm

LINE 36: Y00, Y20, Y22+, Y40, Y42+, Y44+ (6 reals)

LINE 36-1: CY00, CY20, CY22+, CY40, CY42+, CY44+ (Codewords 6 reals)

Comment lines are directly the names of the variables as given below.

[*Size-Model*=19] Laue classes: 6/m, 6/mmm. For 6/mmm $Y66=0$.

LINE 36: Y00, Y20, Y40, Y60, Y66+, Y66- (6 reals)
LINE 36-1: CY00, CY20, CY40, CY60, CY66+, CY66- (Codewords 6 reals)

Comment lines are directly the names of the variables as given below.

[*Size-Model*=20] Laue class: -3 H.

LINE 36: Y00, Y20, Y40, Y43-, Y43+ (5 reals)
LINE 36-1: CY00, CY20, CY40, CY43-, CY43+ (Codewords 5 reals)

Comment lines are directly the names of the variables as given below.

[*Size-Model*=21] Laue classes: 4/m, 4/mmm. For 4/mmm $Y44=0$ and $Y64=0$.

LINE 36: Y00, Y20, Y40, Y44+, Y44-, Y60 (6 reals)
LINE 36-1: CY00, CY20, CY40, CY44+, CY44-, CY60 (Codewords 6 reals)
LINE 36-2: Y64+, Y64- (2 reals)
LINE 36-3: CY64+, CY64- (Codewords 2 reals)

Comment lines are directly the names of the variables as given below.

[*Size-Model*=22] Laue class: -1.

LINE 36: Y00, Y20, Y21+, Y21-, Y22+, Y22- (6 reals)
LINE 36-1: CY00, CY20, CY21+, CY21-, CY22+, CY22- (Codewords 6 reals)

Comment lines are directly the names of the variables as given below.

Variables

[**Sz1, Sz2, Sz3**] - Vector defining the platelets

Is the vector defining the platelets. Must be the same for all patterns.

n1, n2, n3, n4, n5, SZ – User-defined rules for selective hkl size-broadening

Set of ABS(**Size-Model**) (≤ 9) lines, defining rules to be satisfied by reflections undergoing selective “size-like” broadening due to some kind of defects (anti-phase domains, ...).

Are set of integers that should be the same for all patterns. If the Miller indices of the reflection satisfy the relation:

$$\mathbf{n1} H + \mathbf{n2} K + \mathbf{n3} L = \mathbf{n4} n + \mathbf{n5}$$

The Lorentzian broadening of the reflection is given by the expressions:

$$\begin{aligned} H_L &= 0.360 \mathbf{SZ} \lambda \cos\theta / \pi^2 && (\text{2}\theta \text{ space}) \\ H_L &= (2/\pi) \mathbf{SZ} d^5 \mathbf{Dtt1} \times 10^{-3} && (\text{T.O.F. space}) \\ H_L &= \mathbf{SZ}/(2d) \mathbf{Dtt1} \times 10^{-3} && (\text{Energy space}) \end{aligned}$$

Clm±, Klm - SH coefficients

The Lorentzian broadening is calculated as previously, but for each reflexion (hkl) with spherical coordinates (h, θ , φ) in the reciprocal space, SZ is calculated by the development onto the

Spherical Harmonics basis functions as: $SZ(\theta, \varphi) = \sum_{lm\pm} C_{lm\pm} Y_{lm}^{\pm}(\theta, \varphi)$

POWDER DATA IV- ADDITIONAL ANISOTROPIC STRAIN AND SIZE PARAMETERS

(Optional LINE 37-38 are read if $Str \neq 0$ on [line 19](#), or if $Strain-Model \geq 7$ and $Str=0$)

Input format

Common Single/Multi-Pattern format:

This specific optional sub-block of the block of lines 26 to 38 (see, [loop over phases](#)) should be given for some specific strain models (controlled through [Str](#) ([LINE 19](#)) and [Strain-Model](#) ([LINE 26](#))), and for each pattern to which the phase contributes (controlled by [JCONTR](#)(*n_pat*) in [line 19](#). These lines may be grouped as:

Size/strain models:

(1) *Axial vector Microstrain*
[If [Strain-Model](#)=7]

Comment line! Axial vector Microstrain

LINE 37: [St1](#), [St2](#), [St3](#) (3 reals)

(2) *Other microstrain models*
[If [Strain-Model](#)>8 and [Str](#)=0]

Comment line! 5 additional strain parameters ($StrainModel > 8$)

LINE 37: [STR4](#), [STR5](#), [STR6](#), [STR7](#), [STR8](#) (5 reals)

LINE 37-1: [CSTR4](#), [CSTR5](#), [CSTR6](#), [CSTR7](#), [CSTR8](#) (Codewords 5 reals)

(3) *Generalized size model*
[If [Str](#)=-1, 2, 3]

Comment line! Generalised size parameters (quadratic form):

LINE 37: [SZ1](#), [SZ2](#), [SZ3](#), [SZ4](#), [SZ5](#), [SZ6](#) (6 reals)

LINE 37-1: [CSZ1](#), [CSZ2](#), [CSZ3](#), [CSZ4](#), [CSZ5](#), [CSZ6](#) (Codewords 6 reals)

(4) *Anisotropic strain broadening*
modelled using a quartic form in reciprocal space
[If $ABS(Str)=1$ and $Strain-Model \neq 0$]

[Strain-Model](#) =1 Laue class: -1

Comment lines are directly the names of the variables as given below.

LINE 37: [S_400](#), [S_040](#), [S_004](#), [S_220](#), [S_202](#) (5 reals)

LINE 37-1: [CS_400](#), [CS_040](#), [CS_004](#), [CS_220](#), [CS_202](#) (Codewords 5 reals)

LINE 37-2: [S_022](#), [S_211](#), [S_121](#), [S_112](#), [S_301](#) (5 reals)

LINE 37-3: [CS_022](#), [CS_211](#), [CS_121](#), [CS_112](#), [CS_301](#) (Codewords 5 reals)

LINE 37-4: [S_301](#), [S_130](#), [S_103](#), [S_013](#), [S_031](#) (5 reals)

LINE 37-5: [CS_301](#), [CS_130](#), [CS_103](#), [CS_013](#), [CS_031](#) (Codewords 5 reals)

[Strain-Model](#)=2 Laue class: 1 2/m 1

Comment lines are directly the names of the variables as given below.

LINE 37: [S_400](#), [S_040](#), [S_004](#), [S_220](#), [S_202](#) (5 reals)

LINE 37-1: [CS_400](#), [CS_040](#), [CS_004](#), [CS_220](#), [CS_202](#) (Codewords 5 reals)

LINE 37-2: [S_022](#), [S_121](#), [S_301](#), [S_103](#) (4 reals)

LINE 37-3: [CS_022](#), [CS_121](#), [CS_301](#), [CS_103](#) (Codewords 4 reals)

Strain-Model = -2 *Laue class: 1 1 2/m*

Comment lines are directly the names of the variables as given below.

