

RMCPow

Version 2.00
1 October, 2003

A. Mellergård and R.L. McGreevy
NFL, Studsvik

RMCPow	1
1. INTRODUCTION	2
1.1 Changes from version 1.57	2
2. RMCPow - THE BASIC METHOD	2
3. SCATTERING DEFINITIONS	3
3.1 Neutron nuclear scattering	4
3.2 Neutron magnetic scattering	4
3.3 X-ray scattering	4
3.4 Incoherent scattering and average particle types	5
3.5 Partial structure factors	6
3.6 Analytical models for thermal disorder	6
4. THE RMCPow PROGRAM	7
4.1 Installing the RMCPow files	7
4.2 Running the program	8
4.3 The configuration files	9
4.4 The control data file	9
4.5 The experimental data files	17
4.6 Example files	18
4.7 Other programs available	18
5. RMCPow - SIMULATION DETAILS	18
5.1 Configuration size and shape	18
5.2 Random moves	20
5.3 Constraints	21
5.4 Q spacing and range	23
5.5 Smoothing the diffuse data	23
5.6 Experimental error	24
5.7 Refinable parameters	24

1. Introduction

RMCPow is a general purpose Fortran code for reverse Monte Carlo (RMC) modelling of static and dynamic disorder in crystalline materials using powder diffraction data. Starting from configurations corresponding to the average crystal structure (as obtained from e.g. Rietveld methods) the method can be used to refine the local structure. It has been developed to overcome some of the disadvantages with the standard RMC methods for crystalline materials as applied in the RMCA and RMCX programs. These include the truncation errors related to Fourier inversion of the pair correlation function in RMCA and difficulties to introduce experimental resolution in both RMCA and RMCX, particularly affecting the important constraint on the time-average long-range order that is related to the Bragg scattering information. Furthermore, RMCPow allows a proper way of modelling magnetic structure of localised moments in crystalline materials. This manual first describes the RMCPow method, with respect to differences to the standard RMC method, and then goes on to describe the RMCPow program. For a general introduction to the RMC method we strongly recommend reading the RMCA manual [1] and references therein.

1.1 Changes from version 1.57

- New parameter `dqp` determines internal Q binning, see 4.4.13 and 5.4.
- Options for modelling crystalline structures using fixed Debye-Waller-factors and amorphous structures have been introduced, see 3.6 and 4.4.13.
- Absorption correction option added, see 4.4.14 and 4.5.
- Data can be read as $I(Q)$, $I(2\theta)$ or $I(TOF)$ or generated from the input model configuration, see 4.4.14. Also, scattering data now do *not* need to be given at equidistant points, see 4.5 and 5.4.
- Resolution range parameters made individual for each experiment, see section 4.4.14.
- 2θ or TOF zero-point parameter (possibly refinable) introduced, see 4.4.14.
- New resolution profiles and revised non-linear parameter minimisation, see 4.4.14 and 5.7.2.
- Overall magnetic scale factor added, see 4.4.15.1 and 5.7.1.
- Linear constraints on parameters refined using non-linear least-squares are now applicable, see 4.4.17 and 5.7.

2. RMCPow - the basic method

In RMCA the model scattering is obtained by computing the pair correlation function which is then Fourier transformed and compared to the experimental scattering. The RMCPow method instead uses a direct calculation of the scattered intensities on a mesh of points in reciprocal space. These intensities are subsequently reduced to a powder cross-section by appropriate integration over spherical shells. The mesh is chosen as the reciprocal lattice points of the model configuration cell (hereafter referred to as the configuration cell or cc). Obviously the configuration cell should be chosen with dimensions such that it is a supercell of the conventional unit cell (hereafter called unit cell or uc) to allow periodic boundary conditions. This also means that the reciprocal lattice points of the unit cell are a subset of those of the configuration cell and so the calculated scattering can be separated into Bragg and diffuse scattering (with respect to the unit cell size) and full use of experimental resolution can be applied to the computed powder cross-section.

The total differential scattering cross-section is conventionally split into a coherent and an incoherent (see 3.4) contribution in such a way that the cross-section per atom for a general system of N particles is

$$\frac{d\sigma}{d\Omega} = I^{coh} + I^{inc} = \frac{1}{N} \left| \sum_{j=1}^N f_j \exp(i\mathbf{Q} \cdot \mathbf{r}_j) \right|^2 + I^{inc} \quad (1)$$

where f_j and \mathbf{r}_j are the scattering amplitude and position of particle j . For a configuration cell the coherent cross-section corresponding to a powder experiment is then given by [2]

$$I^{coh}(\mathbf{Q}) = \frac{2\pi^2}{NV_{cc}} \sum_{\tau_{cc}} |F(\tau_{cc})|^2 \frac{R(\mathbf{Q} - \tau_{cc})}{\tau_{cc}^2} \quad (2)$$

Here V_{cc} is the volume of the configuration cell, τ_{cc} runs over the set of reciprocal lattice points for the configuration cell and $R(\mathbf{Q} - \tau_{cc})$ is the experimental resolution function. The amplitude sum at a reciprocal point \mathbf{Q} is simply

$$F(\mathbf{Q}) = \sum_{j=1}^N f_j(\mathbf{Q}) e^{i\mathbf{Q} \cdot \mathbf{r}_j} \quad (3)$$

and runs over all N particles in the configuration cell (since both thermal *and* static fluctuations are being modelled no Debye-Waller factors are present).

For the purposes of RMCPOW we define a structure factor $I(\mathbf{Q})$ as the total differential scattering cross-section per scattering atom. As in RMCA the agreement between the model structure factor, $I^C(\mathbf{Q})$, and the experimental $I^E(\mathbf{Q})$ is probed by a χ^2 -test (m is the number of experimental data points);

$$\chi^2 = \sum_{i=1}^m (I^C(\mathbf{Q}_i) - I^E(\mathbf{Q}_i))^2 / \sigma(\mathbf{Q}_i)^2 \quad (4)$$

In the RMCA program the pair distribution function $G(r)$ is used for intermediate results. RMCPOW instead uses the scattering amplitude sums and the following algorithm is used.

1. Start with an initial configuration as in RMCA. Find all configuration cell reciprocal points within the "experimental sphere" with radius Q_{max} . Separate them into Bragg (if they are also points on the reciprocal lattice of the unit cell) and diffuse points (if they are on the configuration cell reciprocal lattice only).
2. Calculate the amplitude sums for this old (o) configuration.
3. Obtain the Bragg and diffuse contributions to the model structure factor using eq. (2) and add any incoherent scattering.
4. Calculate the difference between the experimental and model structure factors, χ_o^2 , as in eq. (4).
5. Move one atom at random. Calculate the new (n) amplitude sums and model structure factor and the new χ_n^2 .
6. If $\chi_n^2 < \chi_o^2$ the move is accepted and the new configuration becomes the old configuration. If $\chi_n^2 > \chi_o^2$ then the move is accepted with probability $\exp(-(\chi_n^2 - \chi_o^2)/2)$. Otherwise it is rejected.
7. Repeat from step 5.

As this process is iterated χ^2 will initially decrease until it reaches an equilibrium value about which it will fluctuate. The resulting configuration should be a three dimensional structure that is consistent with the experimental total structure factor within the experimental error.

3. Scattering definitions

The algorithm described in the previous section is specifically for modelling a single set of diffraction data, which could be obtained using X-rays, neutrons, or electrons. The RMC method is more general than this simple algorithm in that any set or sets of data which can be directly calculated from the structure can be modelled. It can be applied to isotopic substitution in neutron diffraction or

equivalently to anomalous scattering in X-ray diffraction, to EXAFS and possibly to NMR data. All data sets can be modelled simultaneously by adding the respective χ^2 values. The current version of the RMCPOW program however only allows modelling to one neutron experiment and/or one x-ray experiment.

For a multicomponent system where the fit is to several different total structure factors (indicated by index n) we have

$$\chi^2 = \sum_n \chi_n^2 = \sum_n \sum_{i=1}^{m_n} (I_n^C(Q_i) - I_n^E(Q_i))^2 / \sigma_n(Q_i)^2 \quad (5)$$

For simultaneous fitting of data sets obtained by different experimental techniques the separate χ^2 values are simply summed to give one value. The relative weighting of the different data sets is determined by the choice of the various σ values. Clearly the required computer time increases significantly if multiple data sets are fitted. We give below the definitions for the various types of scattering that are used in RMCPOW.