LINE 37: S_400, S_040, S_004, S_220, S_202 (5 reals)
LINE 37-1: CS_400, CS_040, CS_004, CS_220, CS_202 (Codewords 5 reals)
LINE 37-2: S_022, S_112, S_310, S_130 (4 reals)
LINE 37-3: CS_022, CS_112, CS_310, CS_130 (Codewords 4 reals)

Strain-Model = 3 *Laue class: mmm*

Comment lines are directly the names of the variables as given below.

LINE 37: S_400, S_040, S_004, S_220, S_202, S_022 (6 reals)
LINE 37-1: CS_400, CS_040, CS_004, CS_220, CS_202, CS_022 (Codewords 6 reals)

Strain-Model = 4, 5 *Laue classes: 4/m, 4/mmm*

Comment lines are directly the names of the variables as given below.

LINE 37: S_400, S_004, S_220, S_202 (4 reals)
LINE 37-1: CS_400, CS_004, CS_220, CS_202 (Codewords 4 reals)

Strain-Model = 6, 7 *Laue classes: -3 R, -3m R*

Comment lines are directly the names of the variables as given below.

LINE 37: S_400, S_004, S_112, S_211 (4 reals)
LINE 37-1: CS_400, CS_004, CS_112, CS_211 (Codewords 4 reals)

Strain-Model = 8, 9, 10, 11, 12 *Laue classes: -3, -3m1, -31m, 6/m, 6/mmm*

Comment lines are directly the names of the variables as given below.

LINE 37: S_400, S_004, S_112 (3 reals)
LINE 37-1: CS_400, CS_004, CS_112 (Codewords 3 reals)

Strain-Model = 13, 14 *Laue classes: m3, m3m*

Comment lines are directly the names of the variables as given below.

LINE 37: S_400, S_220 (2 reals)
LINE 37-1: CS_400, CS_220 (Codewords 2 reals)

(5) Generalized strain model

[Str=1 and Strain-Model=0, or Str=3]

Comment lines are directly the names of the variables as given below.

LINE 37: STR4, STR5, STR6, STR7, STR8, STR9 (6 reals)
LINE 37-1: CSTR4, CSTR5, CSTR6, CSTR7, CSTR8, CSTR9 (Codewords 6 reals)
LINE 37-2: STR10, STR11, STR12, STR13, STR14, STR15 (6 reals)
LINE 37-3: CSTR10, CSTR11, CSTR12, CSTR13, CSTR14, CSTR15 (Codewords 6 reals)

Lorentzian strain broadening

(if $N_{pr} \geq 7$ and $Strain-Model \neq 0$ or $Str=1$ or $Str=3$.)

Comment line: ! Lorentzian strain coeff.+ code

LINE 38: XI, CXI (2 reals, one parameter and codeword)

Variables

[St1, St2, St3] - Axial microstrain vector

Is the vector defining the axial microstrain.

STR4, STR5, STR6, STR7, STR8

STR9, STR10, STR11, STR12, STR13 – Additional strain parameters

Additional strain parameters. If Str=0 Their meaning depend on the particular value of Strain-Model. If Str =1 and Strain-Model = 0, or Str = 3, the **STR4** to **STR15** parameters correspond to another (generalized) formulation of the strain parameters. See Mathematical section for explanation.

SZ1, SZ2, SZ3, SZ4, SZ5, SZ6 – Generalized size parameters

Coefficients of the quadratic form:

$$DSIZ = k_s \times d^2 \times (SZ1 h^2 + SZ2 k^2 + SZ3 l^2 + SZ4 kl + SZ5 hl + SZ6 hk)$$

Where k_s is defined as:

$$\begin{aligned} k_s &= 360/\pi^2 \times \lambda \cdot 10^{-3} && (2\theta \text{ space}) \\ k_s &= 2/\pi \times Dtt1 \cdot 10^{-3} && (\text{TOF and Energy space}) \end{aligned}$$

The FWHM of the Lorentzian size broadening is given by the expression:

$$\begin{aligned} H_L &= DSIZ/\cos\theta && (2\theta \text{ space}) \\ H_L &= DSIZ d^2 && (\text{TOF space}) \end{aligned}$$

Codewords of the size parameters (quadratic form).

S HKL – Coefficient of the quartic form for strains

The following items correspond to anisotropic strain broadening modelled using a quartic form in reciprocal space. This correspond to an interpretation of the strains as due to static fluctuations and correlations between metric parameters [see J. Rodríguez-Carvajal, M.T. Fernandez-Díaz and J.L. Martínez, *J. Phys: Condensed Matter* **3**, 3215 (1991)]. The notation used is that of P. Stephens [Peter W. Stephens, *J. Appl. Cryst.* **32**, 281 (1999)].

$$DST = \sum_{\{H+K+L=4\}} S_{HKL} h^H k^K l^L$$

A maximum of 15 parameters can be given. The input format is different for the different Laue classes.

XI – Lorentzian anisotropic strain parameter

Lorentzian weigh of the strained broadened profile, if modelized by a pseudo Voigt function (Npr>7)

User defined parameters for the phase

(LINE 43, read only if Furth ≠0)

Input format

LINE 43: NAMEPAR, VALUEPAR, CODEPAR (Furth Lines)

Comment line:!
Futher parameters

Variables

NAMEPAR – User-defined parameter name

Character A4. Name of the parameter given by the user.

VALUEPAR – User-defined parameter value

Value of the parameter

CODEPAR – User-defined parameter codeword

Codeword of the parameter

Propagation vectors

(LINE 44 is read if Nvk≠0)

Input format

(ABS(Nvk) pairs of Lines)

Comment line:!
Propagation vectors

LINE 44: PVK (3 reals)

LINE 44-1: CPVK (3 reals, codewords)

Variables

PVKX, PVKY, PVKZ – Propagation vector components

Components of the propagation vector **k** in reciprocal lattice units.

CPVKX, CPVKY, CPVKZ

Codewords of the component of **k**.

Soft distance constraints

(LINE 45, read if Dis≠0)

Input format

Comment line: **! Soft distance constraints:**

LINE 45: CATOD1, CATOD2, ITnum, T1, T2, T3, Dist, Sigma (2A4, 1 integer, 5 reals)(NDIC Lines)

Variables

CATOD1, CATOD2 – Pairs of constrained atoms

Names of the atoms to be constrained. They must coincide with labels (Atom in LINE 25) in the asymmetric unit (see Atom in LINE 25).