3.1 Neutron nuclear scattering

For neutron diffraction the atomic scattering amplitudes are simply the coherent scattering lengths, \bar{b}_j , which are independent of Q and can be found in e.g. [3]. Note that the scattering lengths are entered in fm units whereas cross-sections are computed in barns ($1 \text{ fm}^2 = 0.01 \text{ barns}$). A few elements/isotopes have complex scattering lengths; this case is however not yet treated by RMCPOW.

3.2 Neutron magnetic scattering

For magnetic neutron diffraction the atomic scattering amplitudes are the vectors

$$f_j = \frac{\gamma r_0}{2} f_j^m \hat{Q} \times (\boldsymbol{\mu}_j \times \hat{Q}) \quad (6)$$

where $\boldsymbol{\mu}_j$ [μ_B] is an effective single-ion magnetic moment. $\gamma = -1.9132$ is the neutron gyromagnetic ratio and $r_0 = e^2/(4\pi\epsilon_0 m_e c^2) = 2.81 \text{ fm}$ is the classical electron radius. In the RMCPOW program magnetic form-factors f^m are approximated by the analytical expressions

$$f^m(Q = 4\pi \sin \theta / \lambda) = \sum_l a_l \exp(b_l (\sin \theta / \lambda)^2) + c \quad (7)$$

for which values of the coefficients a_l , b_l , and c can be found in [4,5]. Note that RCMPOW uses a classical approach to magnetic moments, i.e. all three components of the magnetic moment vector are considered. μ is then the total single-ion moment, not the z component. In many cases (especially for 3d-elements where the orbital momentum is quenched so that the Landé factor $g=2$) $\mu^2 = g^2 S(S+1)$ is a good estimate but e.g. covalency effects can change this value.

3.3 X-ray scattering

For X-ray diffraction the form-factors are approximated by the analytical expressions

$$f^x(Q = 4\pi \sin \theta / \lambda) = \sum_{l=1}^4 a_l \exp(b_l (\sin \theta / \lambda)^2) + c + \Delta f' + i\Delta f'' \quad (8)$$

for which values of the coefficients a_l , b_l , and c and dispersion corrections $\Delta f'$ and $\Delta f''$ can be found in [6]. With these definitions of form-factors X-ray cross-sections are obtained in electron units (the differential scattering cross-section in barns units is then obtained by a multiplication with $r_0^2 = 0.0790$ barns). Note that the total x-ray structure factor as defined in RMCPOW is not normalised by any average form-factor.

3.4 Incoherent scattering and average particle types

Incoherent scattering, i.e. uncorrelated scattering from particles, occurs mainly because of random nuclear isotopic substitution for neutron scattering and inelastic Compton scattering for x-rays. For a system with concentration c_k of element k the neutron incoherent scattering cross-section is

$$I^{inc} = \frac{1}{4\pi} \sum_k c_k \sigma_k^{inc} \quad (9)$$

where σ_k^{inc} can also be found in [3]. For properly reduced experimental data this is the expected minimum scattering level. It is possible to subtract this contribution from $I^E(Q)$ prior to running the simulation. However in general it is better to allow the program to add this estimate of incoherent scattering to the model scattering and then refine a constant background to be subtracted from the experimental data since it might be affected by small normalisation errors.

X-ray Compton scattering may be approximated in RMCPOW by interpolating the values for heavy atoms given in Table 3.4.4.1 in [4]. The table is given for values of a parameter $w \propto Q/Z^{2/3}$ so the effective number of electrons Z should be supplied for each particle type in the control data file.

Additional incoherent scattering is introduced if two or more atom types randomly occupy some equivalent (sublattice) sites and can be dealt with using two different approaches in the context of RMCPOW modelling: Either a) the scattering amplitude for particles on a sublattice can be replaced by the occupational average or b) the different atomic types can be introduced explicitly by first generating all relevant positions and then randomising their atomic identities in due proportions. The first approach can also be used in the initial stage of a simulation, even if the distribution is not expected to be completely random, and later individual identities can be assigned. For nuclear neutron scattering the occupational average (denoted by $\langle \dots \rangle$ and taken over the occupancies c_{kl}^{occ} of the atom types l at the sites for particles k) is simply

$$\langle \bar{b} \rangle_k = \sum_l c_{kl}^{occ} \bar{b}_{kl} \quad (10)$$

and is independent of Q . In the case of x-rays and magnetic neutron scattering the Q -dependency must also be taken into account. For this reason the program offers the possibility to assign occupation numbers (see 4.4) for a collection of atomic species for each particle type and then the average scattering amplitudes are computed internally.

When average scattering amplitudes are used for Bragg scattering there is an additional incoherent diffuse contribution for particle type k , the Laue monotonic scattering

$$c_k \left(\langle \overline{f^2} \rangle_k - \langle \bar{f} \rangle_k^2 \right) = c_k \left(\sum_l c_{kl}^{occ} \overline{|f_{kl}|^2} - \left| \sum_l c_{kl}^{occ} \bar{f}_{kl} \right|^2 \right) \quad (11)$$

where the bar over f indicates nuclear averaging in the case of neutrons. Although this term normally is small it is important for the overall consistency to add this to the total structure factor. For neutron magnetic scattering there is also a spin orientation dependence so in this case eq. (11) should be multiplied by $2/3$ if spins are uncorrelated (if there is short- or long-range magnetic order then this is still a proper estimate of the *average* extra magnetic scattering). If there are multiple sets of average types each will have its own average coherent scattering amplitude and additive incoherent contribution.

The calculated scattering cross-sections are normalised by the total effective number N_{eff} of scattering particles, obtained as the sum of the occupancies of each particle type times the number of

particles of each type. Occupation assignments can be different for each experiment being modelled but the program checks that N_{eff} is the same for all experiments.

3.5 Partial structure factors

To facilitate the inspection of the scattering contributions from different atomic types an option to compute the partial coherent cross-sections is included in the program (parameter `ijob` = 1, see 4.4.13). This is done for the initial configuration only and is not further updated since it does not affect the progress of the simulation and is rather time-consuming. These partials are computed according to eq. (2) but summing only over relevant atoms for each partial, the total sum of partials being equal to $I^{coh}(Q)$. Note that individual partials can be negative over part of or the whole scattering range, only the sum of all coherent scattering must be positive definite.

3.6 Analytical models for thermal disorder

RMCPow can also be run in a mode similar to simulated annealing. Fitting is still done to the experimental data profile directly but here thermal diffuse scattering (TDS) is dealt with using analytical expressions rather than explicitly introducing (thermal) displacements from equilibrium positions. This can be used to refine the average atomic and/or magnetic structure, still keeping a constraint on the shape and amount of diffuse scattering, or simply to get an estimate of what TDS effects would look like in some particular system. The expressions that are used neglect the details of the crystal symmetry but they give the correct amount of diffuse scattering. They have been obtained by slight generalisations of similar expressions in [7,8].