ITnum – Number selecting a symmetry operator, hence a representative atom in the orbit of CATOD2

Integer for selecting the rotation part of the symmetry operator to be applied to the coordinates of the atom CATOD2 given as representative in the asymmetric unit.

(T1, T2, T3) - Translation part of the symmetry operator

Translation part of the above symmetry operator

Dist

Value of the required distance in angstroms (Å).

Sigma - Standard deviation of the distance

Standard deviation of the distance.

The numbering of symmetry operators to be given in distance constraints conditions correspond to the listed values of ITnum in the tables below. If combination with a centre of symmetry is needed the value must be entered as negative.

Non-hexagonal frames

<u>ITnum</u>	Symmetry symbol	Rotation matrix
(1)	1	--> (x, y, z)
(2)	2 (0, 0, z)	--> (-x, -y, z)
(3)	2 (0, y, 0)	--> (-x, y, -z)
(4)	2 (x, 0, 0)	--> (x, -y, -z)
(5)	3+ (x, x, x)	--> (z, x, y)
(6)	3+ (-x, x, -x)	--> (z, -x, -y)
(7)	3+ (x, -x, -x)	--> (-z, -x, y)
(8)	3+ (-x, -x, x)	--> (-z, x, -y)
(9)	3- (x, x, x)	--> (y, z, x)
(10)	3- (x, -x, -x)	--> (-y, z, -x)
(11)	3- (-x, -x, x)	--> (y, -z, -x)
(12)	3- (-x, x, -x)	--> (-y, -z, x)
(13)	2 (x, x, 0)	--> (y, x, -z)
(14)	2 (x, -x, 0)	--> (-y, -x, -z)
(15)	4- (0, 0, z)	--> (y, -x, z)
(16)	4+ (0, 0, z)	--> (-y, x, z)
(17)	4- (x, 0, 0)	--> (x, z, -y)

```

(18) 2 ( 0, y, y) --> (-x, z, y)
(19) 2 ( 0, y, -y) --> (-x, -z, -y)
(20) 4+ ( x, 0, 0) --> ( x, -z, y)
(21) 4+ ( 0, y, 0) --> ( z, y, -x)
(22) 2 ( x, 0, x) --> ( z, -y, x)
(23) 4- ( 0, y, 0) --> (-z, y, x)
(24) 2 (-x, 0, x) --> (-z, -y, -x)

```

Hexagonal frames

<u>Itnum</u>	Symmetry symbol	Rotation matrix
(25)	1	--> (x , y, z)
(26)	3+ (0, 0, z)	--> (-y, x-y, z)
(27)	3- (0, 0, z)	--> (-x+y, -x , z)
(28)	2 (0, 0, z)	--> (-x , -y, z)
(29)	6- (0, 0, z)	--> (y, -x+y, z)
(30)	6+ (0, 0, z)	--> (x-y, x , z)
(31)	2 (x, x, 0)	--> (y, x , -z)
(32)	2 (x, 0, 0)	--> (x-y, -y, -z)
(33)	2 (0, y, 0)	--> (-x , -x+y, -z)
(34)	2 (x, -x, 0)	--> (-y, -x , -z)
(35)	2 (x, 2x, 0)	--> (-x+y, y, -z)
(36)	2 (2x, x, 0)	--> (x , x-y, -z)

The string representing a distance soft constraint may be obtained after a run with fixed parameters asking for output a distance file CODFILn.dis, another file called "dconstrn.hlp" is automatically generated. After editing this file, the user may paste into the file CODFIL.pcr the desired strings and modify them as wished.

Soft angle constraints

(LINE 45-1: read if Mom(Angles)≠0)

Input format

Comment line: **! Soft angle constraints:**

LINE 45-1: CATOD1, CATOD2, CATOD3, ITnum1, ITnum2, T1, T2, T3, t1, t2, t3, Angl,
Sigma
(3A4, 2 integer, 8 reals) (Mom(Angles) Lines) **[If]**

Variables

CATOD1, CATOD2, CATOD3 – Name of the constrained atoms

Names of the atoms to be constrained. They must coincide with labels (Atom in LINE 25) in the asymmetric unit (see Atom in LINE 25).

The central atom for angle constraints is the atom CATOD2.

ITnum1 - Number selecting a symmetry operator, hence a representative atom in the orbit of CATOD2

Integer for selecting the rotation part of the symmetry operator to be applied to the coordinates of the atom CATOD2 given as representative in the asymmetric unit.

ITnum2 - Number to select a symmetry operator, hence a representative atom in the orbit of CATOD3

Integer for selecting the rotation part of the symmetry operator to be applied to the coordinates of the atom CATOD3 given as representative in the asymmetric unit.

(T1, T2, T3) - Translation part of the symmetry operator ITnum1

Translation part of the symmetry operator indexed by ITnum1.

(t1, t2, t3) - Translation part of the symmetry operator ITnum2

Translation part of the symmetry operator indexed by ITnum2.

Angl – Angle constraint

Value of the required angle in degrees.

Sigma - Angle Sdev

Standard deviation of the angle.

As in the case of distance constraints, the string representing an angle soft constraint may be obtained from the file “dconstrn.hlp” (remember to put ANG MAX≠0, see LINE 20). After editing this file, the user may paste into the file CODFIL.pcr the desired strings and modify them as wished.

Soft moment constraints

(LINE 46: read if Mom(Moment)≠0)

Input format

LINE 46: CATOM, Moment, Sigma (A2, 2 reals) (Mom(Moment) Lines)

Comment line:!
Soft moments constrains

Variables

CATOM - Name of the constrained magnetic atom

Two letters equal to the two first character of the Atom of atoms in asymmetric unit which are constrained.

Moment – Moment value

Value of the required magnetic moment.

Sigma – Moment Sdev

Standard deviation of Moment. It does not work with incommensurate magnetic structures.

[END LOOP OVER PHASES]

Goto start of the loop

Hard limits for parameters

(LINE 47: read if Nre≠0 on Line 4)

Input format

Comment line: ! Limits for selected parameters

or

Comment line: ! Limits for selected parameters (+ steps & BoundCond for SA) [If Cry=3]

LINE 47: NUMPAR, LowLIMIT, HighLIMIT, Step, IBOUND, NAMPAR

(1 integer, 2 reals, 1 integer, character)

(Nre Lines)

Variables

NUMPAR - “number” of the parameter

Is the “number” of the parameter to be constrained within the limits specified by the interval [LowLIMIT, HighLIMIT] (NUMPAR=INT(CODEWORD/10)). For a proper use of this option one has to put limits to the variable appearing for the first time (from top to bottom in the CODFIL.pcr file) with the wished parameter code, this must have a positive sign and a unit multiplier; only then the constraints established by using the same codeword will work.