Two options are available, `ijob` = 2 or 3, for estimating the TDS. In both modes amplitude sums are calculated separately for each particle type and at all points τ of the configuration cell, i.e. assuming equality of configuration and unit cells. These modes are suitable for small configurations (perhaps only one primitive cell) and allows refinement of individual mean square displacements. Using the Debye-Waller factor for particle type k , $W_k(Q) = \frac{1}{2}Q^2 u_k^2$, with u_k^2 the mean square displacement (msd) for particles of type k , we now have for the Bragg scattering amplitudes

$$F(\mathbf{Q}) = \sum_{j=1}^N f_j(\mathbf{Q}) e^{W_j(Q)} e^{i\mathbf{Q}\cdot\mathbf{r}_j} \quad (12)$$

where \mathbf{r}_j are now "ideal" equilibrium positions. For `ijob` = 2 we assume uncorrelated, Gaussian distributed, displacements and so single- and multi-phonon scattering can be approximated by

$$I_{diffuse}(Q) = \sum_k c_k |f_k|^2 (1 - \exp[-2W_k(Q)]) \quad (13)$$

For `ijob` = 3 we assume a Debye spectrum $\hbar\omega(q) = k_B T_D q/k_D$ (T_D is the Debye temperature and $k_D = (6\pi N/V_{cc})^{1/3}$ is the Debye wave-vector cut-off) giving correlated displacements. The msd for particle k with effective mass M_k is now taken to be

$$u_k^2(T) = \frac{3\hbar^2}{4M_k k_B T_D} \frac{2T}{T_D} \int_0^{T_D/2T} \coth(x) dx = u_k^2(0) I(T) \quad (14)$$

where $u_k^2(0) = 3\hbar^2/(4M_k k_B T_D)$ is the zero-point msd. In the neighbourhood $|\mathbf{Q} - \mathbf{a}| < k_D$ of Bragg point τ single-phonon amplitude sums will be

$$F_{single}(\mathbf{Q}, \tau) = \sum_{j=1}^N u_j(T) f_j(\mathbf{Q}) e^{-W_j(Q)} e^{i\tau\cdot\mathbf{r}_j} \quad (14)$$

and the corresponding diffuse powder scattering term (integrated over constant Q) is

$$I_{\text{single}}(Q, \tau) = \frac{2}{3N^2} \frac{2Tk_D^2}{T_D I(T)} \frac{Q}{\tau} \ln \left[\sinh\left(\frac{T_D}{2T}\right) / \sinh\left(\frac{T_D}{2T} \left| \frac{Q - \tau}{k_D} \right| \right) \right] \left| F_{\text{single}}(Q, \tau) \right|^2 \quad (15)$$

in the region $|Q - \tau| < k_D$ for each τ . For scattering around $\tau = 0$,

$$I_{\text{single}}(Q, 0) = \frac{2}{3N^2} \frac{k_D Q}{I(T)} \coth\left(\frac{T_D}{2T} \frac{Q}{k_D}\right) \left| F_{\text{single}}(Q, 0) \right|^2 \quad (15)$$

Since multi-phonon scattering is rather featureless it can be approximated by the expression for the uncorrelated case,

$$I_{\text{multi}}(Q) = \sum_k c_k |f_k|^2 (1 - \exp[-2W_k(Q)] [1 + 2W_k(Q)]) \quad (16)$$

In both cases diffuse magnetic scattering (if present) is given by

$$I_{\text{diffuse}}^m(Q) = \frac{2}{3} \left(\frac{\gamma r_0}{2} \right)^2 \sum_k c_k |f_k^m|^2 (\mu_k^2 - \mu_{kz}^2 e^{-2W_k(Q)}) \quad (17)$$

where μ_k is the total magnetic moment and μ_{kz} is the average component along the ordering axis (which direction can be different for the individual spins of type k as given by their unit vectors). Magnetic Bragg scattering amplitude sums are modified analogous to eq. 14.

For large configurations partial amplitudes becomes impractical and for these cases the `ijob = 4` option can be used which is similar to the `ijob = 0` and `2` options but using fixed Debye-Waller factors. Also in this option the amplitude sums are calculated at each configuration r.l.p. but here the diffuse points in between the unit cell Bragg points are by definition thought to build up the diffuse scattering so no analytical terms are added.

Finally, going to the extreme case of amorphous structures, this situation corresponds to *all* r.l.p.'s being diffuse points. Although RMCPOW was not designed for these purposes it can sometimes (when sharp but non-Bragg features are present) be useful to model amorphous structures using the RMCPOW approach. This option is obtained with `ijob = 5`.

4. The RMCPOW program

4.1 Installing the RMCPOW files

The current version of RMCPOW (version 1) is written as a general-purpose program for modelling multi-component systems using experimental diffraction data as constraints. The program is written in Fortran-90 and compiled under MS-Fortran Powerstation, VMS Fortran (for ALPHA-stations) and ABSOFT Fortran for LINUX as well as some other versions of the language. In the MS-Windows version some QuickWin procedures are used for running as a Windows application. In the standard version of the program array space, depending on the number of atoms etc, is allocated at run-time. Static memory versions are however also available on request. Plotting is done using PGPLOT routines [9].

If RMCPOW is downloaded as part of the WinNFLP suite of programs, then all necessary set-up should have been done following the installation procedure in the WinNFLP manual. The WinNFLP suite can be downloaded from <ftp://ftp.studsvik.uu.se/Pub/WinNFLP> or <http://www.studsvik.uu.se>. Some of the program features that involve reading values from tables are only available when RMCPOW is run as a part of the WinNFLP suite. The examples provided are also more interesting when run in this mode. Otherwise, if the RMCPOW program is downloaded as standalone from <ftp://ftp.studsvik.uu.se/Pub/RMCPOW> then the package contains the following files:

readme.txt	- release information
rmcpow.exe	- the executable file
rmcpow.pdf	- this manual in PDF format
nxsection.dat	- neutron scattering parameters
RMC_logicals.ini	- environment variable definitions
grfont.dat	- PGPLOT fonts
rmcpow_mno.zip	- example files for modelling MnO
rmcpow_bafewo.zip	- example files for modelling Ba ₂ FeWO ₆
rmcpow_agbr.zip	- example files for modelling AgBr

which should be extracted into a single directory, e.g. c:\nflp\rmc.

For the plotting to work with VMS or LINUX, Xwindows have to run on the machine. Also, for proper plotting, the PGPLOT font and device environment variables should be defined. In Windows the MCGR program will attempt to read the RMC_logicals.ini file to define

```
PGPLOT_FONT = c:\nflp\rmc\grfont.dat
PGPLOT_DEV = /W9
```

assuming you extracted files into c:\nflp\rmc. For LINUX version this could be done in your login file using the setenv command, also set the path to include your RMCPOW directory. For VMS version define a symbol to run RMCPOW.

```
$ RMCPOW := "$user$disk:[user.directory] RMCPOW.exe"
```

This and the environment settings could be done in your login file. Both LINUX and VMS works best with device /xw. If PGPLOT_DEV is not defined, or set to ?, a list of selectable modes will be displayed at run-time.

4.2 Running the program

When RMCPOW is run it must be supplied with a name used for all its data files. The Windows version is simply started by clicking the program icon, or it may be started from the RMC menu using the WinNFLP program. An Open file window appears where the user can choose the relevant name.dat file and then the simulation will start. During the run menus can be accessed to save current data promptly or to exit the program with or without saving. Plotting of the current fitted patterns and the progress of χ^2 can also be turned on/off. A record of the last work directory will be saved to a file rmcpow.ini in the directory where the exe file is stored so that easy continuation is possible next time the program is run.

For all other versions it is intended that the program be run in such a way that the data file name can be supplied on the command line. The Linux version of the machine dependent routines allow this with no further ado. On VMS systems it is necessary to define a symbol as described above and then the program can be run. For example

```
rmcpow name
```

would run the RMCPOW program with files called *name* with various extensions. This name is given to a number of files used by the program. The general program parameters are supplied in the control file name.dat. There must also be a file *name.cfg* containing the positions, or configuration, of the atoms. It is only necessary to calculate the full scattering amplitudes once at the beginning of the first run; thereafter only the change in the amplitude sums needs to be updated. To facilitate this the different amplitude sums are written to binary files. These files are given the names *name.aqz*, where z is n for nuclear neutron scattering, m for magnetic neutron scattering and x for x-ray scattering. At specified intervals and when the program starts and terminates the amplitude and configuration files are updated. Also, the output results (experimental and calculated structure factors) will be written to

name.out. Information on the parameters read from *name.dat* and the status of the program are written to *name.log*. Sample files are included in the examples.

On the next run the amplitudes will be read from the relevant files. If any of the required files does not exist, or if there is a read error, or *ijob* = 1, or *lexists* = 0, then amplitudes will be recalculated. Also if any of the following parameters are being changed the amplitudes *have* to be recalculated since these parameters affect either the number of reciprocal points used or the value of the amplitudes sums: *nq2*, the maximum *Q* point to be fitted; *qwmax*, the additional *Q* range used to eliminate resolution truncation effects at *Q(qwmax)*; any lattice parameters and any parameters determining the scattering amplitudes of particle types (scattering lengths, form factor parameters, effective magnetic moments). Changes in these parameters are not monitored by the program so the user has to control this manually, e.g. by setting *lexists* = 0.

If *ncoll* > 0 saved configurations will be written to *name.cfg**, with * replaced by an incremented number. This option is to be used for sampling independent configurations when the simulation has converged.

4.3 The configuration files

In general any parallelepiped can be used for configurations. Since a crystalline system is being modelled the box dimensions *must* be chosen to accommodate an integral number of unit cells in each direction, with the numbers preferably being chosen to make the configuration as cubic as possible. The atomic coordinates are defined in terms of the box coordinates and are normalised to limits of ± 1.0 . The box coordinates are then defined in terms of the laboratory coordinates by the matrix given in the top of the configuration file. The values in this matrix are recalculated from the corresponding unit cell parameters in *name.dat* every time the program is run and these are then written to the new configuration file. For a cubic box the diagonal terms in the matrix are equal to half the box length (*L*/2) in Å.

The numbers of different types of atoms and the order of their coordinates are given in the top of the configuration file. The atomic coordinates then follow in a single list. The choice of a format in which the system is not explicitly identified in the configuration file (other than by the title) is made deliberately; configurations can then be easily modified or used as starting configurations for different systems with a minimal amount of editing. The same format of configuration file is also used for molecular systems, where both coordinates and Euler angles define the molecules. For this reason atoms are defined as molecules with a single atomic site. An example configuration file is included in the package.

For modelling magnetic structures a similar spin configuration file, *name.scfg*, is used. Here the coordinates are replaced by the unit vectors of the magnetic moments (i.e. orientations w.r.t the Cartesian coordinate system defined by the unit cell matrix given in *name.dat*) for each magnetic atom. The magnitudes of the moments of particular types are given as parameters in the *name.dat* file. For each magnetic atom type there have to be as many spins as there are atoms. The order (within a specific type) of the magnetic atoms has to be the same in the atomic and spin configuration files. However, non-magnetic atoms should not be included in the spin configuration file.

4.4 The control data file

The RMCPOW control data file, *name.dat*, can be thought of as being divided into various sections. The description that follows is given section by section. Items specified in the same block below can in most cases be split into multiple lines in the data file. Also a comment can be entered after the final parameter of each block (on the same line). Codes for refining the background, normalisation and experimental resolution are used such that a zero value indicates a fix parameter and values > 0 free parameters, possibly constrained to other parameters. An example control data file is included in the package.

4.4.1 General parameters.

title	character*80	A title for the run.
ncoll,ncycles	integer	ncoll is the number of configurations to collect after convergence; ncycles is then the average number of moves per atom inbetween each save. The program will stop when ncoll configurations have been collected.
iprint, iplot, timelim, timesav	2*integer,2*real	iprint determines how often a summary will be written to the standard output. It will be written after every iprint moves generated. iprint also determines how often a full non-linear fit, see 5.7, will be performed, and how often a plot of the current results will be updated. If iplot > 0 then plotting will be turned on at start-up. In the Windows version it can be turned on later from the Plot menu. iprint should not be too small or otherwise time will be wasted in continually writing to the .log file and this will become very large; timelim is the time the program should run for, in minutes; timesav is the interval at which the results should be saved to the output files (they are always saved when the program starts and ends anyway). If the program is left running for a very long time it is best to save the results every now and then, perhaps once an hour. The results should not be saved too often as this simply wastes time writing large files.
a1,a2,a3, g1,g2,g3	real*6	The unit cell lattice parameters (lengths [Å] and angles [°])
ca1,ca2,ca3, cg1,cg2,cg3	integer*6	Codes for refining lattice parameters (fit option not yet available, these should be zeroes).
isym,ncell	integer*4	isym - Centering symmetry code, $-8 < \text{isym} < 8$: 1=P, 2=A, 3=B, 4=C, 5=I, 6=F, 7=R (rhombohedral setting) and 8=Hexagonal, isym also affects choice of Cartesian coordinates, see 5.1; ncell - The number of intermediate cells in the configuration, along each direction.
pui,denui	integer*10	pui – The 9 elements of the UC to IC transformation matrix, see 5.1; denui – The common denominator of pui. This and next line only read if isym < 0.
icart	integer	Determines the Cartesian coordinate system if isym < 0.
ntypes,ntypesm	integer*2	ntypes – no. of particle types; ntypesm – no. of magnetic types.
attypes	integer*ntypesm	For each magnetic type the particle type to be associated with. Only read if ntypesm > 0.

4.4.2 Parameters for moves.

delta	real array	The maximum move for each type of particle. Recommended values are in the range 0.01-0.2. A value of zero is allowable, in which case the program will not attempt to move those particles. See section 5.2 for a description of the use of this parameter.
deltam	real array	The maximum rotation move [°] for each magnetic type. Only

		read if $n_{typesm} > 0$.
nswap,swapfrac	integer,real	nswap - The number of kinds of particle swap moves, see 5.2, $0 \leq n_{swap} \leq 10$; swapfrac - the fraction of generated moves that will be of the swap kind.
swaptyp1, swaptyp2, swapcut	2*integer,real	swaptyp1/swaptyp2 - The first and second particle types to swap; swapcut - the distance from the first particle, within which to find the second particle. If swapcut=0 then any particle of type swaptyp2 can be chosen. This block is repeated for each of the nswap types.
nref,reffrac	integer,real	nref - The number of kinds of particle reference moves, see 5.2, $0 \leq n_{ref} \leq 10$; reffrac - the fraction of generated moves that will be of the reference kind.
reftyp1, reftyp2, refcut, deltaref	2*integer,2*real	reftyp1 - The particle type to move; reftyp2 - The particle type to move to; refcut - the distance from the first particle, within which to find the second particle. If refcut=0 then any particle of type reftyp2 can be chosen. If deltaref > 0, then it is the maximum move that particle 1 is displaced from the reference position of particle 2. This block is repeated for each of the nref types.

4.4.3 Parameters for closest approach constraints.

rcut	real array	The closest allowed approach of two particles. There should be a value for each partial $g(r)$. There are thus $n(n + 1)/2$ values for a n-component system. As in all other parts of the program the partials are given in the order 1-1, 1-2, ..., 1-n, 2-2, ..., 2-n, ... n-n. See section 5.3.1 for a description of the use of these parameters.
------	------------	--

4.4.4 Parameters for co-ordination constraints (5.3.2).

ncoord	integer	The number of co-ordination constraints, $0 \leq n_{coord} \leq 20$.
--------	---------	---

4.4.4.1 Parameters for each co-ordination constraint (one block per constraint).

typc	integer	The type of the central particle.
typn	integer	The type of the neighbour particles.
rcoord	2*real	The two distances between which to calculate the co-ordination number.
coordno	2*integer	The lower and upper bound of the desired co-ordination number.
coordfrac	real	The fraction of the central particles desired to have this co-ordination constraint.
sigmac	real	Effectively a parameter weighting this constraint relative to others and the fit to the data.

4.4.5 Parameters for average co-ordination constraints (5.3.2).

navc	integer	The number of average co-ordination constraints, $0 \leq n_{avc} \leq 20$.
------	---------	---

4.4.5.1 Parameters for each average co-ordination constraint (one block per constraint).

avtypec	integer	The type of the central particle.
avtypen	integer	The type of the neighbour particles.
ravc	2*real	The two distances between which to calculate the co-ordination number.
avcno	real	The desired average co-ordination number.
sigmaavc	real	Effectively a parameter weighting this constraint relative to others and the fit to the data.

4.4.6 Parameters for bond valence constraints (5.3.3).

nval	integer	The number of average valence constraints, $0 \leq nval \leq 20$.
------	---------	--

4.4.6.1 Parameters for each bond valence constraint (one block per constraint).

valtypec	integer	The type of the central particle.
valtypen	integer	The type of the neighbour particles.
rval	real	The maximum bond distance for which to calculate the bond valence sum.
valreq	real	The required valence.
ival	integer	Code to select bond valence expression, ival =1 power law, ival =2 exponential.
r0val, bval	real*2	Parameters for the ival expression, see 5.3.3.
sigmaval	real	Effectively a parameter weighting this constraint relative to others and the fit to the data.