LowLIMIT - Smallest admissible value

Smallest admissible value of the parameter number NUMPAR.

HighLIMIT - Highest admissible value

Highest admissible value of the parameter number NUMPAR.

Step – If simulated Annealing job...

Only used if Cry = 3 (Simulated Annealing job). Value of the maximum allowed variation of the parameter number NUMPAR in order to generate a new configuration. A configuration means a set of Nre values of the selected parameters within the box defined by the limits. The change in a single parameter generates a new configuration. If PAR_{old} is the current value of the parameter, the new configuration is generated by modifying the value of the parameter as PAR_{new} = PAR_{old} + Step (2 × RND–1.0), where RND is a random number between 0 and 1.

IBOUND - Boundary conditions type

Integer to select the type of boundary conditions for the parameter number NUMPAR.

If IBOUND=0 the boundaries are treated as hard, e.g. if PAR_{new} exceed the higher limit, then the value is set to the limit: PAR_{new} = HighLIMIT.

If IBOUND=1 the boundaries are treated as periodic, e.g. if PAR_{new} exceed the higher limit, then the value is calculated as: PAR_{new} (corrected) = PAR_{new} – (HighLIMIT–LowLIMIT).

NAMPAR – User given name for the selected parameter

Name given by the user to the parameter number NUMPAR.

Monte-Carlo search parameters

(LINE 48, read if `If Cry=2`)

Input format

LINE 48: `NCONF, NSOLU, NREFLEX, NSCALEF` (4 integers)
Comment line: `! Nconf Nsolu Num_Ref Nscalef`

Variables

NCONF, NSOLU, NREFLEX, NSCALE – Monte-carlo parameters

This option is a crude Montecarlo way of searching a starting configuration suitable for further refinement using LSQ. A configuration means a set of **Nre** values of the selected parameters within the box defined by the limits in [LINE 47](#), here the **Step** value is not used. The program tries at random **NCONF** configurations and select the best **NSOLU** solutions (lower R-factors) using the first **NREFLEX** reflections of the file CODFILn.hkl or CODFIL.int. If **NSCALEF** is different from zero, then the scale factor used in the program is obtained from the relation:

$$\sum I_{obs} = Scale \sum I_{calc}$$

Note: The constraints established with the coding of parameters have the same meaning as with least-squares(LS). In LSQ refinement, or in Montecarlo search, two variables having codes `xx1.00` and `xx0.5` means that the shift applied to the second variable is half the shift applied to the first one irrespective of their initial values. Remember the prescription given in [LINE 47](#).

Simulate annealing parameters

(LINE 49, read if `Cry =3`)

Input format

Comment line: `! T_ini Anneal Accept NumTemps NumThCyc InitConf`
LINE 49: `T_INI, ANNEAL, ACCEPT, NUMTEMPS, NUMTHCYC, INITCONF`
(3 reals - 3 integers)
Comment line: `! NCyclm Nsolu Num_Ref Nscalef NAlgor`
LINE 49-1: `NCYCLM, NSOLU, NREFLEX, NSCALEF, NALGOR` (5 integers)
Comment line: `! ISwap Var-Real/Imag`
LINE 49-2: `ISWAP, MCOMPL` (2 integers)
(Optional line, read if `Isy=-2`)
Comment line: `! Coefficients for Real/Imag switch`
LINE 49-3: `(CPL(i), i=1,MCOMPL)` (MCOMPL integers)

Variables

The parameters given in the line 49 and the next two lines 49-1 and 49-2 are specific to the Simulated Annealing algorithm for optimising a crystal and/or magnetic structure using integrated intensities. This method may be used for *solving* crystal structures when more efficient methods (like direct methods) fail. It may be used always for magnetic structure determination when simple trial an error (even using symmetry analysis) fail.

T INI - Starting temperature

Starting temperature in the Simulated Annealing procedure. Arbitrary value, the user should experiment with different values for each problem. A good starting point is a temperature for which the percentage of accepted configurations in the first Montecarlo cycle is of the order of 80-90%.

ANNEAL – Reduction factor of the temperature between the MC cycles

Value of the constant needed for reducing the temperature for the next Montecarlo cycle. Normally 0.9 is a good value.

ACCEPT - Lowest percentage of accepted configurations

Lowest percentage of accepted configurations to consider the algorithm has converged. In case of using NALGOR=0,1 (see below) the meaning of ACCEPT is the value of the lowest admissible average step, i.e. the program stops if $\langle \text{Step} \rangle \leq \text{ACCEPT}$.

NUMTEMPS - Maximum number of temperatures

Maximum number of temperatures.

The program stops after running the algorithm for NUMTEMPS temperatures irrespective of the convergence criterion based on ACCEPT.

NUMTHCYC - Number of Montecarlo cycles

Number of Montecarlo cycles to be excluded from the calculation of averages. Normally 0.

INITCONF – Switch to random or user defined initial configuration

Flag for select the treatment of the initial configuration. If INITCONF=0 the initial configuration is totally random. If INITCONF=1, the initial configuration is the one given by the values of the parameters in the PCR-file. This last option is useful when one tries to optimise an already good starting configuration, by controlling the box limits and the steps.

NCYCLM - Number of Montecarlo cycles per temperature

Number of Montecarlo cycles per temperature. A Montecarlo cycle corresponds to the change of all the free parameters one by one. NCYCLM must be at least 10 to 20 (or more!) times the number of free parameters. This is crucial for the success of the algorithm, however the user should experiment with this number. Remember that run time increases proportional to the value of NREFLEX × NCYCLM × NUMTEMPS × Nre.

NSOLU - Number of solutions to be stored

Number of solutions to be stored. Due to the nature of the Simulated Annealing algorithm this number should be fixed to 1.

NREFLEX – Number of used reflexions

The program uses the first NREFLEX reflections of the file CODFILn.hkl or CODFIL.int

NSCALEF -

If different from zero the scale factor used in the program is obtained from the relation:

$$\sum I_{obs} = Scale \sum I_{calc}$$

NALGOR - Flag to select the type of algorithm

Flag to select the type of algorithm to be applied.

=0 The Corana algorithm is selected. This algorithm do not use fixed steps for moving the parameters defining the configuration, instead the program starts by using then whole interval as initial step for all parameters and then adapt progressively their values in order to maintain an approximate rate of accepted configurations between two pre-set values (40% and 60% in our case).

- =1 The Corana algorithm is selected using as initial steps, instead of the whole interval, the steps given in [LINE 47](#).
- =2 The conventional algorithm is selected using the fixed steps as given in [LINE 47](#).

ISWAP - Flag to allow interchange of atoms

Flag to allow interchange of atoms in the Simulated Annealing algorithm. If ISWAP≠0 the program enter into a procedure of interchanging pairs of atoms each ISWAP Montecarlo cycles. The new configurations are accepted only if the cost function is reduced. For magnetic structures this flag has to be fixed to zero.

MCOMPL – Controls of the number of “switchable” coefficients of BSF

Number of coefficients C_i to be switched between real or pure imaginary in magnetic structure determination when the irreducible representations of the propagation vector group are used.

CPL(i) - Flags to indicate which BSF coefficients can be switched between real and pure imaginary value

Flags to indicate if the coefficient C_i (see [LINE](#)) will be switched, CPL(i)=1, or will remain fixed, CPL(i)=0, in the Simulated Annealing algorithm.

Line printer plot

([LINE 50](#), read if Ppl =1)

Input format

For each pattern, in the order they are given in [LINE 5](#), a line should be given depending on Ppl(n_{patt}):

Comment line: ! IScale IDif → Pattern #N
 LINE 50 : ISCALE, IDIF (2 integers)

Variables

ISCALE

Counts per character position for observed and calculated curves on line printer plot

IDIF

Counts per character for difference curve

Output reflexion list

([LINE 51](#), read if Rpa =2)

Input format

Comment line: ! 2Th1 2Th2
 LINE 51: THET1, THET2 (2 reals)

The reflection list between these angles is saved in the file CODFIL.sav

Other Input/Output files

CODFIL.shp or global.shp

File providing a numerical table for calculating the peak shape and its derivative. This option is useful only in very special cases where the peak shape characteristic of a given instrument is not possible to be fitted with one of the peak shape analytical functions provided by **FullProf**.

The peak shape should be given in a normalised form $P(x)$ where the variable $x \in [x_1, x_2]$ is chosen to give a FWHM=1 and the area is 1, so $\int_{x_1}^{x_2} P(x)dx = 1$.

That allows the use of the conventional U,V, and W parameters for defining the FWHM as a function of angle.

The format of this file is the following:

- Line1: Any comment
- Line2: $Np\delta, nupr, 2\theta_1, 2\theta_2, \dots, 2\theta_{nupr}$
 - $Np\delta$: Number of points
 - $nupr$: Number of different profiles
 - $2\theta_j$: Angle to which profile j is best adapted

The rest of the lines are columns with $x, P_1(x), P_1'(x), P_2(x), P_2'(x), \dots, P_{nupr}(x), P_{nupr}'(x)$ in free format. $P_j'(x)$ is the derivative of $P_j(x)$ with respect to x . The profile of a reflection situated between $2\theta_j$ and $2\theta_{j+1}$ is linearly interpolated between the profiles $P_j(x)$ and $P_{j+1}(x)$.

CODFIL.cor / cor.cor

The format of this file depends on the value of the input variable **Cor**.

Cor=1

The file CODFIL.cor (or cor.cor) starts with a comment and follows with a list of pairs: a simple list of scattering variable and correction values.

<i>TITLE ...</i>	
<i>Scattering variable (T)</i>	<i>Value of the correction</i>
"	"
.....

Data are read in free format. For peaks between points provided in the CODFIL.cor file, the correction is linearly interpolated.

Example:

```
This is my correction FILE for integrated intensities
10.0      1.3
20.0      1.1
30.0      1.0
40.0      0.9
80.0      0.8
```


$$H_L = (X_i 2\theta + Y_i) 2\theta + Z_i$$

Res=3 $H_G^2 = (U_i 2\theta + V_i) 2\theta + W_i$

$$H_L = (X_i 2\theta + Y_i) 2\theta + Z_i$$

Res=4 List of values 2θ , $H_G(2\theta)$, $H_L(2\theta)$

(a linear interpolation is applied for intermediate 2θ)

The first line is considered as a title

For **Res**=1,2,3 the 12 parameters U_i, V_i, W_i, X_i, Y_i and Z_i are read from lines 2 and 3 (see the above line for the available instrumental functions).

Example:

```
Resolution function of MyXrayDiffractometer
0.00802 -0.00936 0.01024 0.0029 0.0 0.0 ! U1,V1...
0.00774 -0.00552 0.00814 0.0000 0.0 0.0 ! U2,V2...
```

For **Res**=4, the file **FILERES** starts with a line containing the title followed by a line with the number of points (**NPOINS**) where the instrumental Gaussian and Lorentzian FWHM are given. **NPOINS** lines follow containing the three items: 2θ , H_G and H_L . The Bragg peaks of the diffraction pattern must be between $2\theta(1)$ and $2\theta(\text{NPOINS})$ For this case the same resolution function is applied to both wavelengths. The maximum number of **NPOINS** is 60.

FILE.dat

The format of this file, containing the input profile intensity data, depends on the value of the variable **Ins**. The different formats are described here in more detail than was described in the explanation of the **Ins** variable within the PCR-file.

When adequate a small piece of Fortran code, without specifying tests and writings, is provided in order to help users understand how the program reads the data. The module containing the procedures for reading profile intensity files in **FullProf** is available on request to the author. The important variables are the following: the variable `n_pat` indexes the pattern; **Thmin**, **Step** and **Thmax** correspond to the initial value of the scattering variable (2θ , TOF, Energy); `npts` is the number of channels in the diffraction pattern; `y` is the profile intensity; `var` is the variance of the intensities; `txv` contains the values of the scattering variable for each channel.

Ins=0

Data supplied in free format. Up to seven comment lines are accepted. The first three real numbers found at the beginning of a line are interpreted as T_i , *step* and T_f and. The following lines after T_i , *step* and T_f must contain $NPTS=(T_f-T_i)/step+1$ values of the intensity profile. Data format of TOF raw data from Argonne are also interpreted by this value of **Ins**.