4.4.7 Parameters for average bond valence constraints (5.3.3).

naval	integer	The number of average valence constraints, $0 \leq nval \leq 20$.
-------	---------	--

4.4.7.1 Parameters for each average bond valence constraint (one block per constraint).

avaltypec	integer	The type of the central particle.
avaltypen	integer	The type of the neighbour particles.
raval	real	The maximum bond distance for which to calculate the bond valence sum.
avalreq	real	The required average valence.
ival	integer	Code to select bond valence expression, ival =1 power law, ival =2 exponential.
r0aval, baval	real*2	Parameters for the ival expression, see 5.3.3.
sigmaaval	real	Effectively a parameter weighting this constraint relative to others and the fit to the data.

4.4.8 Parameters for average magnetisation constraints (5.3.4).

navm	integer	The number of average magnetisation constraints, $0 \leq ntypesm$.
------	---------	---

4.4.8.1 Parameters for each average magnetisation constraint (one block per constraint).

avmtype, avmno, sigmaavm	integer,2*real	avmtype – The atomic type for which to constrain the average magnetisation; muav – average magnetisation constraint [μ_B]; sigmam – effectively a parameter weighting the average magnetisation constraint for this type relative to other constraints and the fit to the data.
--------------------------------	----------------	---

4.4.9 Parameters for spin co-ordination constraints (5.3.5).

nsc	integer	The number of spin co-ordination constraints, $0 \leq \text{nsc} \leq 20$.
-----	---------	---

4.4.9.1 Parameters for each spin co-ordination constraint (one block per constraint).

sctypesc	integer	The type of the central particle.
sctypen	integer	The type of the neighbour particles.
rsc	2*real	The two distances between which to calculate the co-ordination number.
scno	2*integer	The lower and upper bound of the desired co-ordination number.
sthsc	2*real	The lower and upper bound of the desired spin-spin cosines.
scfrac	real	The fraction of the central particles desired to have this co-ordination constraint.
sigmasc	real	Effectively a parameter weighting this constraint relative to others and the fit to the data.

4.4.10 Parameters for average spin co-ordination constraints (5.3.5).

nasc	integer	The number of average spin co-ordination constraints, $0 \leq \text{nasc} \leq 20$.
------	---------	--

4.4.10.1 Parameters for each average spin co-ordination constraint (one block per constraint).

asctypesc	integer	The type of the central particle.
asctypen	integer	The type of the neighbour particles.
rasc	2*real	The two distances between which to calculate the average spin-spin co-ordination.
ascno	real	The desired average spin-spin cosines.
sigmaasc	real	Effectively a parameter weighting this constraint relative to others and the fit to the data.

4.4.11 Parameters for magnetic potential constraints (5.3.6).

nmpot	integer	The number of magnetic potential constraints, $0 \leq \text{nmpot} \leq 20$.
Tmpot, sigmagpot	2*real	Tmpot - the absolute temperature [K]; sigmagpot - effectively a parameter weighting this constraint relative to others and the fit to the data. This line only read if nmpot > 0.

4.4.11.1 Parameters for each magnetic potential constraint (one block per constraint).

mpottypesc	integer	The type of the central particle.
mpottypen	integer	The type of the neighbour particles.
Rmpot	2*real	The two distances between which this magnetic potential

		constraint is defined.
Jmpot	real	The magnetic exchange interaction energy [K].

4.4.12 Parameters for number of experimental data sets

nexpn,nexpc, nsx	3*integer	nexpn and nexpc - the number of neutron and x-ray experiment constraints. In the current version 0 or 1 are the only possible values; if nsx > 0 a single crystal output file is produced (in current version only for neutron scattering).
h1,h2,k1,k2,l1,l2	6*integer	The min and max limits of the single crystal output in units of <i>configuration</i> reciprocal basis vectors. NB: due to the reciprocal lattice inversion symmetry $l1 \geq 0$. This line only read if nsx > 0.

4.4.13 General parameters for experimental data sets (only read if nexpn>0 or nexpc>0)

dqp,swdt	2*real	dqp is the step of the internal Q binning used for reduction to powder cross-sections <i>before</i> resolution is applied; If swdt>0 Savitsky-Golay smoothing of the diffuse scattering is performed, swdt then determines the width of the smoothing interval, see section 5.5.
ijob,iexists	2*integer	If ijob = 0 a normal RMCPOW simulation is performed. If ijob = 1 partial structure factors are calculated for the initial configuration and written to the output file, see section 3.5; If ijob = 2 or 3 the program is run with analytical estimates of thermal disorder scattering, see 3.6, for ijob = 2 D-W uncorrelated diffuse scattering is added and for ijob = 3 correlated diffuse scattering is added. ijob = 4 is the same as ijob = 0, but with fixed D-W factors. If ijob = 5 then <i>all</i> r.l.p.'s are treated as diffuse points, i.e. this option is suitable for amorphous structures; If iexists = 0 amplitudes are recalculated each time, if iexists = 1 the programs attempts to read the relevant amplitude files, if anyone required doesn't exist the amplitudes are recalculated.

4.4.13.1 Parameters for uncorrelated thermal models if ijob = 2 or ijob = 4.

u2	real array	u2 - Mean square displacements [\AA^2] for all particle types.
cu2	integer array	cu2 - codes for refining cu2. If ijob = 4 then cu2 should all be zero.

4.4.13.2 Parameters for correlated thermal models if ijob = 3.

u2,TD,Tabs	real array,real*2	u2 - Mean square displacements [\AA^2] for all particle types; TD - Debye temperature [K]; Tabs - absolute temperature [K].
cu2,cTD	integer array, integer	cu2 - codes for refining cu2; cTD - code for refining TD.

4.4.13.3 Parameters for thermal models if $1 < \text{ijob} < 5$ and ntypesm > 0.

muz	real array	muz - average "z" component [μ_B], see 3.6, for all magnetic types.
cmuz	integer array	cmuz - codes for refining muz.

4.4.14 Parameters repeated for each neutron experimental data set

file	character*80	File containing the experimental data or name for a file with "experimental" data generated based on the input model and parameters, see nx below.
xin,xout	2*integer	Determines the function type $f(x)$ of the input and output data, x_{in} (or x_{out}) = 0 $\Leftrightarrow x = Q$, $x_{in} = 1 \Leftrightarrow x = 2\theta$ or $x_{in} = 2 \Leftrightarrow x = TOF$.
x1,x2,nx	2*real,integer	The min and max x values in the experimental data file to be fitted. If x_2 is larger than the max x in the data file, the last data point used will be the last point given in the data file; If $n_x = 0$, data is read from file. If $n_x > 0$ a new dataset is created and written to file with n_x equidistant points from x_1 to x_2 .
nexcl	integer	No. of excluded regions, not to be fitted.
xexcl1,xexcl2	2* integer	The min and max x for each excluded region. This line repeated n_{excl} times.
mur	real	μ_r is the macroscopic absorption, see 4.5, if $\mu_r = 0$ then no absorption correction is applied.
isig,sigma	integer,real	$isig$ - code for determining the experimental standard errors. If $isig=1$ standard errors are read from a third column in the data file. If $isig=2$ standard errors are estimated as the square root of the experimental data. If $isig=3$ standard errors are constant =1; σ - standard errors obtained as above are scaled by this value, effectively a parameter weighting this constraint relative to others and the fit to the data.
nbpol,nbfix	integer	n_{bpol} is the no of terms in a polynomial background to be subtracted from the data, $0 \leq n_{bpol} \leq 5$. $n_{bpol} - 1$ is then the polynomial degree; n_{bfix} - no of additional "fix" backgrounds, see 5.7.1, $0 \leq n_{bfix} \leq 2$.
$\alpha_0, \alpha_1, \dots, \alpha_{n_{bpol}}, \alpha_1, \dots$	$n_{bpol} * \text{real}, n_{bpol} * \text{integer}$	α_i - Initial coefficients for the polynomial background; $c\alpha$ - Codes for refining the coefficients. This line only read if $n_{bpol} > 0$.
bfixfile	character*80	File containing the fix background data. Q values must be the same as in the experimental data file. This line only read if $n_{bfix} > 0$, and is repeated for each fix background.
$\beta_1, \beta_2, \dots, \beta_{n_{bfix}}, \beta_1, \beta_2, \dots$	real,integer	β_i - Initial scale factors for the fix backgrounds; $c\beta_i$ - Codes for refining β_i . This line only read if $n_{bfix} > 0$.
$\beta, c\beta$	real,integer	β - Initial scale factor for the model scattering; $c\beta$ - Code for refining β .
rwdt,qwmax,profile	2*real,integer	$rwdt$ is the range of the resolution in FWHM units, normally $rwdt > 3.5$; $qwmax$ - maximum expected Q resolution range at $Q(nq_2)$ [\AA^{-1}]; $profile$ is the index of the resolution profile function, $0 < profile < 6$, cfm. 5.7.2 for definitions.
wave, zero, thmon	2*real or 3*real	wave – for $x_{in} = 0$ or 1 the incident wavelength, for $x_{in} = 2$ the corresponding for TOF; zero - the x experimental data offset; thmon – the monochromator θ Bragg angle, only read if $profile = 4$.