The following piece of Fortran is used for reading intensity data for **Ins**=0:

```
line_da=0
DO
  line_da=line_da+1
  IF(line_da > 7) GO TO 99999
  READ(i_dat, '(a)', END=99998) aline
  IF(aline(1:4) == 'BANK') THEN
    READ(aline(5:41), *) inum, npts(n_pat)
    READ(aline(47:90), *) thmin(n_pat), step(n_pat)
    thmax(n_pat)=thmin(n_pat)+(npts(n_pat)-1)*step(n_pat)
```

```

ELSE
  READ(aline,*,iostat=ier)thmin(n_pat),step(n_pat),thmax(n_pat)
  if(ier /= 0) cycle
  npts(n_pat) = (thmax(n_pat)-thmin(n_pat))/step(n_pat)+1.5
END IF
IF(npts(n_pat) < 4) THEN
  IF(line_da < 7) CYCLE
  GO TO 99999
END IF
END DO
READ(i_dat,*,END=99998,ERR=99999)(y(i,n_pat),i=1,npts(n_pat))

```

Ins =1

D1A/D2B format (original Rietveld-Hewat format: the first line must be T_i , $step$ and T_f)
The following piece of Fortran is used in *FullProf* for reading intensity data for Ins =1:

```

READ(i_dat,*,END=99998,ERR=99999)thmin(n_pat),step(n_pat),thmax(n_pat)
npts(n_pat) = (thmax(n_pat)-thmin(n_pat))/step(n_pat)+1.5
READ(i_dat,3,END=99998,ERR=99999)(iww(i,n_pat),y(i,n_pat),i=1,npts(n_pat))
DO i=1,npts(n_pat)
  IF (y(i,n_pat) <= 0.00001) y(i,n_pat) = 1.0
  IF (iww(i,n_pat) == 0) iww(i,n_pat) = 1
  var(i,n_pat) = y(i,n_pat)/REAL(iww(i,n_pat))
  txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
END DO

```

Ins =2

D1B old ILL format (DEC-10). The code for reading this format is:

```

READ(i_dat,5,END=99998,ERR=99999)date
READ(i_dat,5,END=99998,ERR=99999)txt1
READ(i_dat,*,END=99998,ERR=99999)io,numor(n_pat)
tsamp(n_pat) = REAL(numor(n_pat))
inum = 1
READ(i_dat,5,END=99998,ERR=99999)txt2
READ(i_dat,5,END=99998,ERR=99999)txt3
j = 0
READ(i_dat,'(f7.3,10f6.0)',END=99998,ERR=99999) a1,(y(j+no,n_pat),no=1,10)
j = j+10
DO i=2,40
  READ(i_dat,'(f7.3,10f6.0)',END=99998,ERR=99999)(y(j+no,n_pat),no=1,10)
  j = j+10
END DO
y(1,n_pat) = y(2,n_pat)
y(400,n_pat) = y(399,n_pat)
thmin(n_pat) = 2.0*a1
step(n_pat) = 0.2
npts(n_pat) = 400
thmax(n_pat) = thmin(n_pat)+399*step(n_pat)
DO i=1,npts(n_pat)
  var(i,n_pat) = y(i,n_pat)
  txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
END DO

```

Ins =3

Format corresponding to the ILL instruments D1B and D20.
The code for reading the files corresponding to these instruments is:

```

READ(i_dat,11,END=99998,ERR=99999)nset

```

```

READ(i_dat, '(1x,a,2X,a,2X,a)',END=99998,ERR=99999) jdate, noexp, txt1
READ(i_dat, '(i3,1X,i6,1X,i4)',END=99998,ERR=99999) nset1, numor(n_pat), nsumad
READ(i_dat, '(f13.0,f9.0,1X,10(f8.3,1X))',END=99998,ERR=99999) &
    xm, ti, x0, ome, chi, phi, tr1, tr2, step(n_pat), tset, tcons, tsamp(n_pat)
thmin(n_pat) = x0
READ(i_dat, '(i4,a)',ERR=99999) npts(n_pat), txt1
j = 0
npunt = nint(REAL(npts(n_pat))/10.)
thmax(n_pat) = thmin(n_pat)+(npts(n_pat)-1)*step(n_pat)
DO i=1,npunt
    READ(i_dat,19,END=99998,ERR=99999) (iww(j+no,n_pat),y(j+no,n_pat),no=1,10)
    j = j+10
END DO
READ(i_dat,20,END=99998,ERR=99999) inum
DO i=1,npts(n_pat)
    IF (y(i,n_pat) <= 0.00001) y(i,n_pat) = 1.0
    IF (iww(i,n_pat) <= 0) iww(i,n_pat) = 1
    var(i,n_pat) = y(i,n_pat)/REAL(iww(i,n_pat))
    txv(i,n_pat) = thmin(n_pat)+(i-1)*step(n_pat)
END DO

```

Ins =±4

Brookhaven synchrotron.

4: First line: $2\theta_i$, step, $2\theta_f$ (free format). Rest of file: pairs of lines with 10 items like

$Y1 Y2 \dots Y10$ -- (10F8) intensities

$S1 S2 \dots S10$ -- " standard deviations

-4: Format given by DBWS program for synchrotron data. (Version DBW3.2S-8711)

The Fortran code for Ins = 4 is:

```

READ(i_dat,*,END=99998,ERR=99999) thmin(n_pat), step(n_pat), thmax(n_pat)
npts(n_pat) = (thmax(n_pat)-thmin(n_pat))/step(n_pat)+1.5
nlines = npts(n_pat)/10+1
j = 0
DO i=1,nlines
    READ(i_dat,23,END=99998,ERR=99999) (y(j+no,n_pat),no=1,10)
    READ(i_dat,24,END=99998,ERR=99999) (var(j+no,n_pat),no=1,10)
    j = j+10
END DO
var(1,n_pat) = var(1,n_pat)**2
txv(1,n_pat)=thmin(n_pat)
DO i=2,npts(n_pat)
    IF ( y(i,n_pat) < 0.00001) y(i,n_pat) = y(i-1,n_pat)
    IF (var(i,n_pat) < 0.00001) var(i,n_pat) = 1.0
    var(i,n_pat) = var(i,n_pat)**2
    cnorm(n_pat)=cnorm(n_pat)+var(i,n_pat)/y(i,n_pat)
    txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
END DO

```

Ins =5

Data from GENERAL FORMAT for TWO AXIS instrument. Three lines of text followed by two lines with the items:

$NPTS$, T_{Sample} , T_{regul} , I_{vari} , R_{mon1} , R_{mon2}

T_i , step, T_f

Set of lines containing 10 items corresponding to the Intensities in format 10F8.1, up to $NPTS$ points ($NPTS=(T_f-T_i)/step+1$), followed by the corresponding standard deviations in format (10f8.2) if $I_{vari}=1$. If $I_{vari}=0$ the standard deviations are calculated as

$$\sigma(y) = \sqrt{y \times \frac{R_{mon1}}{R_{mon2}}}$$

The following Fortran code read the data corresponding to this format:

```

READ(i_dat,'(A)',ERR=99999)txt1
READ(i_dat,'(A)',ERR=99999)txt2
READ(i_dat,'(A)',ERR=99999)txt3
READ(i_dat,'(I6,1X,2F10.3,i5,2f10.1)',ERR=99999) &
  npts(n_pat),tsamp(n_pat),treg,ivari,rmon1,rmon2
READ(i_dat,'(3F10.0)',ERR=99999)thmin(n_pat),step(n_pat),thmax(n_pat)
READ(i_dat,'(10F8.1)',ERR=99999)(y(i,n_pat),i=1,npts(n_pat))
READ(i_dat,*,ERR=99999)(y(i,n_pat),i=1,npts(n_pat))
IF(ivari /= 0) THEN      !IVARI
  READ(i_dat,'(10F8.2)',ERR=99999)(var(i,n_pat),i=1,npts(n_pat))
  READ(i_dat,*,ERR=99999)(var(i,n_pat),i=1,npts(n_pat))
  cnorm(n_pat)=0.0
  DO i=1,npts(n_pat)
    IF(y(i,n_pat) < 0.0001) y(i,n_pat)=0.0001
    var(i,n_pat)=var(i,n_pat)*var(i,n_pat)
    IF(var(i,n_pat) < 0.000001) var(i,n_pat)=1.0
    txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
    cnorm(n_pat)=cnorm(n_pat)+var(i,n_pat)/y(i,n_pat)
  END DO
  cnorm(n_pat)=cnorm(n_pat)/REAL(npts(n_pat))
ELSE                      !IVARI
  IF(rmon1 > 1.0 .AND. rmon2 > 1.0) THEN
    cnorm(n_pat)=rmon1/rmon2
  ELSE
    cnorm(n_pat)=1.0
  END IF
  DO i=1,npts(n_pat)
    var(i,n_pat)=y(i,n_pat)*cnorm(n_pat)
    txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
  END DO
END IF                    !IVARI

```

Ins =6

D1A/D2B standard format prepared by D1A(D2B)SUM (ILL), ADDET(LLB), MPDSUM (LLB) or equivalent programs. The Fortran code reading this kind of file is:

```

READ(i_dat,'(A)',ERR=99999) txt1
READ(i_dat,'(16x,F8.3)',ERR=99999) step(n_pat)
READ(i_dat,'(F8.3)',ERR=99999)thmin(n_pat)
READ(i_dat,'(2F8.0)',ERR=99999) rmoni,rmoniold
IF(rmoniold < 1.) THEN
  cnorm(n_pat)=1.00
  rmoniold=rmoni
ELSE
  cnorm(n_pat)=rmoni/rmoniold
END IF
npunt = nint(18./step(n_pat))
j=0
DO i=1,npunt
  READ(i_dat,3,ERR=99999)(iww(j+no,n_pat),y(j+no,n_pat),no=1,10)
  IF(ABS(y(j+1,n_pat)+1000.) < 1.e-03) EXIT
  j = j+10
END DO
j=j-10
npts(n_pat)=j
thmax(n_pat) = thmin(n_pat)+(npts(n_pat)-1)*step(n_pat)
DO i=1,npts(n_pat)
  IF (y(i,n_pat) <= 0.00001) y(i,n_pat) = 1.0
  IF (iww(i,n_pat) == 0) iww(i,n_pat) = 1
  var(i,n_pat) = cnorm(n_pat)*y(i,n_pat)/REAL(iww(i,n_pat))
  txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
END DO

```

Ins =7

Files from D4 or D20L. The Fortran code reading this kind of file is:

```
READ(i_dat, '(I5,1X,F5.0,1X,A72)',ERR=99999)npts(n_pat),tsamp(n_pat),txt1
READ(i_dat, '(5(F7.2,1X,F8.1,1X,F5.1))',ERR=99999) &
  (txv(i,n_pat),y(i,n_pat),var(i,n_pat),i=1,npts(n_pat))
iconstep(n_pat)=0
cnorm(n_pat)=0.0
step(n_pat)=txv(2,n_pat)-txv(1,n_pat)
thmin(n_pat)=txv(1,n_pat)
thmax(n_pat)=txv(npts(n_pat),n_pat)
DO i=1,npts(n_pat)
  var(i,n_pat)=0.5*var(i,n_pat)*var(i,n_pat)
END DO
```

Ins =8

Data from DMC at Paul Scherrer Institute. The Fortran code reading this kind of file is:

```
READ(i_dat, '(A)',ERR=99999)txt1
READ(i_dat, '(A)',ERR=99999)txt2
READ(i_dat,*,ERR=99999) thmin(n_pat),step(n_pat),thmax(n_pat)
npts(n_pat) = (thmax(n_pat) - thmin(n_pat))/step(n_pat) + 1.005
READ(i_dat, '(10F8.0)',ERR=99999) (y(i,n_pat),i=1,npts(n_pat))
READ(i_dat, '(10F8.0)',ERR=99999) (var(i,n_pat),i=1,npts(n_pat))
DO i=1,npts(n_pat)
  var(i,n_pat) = var(i,n_pat)*var(i,n_pat)
  txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
END DO
```

Ins =9

Data of file CODFIL.uxd generated by the Socabim software on x-rays diffractometer.

Ins =10

X, Y, Sigma format with header lines. In all cases the first 6 lines are considered as comments. If in the first line (left adjusted) appears the keyword XYDATA, then the following 5 lines are considered as the heading of the file. Among these 5 lines the following keywords and values have a meaning to the program:

INTER *fac_x fac_y Interpol Stepin*

TEMP *tsamp*

fac_x internal multiplier of X-values

fac_y internal multiplier of Y and Sigma-values

Interpol

=0 Variable step is used in the program

=1 The variable step data are interpolated internally to the constant step *Stepin*.

=2 Data are supplied directly at constant step

If no sigma values are provided the program assumes that $\sigma(y) = \sqrt{y}$. You can add comments to the data file if they start with the character ! in the first position of the line. These lines are ignored by the program.

The Fortran code reading this type of file is too long to be reproduced here.

Ins =11

Data from variable time X-ray data collection. The first four lines are considered as comments The following lines are:

2Thetai, step, 2Thetaf Comment

(Time, Intensity) in format 5(F6, I10). The program uses the information contained in Time to normalise the observed intensities to the average time <Time> and to calculate the variance of the normalised values. The Fortran code reading this type of file is reproduced below:

```

READ(i_dat,'(A)',ERR=99999)txt1           !1
READ(i_dat,'(A)',ERR=99999)txt2           !2
READ(i_dat,'(A)',ERR=99999)txt3           !3
READ(i_dat,'(A)',ERR=99999)txt4           !4
READ(i_dat,*,ERR=99999)thmin(n_pat),step(n_pat),thmax(n_pat)
npts(n_pat) = (thmax(n_pat)-thmin(n_pat))/step(n_pat)+1.5
READ(i_dat,'(5(F6.0,F10.0))',ERR=99999)(bk(i,n_pat),y(i,n_pat),i=1,npts(n_pat))
!Normalize data to constant time
cnorma=0.0
DO i=1,npts(n_pat)
  IF(bk(i,n_pat) < 1.0E-06)CALL stop_cond(' Zero time in *.DAT ',1)
  cnorma=cnorma+bk(i,n_pat)
  txv(i,n_pat)= thmin(n_pat)+(i-1)*step(n_pat)
END DO
cnorma=cnorma/REAL(npts(n_pat))
DO i=1,npts(n_pat)
  y(i,n_pat)=y(i,n_pat)*cnorma/bk(i,n_pat)
  var(i,n_pat)=y(i,n_pat)
  bk(i,n_pat)=0.0
END DO

```

Ins =12

The input data file conforms to GSAS standard data file.
 BINTYP = LOG6, TIME_MAP and LPSD are not yet available.
 The subroutine reading GSAS formats is available on request.