cwave, czero	2*integer	codes for refining wave and zero.
wave2, zero2, wgt2	3*real	wave2 – a 2 nd wavelength; zero2 – 2 nd offset; wgt2 – weight of a 2 nd wavelength component, only read if profile = 3.
cwave2, czero2, cwgt2	3*integer	codes for refining wave2, zero2 and wgt2, only read if profile = 3.
u,v,w,ig or u,v,w,ig,eta or u,v,w,ig,dthi or a0tofn,a1tofn, b0tofn,b1tofn, s0tofn,s1tofn, s2tofn	4*real or 5*real or 5*real or 7*real	Resolution FWHM parameters [degrees ²]. u,...,ig – if profile = 1, u,...,eta – if profile = 2 or 3, u,...,dthi – if profile = 4, a0tofn,a1tofn,... – if profile = 5, see 5.7.2
cu,...,cig or cu,...,ceta or cu,...,cdthi or ca0tofn,..., cs2tofn	4*integer or 5*integer or 5*integer or 7*integer	Codes for refining resolution parameters.
wext	integer	if wext>0 a file with additional resolution widths is read, see 5.7.2
wextfile	character*80	File name for additional resolution widths [degrees]. Q values must be the same as in the experimental data file.

4.4.15 Nuclear scattering parameters repeated for each particle type

noccn	integer	No of atom types for this particle type. $0 \leq \text{noccn} \leq 5$. noccn = 0 indicates a vacancy (zero-scattering) type.
coccn,bcoh, siginc or natsym,coccn	3*real or character,real	One block repeated for each of the noccn types. coccn – occupancy of this atom type; bcoh – coherent neutron scattering length [fm]; siginc – incoherent neutron cross-section [barns]; natsym – atomic symbol (scattering parameters read from table).

4.4.15.1 Parameters for magnetic neutron scattering.

x2m	real	The maximum x for which magnetic diffuse scattering will be calculated. If $x2m \leq 0$ no magnetic scattering is calculated.
betam,cbetam	real,integer	betam – an overall scale factor for the magnetic scattering, relative to the model nuclear scattering; cbetam – code for refining betam.

4.4.15.2 Magnetic scattering parameters repeated for each of the ntypesm magnetic particle types (only read if $q2m > 0$)

noccm	integer	noccm – No. of magnetic types for this particle type, $0 \leq \text{noccm} \leq 5$.
magsym,coccm, mueff or coccm,mueff	character,2*real or 2*real	magsym – magnetic symbol (scattering parameters read from table). This and next block (if no magsym) repeated for each of the noccm types. coccm – occupancy of this magnetic type; mueff – effective magnetic moment [μ_B].
formm	real array	Magnetic form-factor parameters (a1,b1,...,a4,b4,c). Not read if

		there is a magsym entry on previous line
--	--	--

4.4.16 Parameters repeated for each x-ray experimental data set

The same as section 4.4.14 but with the corresponding x-ray parameters, then

compton	integer	if <code>compton</code> >0 Compton scattering will be added to the model x-ray scattering, see section 3.4
---------	---------	--

4.4.16.1 X-ray scattering parameters repeated for each particle type

noccx	integer	No. of atom types for this particle type. $0 \leq \text{noccx} \leq 5$.
xatsym,coccx, zeff or coccx,zeff	character,2*real or 2*real	xatsym – atomic symbol (scattering parameters read from table). This and next block (if no xatsym) repeated for each of the noccx types. coccx – occupancy of this atom type; zeff – effective number of electrons for Compton scattering.
formx,dfp,dfpp	real array	X-ray form-factor parameters (a1,b1,...,a4,b4,c); dfp,dfpp - the real and imaginary dispersion corrections to the x-ray scattering form-factor.

4.4.17 Parameters for linear constraints of non-linearly refined parameters

nlc	integer	No. of linear constraints.
nterms, ccoeff1,cindex1, ccoeff2,cindex2, ...	integer, 2*nterms*real	nterms – no. of terms in the linear constraint; ccoeffi and cindexi – coefficients and parameter codes for each term. This line is repeated for each of the nlc constraints. See 5.7.

4.5 The experimental data files

The files containing the experimental data are in the DATA format as defined for the NDP series of programs. The number of points is given on the first line, the second line contains a title or other information, and the subsequent lines contain the x , $I(x)$ or x , $I(x)$, $err(x)$ values in two or three columns, where x is Q , 2θ or TOF as determined by the parameter `xin`, cf. 4.4.14. Data must be given at increasing x values, but do not need to be at equidistant points. An example data file is included in the package.

Note that if more than one set of diffraction data is supplied, unlike in the RMCA program, the sets do not need to be defined at the same x values, and they can cover different ranges (x types can actually be different for each experiment).

The experimental data should normally be fully corrected for absorption, multiple scattering and Placzek effects (in the case of neutrons) and polarisation. Data must also be normalised with respect to angle-dependent efficiencies etc., so that they truly (apart from a scale factor and background, see 5.7.1) represent the differential scattering cross-section per atom in barns (neutrons) or electron units (x-rays), i.e. with dimensions of area. Data, which are not reduced in this way, *must* be modified accordingly since a quantitative analysis of the diffuse scattering is intended. However, in some cases a rough absorption correction can be sufficient for at least approximate diffuse scattering modelling. To facilitate this the following transmission factor has been implemented

$$T(\theta) = \exp\left[-\left(1.7133 - 0.368\sin^2(\theta)\right)mur + \left(0.0927 + 0.375\sin^2(\theta)\right)mur^2\right] \quad (18)$$

and can be controlled by the macroscopic absorption parameter mur , cfm. 4.4.14 .

4.6 Example files

Three sets of example files are supplied which demonstrate the use of the program and which can be used for testing. Extract the compressed files into a suitable directory and follow the instructions in the PDF document included with each example. The examples are:

mno.zip

The average magnetic structure of MnO in the long-range ordered antiferromagnetic phase at 15 K will be determined. Investigation of the short-range magnetic order at 130 K, i.e. just above the Neel temperature $T_N=120$ K can also be done

bafewo.zip

The average magnetic structure of Ba₂FeWO₆ in the long-range ordered antiferromagnetic phase at 15 K will be determined starting from a random configuration.

agbr.zip

The long-range and local order of crystalline AgBr will be modelled by a simultaneous fit to the (thermal) diffuse and Bragg scattering. Correlation of atomic displacements of atoms with small separation will be studied and compared to the long-range limit results.

For more examples on the use of RMCPOW see [10]

4.7 Other programs available

There are two other Monte Carlo-based codes available at NFL. RMCA is the general program for modelling atomic configurations and MCGR models total or partial pair correlation functions to fit scattering data. There are many programs available in the NFLP suite for display and analysis of the results produced by RMCPOW, and for the creation and modification of configurations. Some of the NFLP suite programs are convenient for preparing experimental data sets in the correct format for input to RMCPOW. All of these programs are documented separately.