CODFILn.hkl, CODFIL.int or HKLn.hkl

The format of CODFILn.hkl files is the following:

For $ABS(\mathbf{Irf}(n_pat)) < 4$:

The first two lines are read as titles (characters) The rest of the lines consist of the following items:

1) No propagation vectors

h	k	l	m		(<u>Irf</u> =1)	(free format)
h	k	l	m	Coeff	(<u>Irf</u> =1 & <u>Sol</u> =1)	(")
h	k	l	m	Intensity (or F)	(<u>Irf</u> =2)	(")
h	k	l	m	Freal Fimag	(<u>Irf</u> =3)	(")

2) Nvk propagation vectors

In the third line you have to give the number of propagation vectors in format (32x, i2), then you give Nvk lines with: nv K_1 K_2 K_3 , where nv is the ordinal number of \mathbf{k} and K_i are the components of \mathbf{k} in free format.

The rest of lines contain the following sets of items:

h	k	l	nv	m		(<u>Irf</u> =1)	(free format)
h	k	l	nv	m	COEFF	(<u>Irf</u> =1 & <u>Sol</u> =1)	(")
h	k	l	nv	m	Intensity (or F)	(<u>Irf</u> =2)	(")
h	k	l	nv	m	Freal Fimag	(<u>Irf</u> =3)	(")

Note: The generated files when Jbt=2,3 may content additional items that are not used by **FullProf**. These items (sigma, angle, FWHM) can be used by other programs. The case Irf=1 and Sol=1 is to be used when shifts of Bragg reflections are observed and a model for it () is known. The user must provide the value of the coefficient COEFF for each reflection.

For ABS(**lrf**(*n_pat*))=4:

- The first line is considered as a **TITLE**
- In the second line the format of the intensity data to be read below is given.
Example: (3i4,2f10.2,i4,3f8.4)
Do not forget parentheses. The program expect to read $h, k, l, F^2, \sigma(F^2), code$, and three angles (or coefficients for extinction calculations) in the case of no propagation vectors. The format must be provided for all items even if some of them (particularly the *code* and the three final items) are not provided.
- **R_lambda**(*n*), **Itpdata**, **ipow**(*n*)
This third line is read in free format, the three items must be given. The meaning of the three items is:

R_lambda(*n*)

Wavelength for phase *n*

Itpdata

= 0 Square of structure factors (F^2) and $\sigma(F^2)$ are input.

= 1 Structure factors (F) and $\sigma(F)$ are input. These quantities are transformed internally to case **Itpdata**=0 (F^2 and $\sigma(F^2)$).

Ipow(*n*)

= 0 Single crystal observations.

= 1 Twinned single crystal observations. At present up to 6 *hkl*'s can contribute to a single observation.

= 2 Powder integrated intensities. In this case cluster of peaks can be given. For this case **Itpdata** is irrelevant.

- **cmono**, **Rpolarz**'s

This fourth line is optional (to be given only for X-rays and **Ipow**=2). The items correspond to variables **Cthm** and **Rpolarz** for monochromator polarisation correction.

The rest of lines correspond to the list of integrated intensities or structure factors. The particular set of items to be read depends on whether there are propagation vectors given or not.

1) No propagation vectors. The set of items read by the program is:

$h \ k \ l \ G_{obs}, \sigma(G_{obs}) \ code \ c_1 \ c_2 \ c_3$

2) Propagation vectors

In the third/fourth line you have to give the number of propagation vectors N_{ok} that must be equal to **Nvk**, then you give **Nvk** lines with: $nv \ K_1 \ K_2 \ K_3$, where *nv* is the ordinal number of **k** and K_i are the components of **k** in free format.

The rest of lines contain the following sets of items:

$h \ k \ l \ nv \ G_{obs}, \sigma(G_{obs}) \ code \ c_1 \ c_2 \ c_3$

The format of the data corresponds to that given explicitly in line 2 of the CODFIL*n*.hkl file. No data reduction is performed. The program expects to be provided with an independent set of reflections. The integer *nv* is the ordinal number of the propagation vector corresponding to the current observation *hkl*. G_{obs} and $\sigma(G_{obs})$ have different meanings depending on the value of **Itpdata** and **Ipow**:

Ipow

=0 **Itpdata**= 0 $\Rightarrow G_{obs}=F^2$
= 1 $\Rightarrow G_{obs}=F$

- =1 As above but if $G_{obs} < 0$ the reflection contributes to the next positive observation.
- =2 $G_{obs} = \Sigma \{jLpF^2\}$: Sums of powder integrated intensities

cod: code for reflections indicating the scale factor number to be applied (for twinned crystals or inhomogeneous data).

If **Ipow**=2 and **Nvk**≠0 *cod* is the multiplicity of the reflection.

If **Ipow**=2 and **Nvk**=0 the multiplicity is automatically calculated using the symmetry operators generated from the symbol of the space group.

For Irf=4

c_1, c_2, c_3 : Not yet used (but they are read!) (coefficients for extinction corrections)

For Irf=-4

c_1, c_2 : Real and Imaginary part of the partial calculated structure factor or the reflection. The program will add this contribution to the structure factor calculated with the given atoms:

$$\Rightarrow F_{tot} = F + F_p = (A + iB) + (c_1 + i c_2). \text{ See comments for } \underline{\text{Irf}}=3.$$

Examples:

Twinned Orthorhombic crystal with two domains

h	k	l	Gobs	Sigma	cod	(a,b,c) (b,a,c)
.....						
2	0	0	-1.0	0.0	2	
0	2	0	3221.0	12.1	1	
3	1	1	-1.0	0.0	2	
1	3	1	1221.0	8.2	1	

Powder cluster of peaks

1	1	1	23.2	0.4	1	Isolated peak
.....						
5	3	1	-1.0	0.0	1	Cluster of peaks: four
3	4	2	-1.0	0.0	1	independent reflections
4	4	1	-1.0	0.0	1	contribute to
5	0	3	832.1	9.4	1	<- this observation