5. RMCPOW - simulation details

5.1 Configuration size and shape

When starting from the initial configuration the amplitude sums must be calculated. This involves a summation of order N^2 . However for each particle move it is only necessary to calculate the change in the amplitude sums corresponding to the moved particle, which is a summation of order N . Since the time for calculating the initial structure factor, and single moves, is much larger for RMCPOW than RMCA some considerations has to be made with regards to the size. Generally we use $N > 1500$, and have used $N = 20000$.

The size of the configuration cell is also important because it determines the major part of RAM memory needed during the simulation. The reciprocal volume per used τ_{cc} is $V_{cc}^* = 8\pi^3/V_{cc}$ where V_{cc} is the volume of the configuration cell. If the maximum Q being modelled is Q_{max} then the number of configuration cell reciprocal points with $Q < Q_{max}$ is

$$N_{\tau}(Q_{max}) = \frac{4\pi Q_{max}^3/3}{8\pi^3/V_{cc}} = \frac{V_{cc} Q_{max}^3}{6\pi^2} \quad (18)$$

A typical model configuration might consist of 5000 atoms with an atomic density 0.08 \AA^{-3} so that $V_{cc} \sim 6 \cdot 10^4 \text{ \AA}^3$ (i.e. linear dimensions are $\sim 40 \text{ \AA}$). At reactor based diffractometers $Q_{max} \sim 10 \text{ \AA}^{-1}$, giving $N_{\tau} \sim 10^6$. Inversion symmetry of the reciprocal lattice reduces the number of points that actually have to be computed and stored by a factor of 2. For updating the amplitudes a copy of each (real and imaginary) sum is used so that the memory required for such a simulation is $>16 N_{\tau}$ bytes (assuming a single precision size of 8 bytes for reals).

The basic entity for building a crystalline model is the unit cell. The shape or metrics of this cell is given by the cell edge and angle parameters or, equivalently, its three basis vectors. The unit cell may be primitive or centred. In addition to simply building configurations as multiples along each basis vectors we can use a generalised approach where in principle any parallelepiped, containing the same periodicity as the primitive cell, can be used as building blocks for the model. For this purpose we define an intermediate cell (IC) being a supercell of the primitive cell but not necessarily the conventional unit cell. The whole configuration is then constructed as multiples of IC:s in the traditional way. Transformation matrices conveniently describe relations between the various cells used and we give below the definitions used.

isym $> 0 \rightarrow$ the IT conventional unit cell (UC) is used as IC

isym = 1 \rightarrow Primitive cell

isym = 2 \rightarrow A centred cell

isym = 3 \rightarrow B centred cell

isym = 4 \rightarrow C centred cell

isym = 5 \rightarrow I centred cell

isym = 6 \rightarrow F centred cell

isym = 7 \rightarrow Rhombohedral cell in hexagonal setting

isym = 8 \rightarrow Hexagonal primitive cell

isym $< 0 \rightarrow$ A non-standard IC is generated from the corresponding UC by a transformation matrix P^{UI}

The transformation matrix $P^{UI} = (p_{ij})$ is defined by the relations between the two sets of basis vectors \mathbf{a}_i and \mathbf{A}_i , where $i=1,2,3$, for the UC and IC respectively according to

$$\mathbf{A}_i = p_{1i}\mathbf{a}_1 + p_{2i}\mathbf{a}_2 + p_{3i}\mathbf{a}_3 \quad (19)$$

For computational purposes a common least denominator is taken out of the matrix and given separately. E.g., to convert a UC to its C-centred supercell we can take

$$\mathbf{A}_1 = \mathbf{a}_1 + \mathbf{a}_2, \quad \mathbf{A}_2 = -\mathbf{a}_1 + \mathbf{a}_2, \quad \mathbf{A}_3 = \mathbf{a}_3 \quad (20)$$

so that the common denominator is 1 and the transformation matrix is

$$P^{UI} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (21)$$

i.e. $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3) = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3) P^{UI}$. The same formalism can of course also be used to construct an IC smaller than the UC. E.g. to construct the rhombohedral primitive cell of a face centred cell we instead use

$$\mathbf{A}_1 = \frac{1}{2}(\mathbf{a}_1 + \mathbf{a}_2) \quad , \quad \mathbf{A}_2 = \frac{1}{2}(\mathbf{a}_2 + \mathbf{a}_3) \quad , \quad \mathbf{A}_3 = \frac{1}{2}(\mathbf{a}_3 + \mathbf{a}_1) \quad (22)$$

so the denominator is 2 and

$$P^{UI} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad (23)$$

For visualisation of a configuration it must be related to a Cartesian coordinate system. Also, spin calculations are easier with Cartesian coordinates. Various RMC and USEFUL programs use the following definitions: ($\hat{\mathbf{A}}_i$ and \mathbf{x}_i are unit vectors of the IC basis and Cartesian systems resp.)

icart = 1 $\rightarrow \hat{\mathbf{A}}_1 = \mathbf{x}_1$, $\hat{\mathbf{A}}_2 \perp \mathbf{x}_3$, $\hat{\mathbf{A}}_{22}, \hat{\mathbf{A}}_{33} > 0$, default for isym=1 to 6

icart = 2 $\rightarrow \hat{\mathbf{A}}_2 = \mathbf{x}_1$, $\hat{\mathbf{A}}_3 \perp \mathbf{x}_3$, $\hat{\mathbf{A}}_{23}, \hat{\mathbf{A}}_{31} > 0$

icart = 3 $\rightarrow \hat{\mathbf{A}}_3 = \mathbf{x}_1$, $\hat{\mathbf{A}}_1 \perp \mathbf{x}_3$, $\hat{\mathbf{A}}_{21}, \hat{\mathbf{A}}_{32} > 0$

icart = 4 $\rightarrow \hat{\mathbf{A}}_1 = \mathbf{x}_1$, $\hat{\mathbf{A}}_3 \perp \mathbf{x}_2$, $\hat{\mathbf{A}}_{22}, \hat{\mathbf{A}}_{33} > 0$, default for isym=7 to 8

icart = 5 $\rightarrow \hat{\mathbf{A}}_2 = \mathbf{x}_1$, $\hat{\mathbf{A}}_1 \perp \mathbf{x}_2$, $\hat{\mathbf{A}}_{23}, \hat{\mathbf{A}}_{31} > 0$

icart = 6 $\rightarrow \hat{\mathbf{A}}_3 = \mathbf{x}_1$, $\hat{\mathbf{A}}_2 \perp \mathbf{x}_2$, $\hat{\mathbf{A}}_{21}, \hat{\mathbf{A}}_{32} > 0$

5.2 Random moves

The maximum size of the random move, **delta**, determines the ratio of accepted to rejected moves, but also determines the amount that the structure may change with each move. If **delta** is too small then nearly all moves will be accepted but the structure will change little, while if it is too large then few moves will be accepted and the average structural change will also be small. If we attempt to choose **delta** such that the ratio of accepted to rejected moves is approximately 1:1, as is often done in MMC, this usually leads to a value **delta** < 0.1 Å. The average structural change per move is usually maximised for $0.05 < \text{delta} < 0.2$ Å, so this range is normally used. If the packing fraction (ratio of excluded volume to total volume) is high, as in crystalline materials, then it is necessary to use small moves so that a sufficient number are accepted. The convergence of the RMCPOW procedure is therefore very slow.

In many systems a tendency for long-range or local ordering of two or more atomic types to occupy a particular lattice site is observed, e.g. substitution of two metallic types on a fcc lattice. Since it can be almost impossible to achieve such ordering from an initial random distribution using just ordinary displacive moves, two options to allow swap moves is given in the program.

In the first option, “swap moves”, the identities of two atoms of different types are interchanged. If one of the selected atoms also carries a magnetic moment then this spin is also moved, but if both are magnetic only the moduli of the moments are interchanged whereas the orientations are maintained at the original positions. Swap moves can also be used to interchange atoms of a particular type with vacancies (which then are defined as a particle type with zero scattering properties and their positions included in the configuration). Note that vacancies do not produce any change in the model scattering cross-section for *displacive* moves. The only effective constraint on vacancy positions is then any closest-approach constraints so **delta** should be kept at zero for vacancies.

In the second option, “reference moves”, the first atom is moved to the (reference) position of the second atom. If the first atom carries a magnetic moment, then this spin is also moved. Additionally, the first atom may be given a random displacement with respect to the reference position, as determined by parameter **deltaref**. The second atom is always maintained at its original position (and possibly spin orientation), thereby acting as a fix reference. Due to the character of this type of move

the second type is naturally of the vacancy kind. This option is particularly useful when atoms show a fractional occupation of certain crystallographic sites, without apparent long-range order but possibly some short-range order.

5.3 Constraints

Other information that cannot be used directly can be made use of in the form of constraints; this may include NMR, EPR, Raman scattering and chemical knowledge. The most commonly used constraint in RMCPOW is on the closest distance of approach of two atoms. Other constraints that can also be used in the current version of RMCPOW are on the co-ordination number of atoms, bond valencies and average magnetisation of spins.

5.3.1 Minimum distance of atom pairs

For perfect data realistic values for the closest approach distances of pairs of atoms can often be determined from the low r cut-off in $g^E(r)$, obtained from direct Fourier transformation of the measured total structure factors. Because of systematic errors in the experimental data, and often because of the limited data range, the data would not forbid some atoms from coming very close together. However we know that this is physically unrealistic so an excluded volume is defined. For this reason it is usually sensible to specify allowed distances of closest approach, in other words to define an excluded volume. This also saves considerable time since moves which would result in atoms being too close together can be rejected before calculation of the change in amplitudes. For good data the specified closest approaches may be somewhat lower than realistic values but for poor data they need to be more carefully chosen. If the values are too large then this is usually apparent because the resulting $g^C(r)$ has a sharp cut-off instead of decreasing more gradually to zero. If they are too low $g^C(r)$ may have a sharp spike in the low r region.

While this is a very simple constraint on the structure it is also very powerful, since the imposition of both an excluded volume and a fixed density restricts possible configurations. One could also view it as the imposition of a hard sphere repulsive potential. For many materials the dominant effect determining the structure is packing, and hence to implicitly include information on atomic sizes in the model (these are minimum sizes rather than, for example, ionic radii) severely limits the number of structures that are consistent with the data.

5.3.2 Co-ordination constraints

Co-ordination constraints are one of the most valuable and instructive 'tools' used in RMC methods. A co-ordination number $N_{\alpha\beta}$ is defined as being the number of atoms of type β between two fixed distances of one of type α . Normally the lower fixed distance is the closest distance of approach of the two types of atom (or equivalently zero). In the RMCPOW program the concept of co-ordination constraints has been slightly generalised to allow constraints over a *range* of co-ordination numbers. If we define the proportion of atoms of type α in the configuration with a particular range of co-ordination numbers as f_{RMC} and the desired proportion with such a range as f_{req} then we can add an additional term to χ^2 :

$$\chi^2 = \dots + (f_{req} - f_{RMC})^2 / \sigma_c^2 \quad (24)$$

Obviously multiple co-ordination constraints can be applied by adding additional terms. The parameter σ_c , in this case simply acts as a weighting of the co-ordination constraint relative to the data. If $\sigma_c \approx 0$, e.g. 10^{-6} , it is effectively impossible for atoms with the constrained co-ordination to change it; this can be used to mimic the effect of covalent bonding. In many cases hard sphere Monte Carlo simulation with such co-ordination constraints, that is RMC with no data, can be used to produce structures with suitable topology prior to fitting the data.

The above refers to constraints on co-ordination numbers of individual (although unspecified) atoms. It is also possible to constrain the average coordination number, in which case a term

$$\chi^2 = \dots + \left(\text{avc}_{req} - \langle c \rangle \right)^2 / \sigma_{avc}^2 \quad (25)$$

is added to the total χ^2 where $\langle c \rangle$ and avc_{req} are the actual and desired average coordination numbers resp.

5.3.3 Bond valence constraints

Bond valence sums [11] are used in RMCPOW to constrain the bond valence of one or more particle types. For each bond valence constraint a term, summing over all particles j of a type,

$$\chi^2 = \dots + \sum_j \left(\text{val}_{req} - v_j \right)^2 / \sigma_{val}^2 \quad (26)$$

is added to the total χ^2 . The bond valence sum for particle j is calculated either as

$$v_j = \sum_i \left(R_0 / R_i^j \right)^N \quad (27)$$

or

$$v_j = \sum_i \exp \left(\frac{R_0 - R_i^j}{B} \right) \quad (28)$$

where the sum runs over all bonds with $R_i^j < \text{rval}$. Depending on which expression(s) that are used (determined by parameter `ival`) the relevant parameters, R_0 and B or R_0 and N , should be given in the control data file. Valence constraints are very useful for modelling e.g. structures with high degree of disorder due to vacancies. It is also possible to constrain the average bond valence, in which case a term

$$\chi^2 = \dots + \left(\text{aval}_{req} - \langle v \rangle \right)^2 / \sigma_{aval}^2 \quad (29)$$

is added to the total χ^2 , where $\langle v \rangle$ and aval_{req} are the actual and desired average bond valencies resp.

5.3.4 Magnetisation constraints

According to quantum mechanics the average magnetisation μ_z (per spin) along any axis z cannot exceed gS , S being the single-ion spin quantum number. For a simple ferromagnet this would be the expected zero-point average, and decrease with increasing temperature. Although RMCPOW uses a classical approach to spins, we have found that, for simple magnetic systems, we normally end up with magnetic configurations exhibiting approximately the expected zero-point and temperature dependence of the average magnetisation. Since in these cases also the diffuse magnetic scattering is very well modelled we believe that RMCPOW can be used as an independent tool for determining average magnetisations. Nevertheless it might sometimes be useful to constrain the model to some specific value, e.g. to zero for an antiferromagnetic structure or to agree with the temperature-dependence obtained from susceptibility data. If the calculated modulus of the average magnetisation for some magnetic particles is μ_{RMC} this is effectuated by adding a term

$$\chi^2 = \dots + \left(\mu_{req} - \mu_{RMC} \right)^2 / \sigma_m^2 \quad (30)$$

where μ_{req} is the required magnetisation and σ_m is an effective weight. Each magnetic particle type can have its own constraint. Note that this is a constraint for the modulus of the average 3-component magnetization vector, in contrast to using the thermal models (`5>ijob>1`) where only a z component is considered.

5.3.5 Spin-spin co-ordination constraints

A magnetic analogue of the atomic co-ordination number is the the spin-spin cosine of neighbours. This number can be used in two ways in RMCPOW to constrain local magnetic environments. If we define the proportion of atoms of type α in the configuration with a particular range of (atomic) co-ordination numbers, *and* each of these neighbours with the spin-spin cosine in some range, as f_{RMC} and the desired proportion with such a range as f_{req} then we can add an additional term to χ^2 :

$$\chi^2 = \dots + (f_{\text{req}} - f_{\text{RMC}})^2 / \sigma_c^2 \quad (31)$$

The average spin-spin cosine can also be constrained in analogy with average co-ordination numbers.

5.3.6 Magnetic potential constraints

This constraint utilizes a standard MC procedure for minimising the energy of Heisenberg classical spins. The Hamiltonian used is then

$$H = \frac{1}{2} \sum_{ij} J_{ij} \mathbf{S}_i \mathbf{S}_j \quad (32)$$

where the \mathbf{S}_i are unit vectors, so the exchange parameters J_{ij} (Jmpot in the control file) should include the spin modules.

5.4 Q spacing and range

It might be considered that the formal real space resolution, given by $2\pi/Q_{\text{max}}$, Q_{max} being the maximum experimental Q , severely limits the usefulness of RMCPOW for modelling to e.g. reactor data. Since we are not relying on a Fourier inversion of $I(Q)$ this is however not really relevant. We have found that details down to at least 0.2 Å can be distinguished using the RMCPOW algorithm and it should be noted that we are actually fitting a physical model similar to e.g. the way Rietveld refinement is used. We recommend however that the Q range be extended as far as possible, at both ends (particularly at low Q for magnetic modelling). The Q step should be chosen appropriately with respect to the experimental resolution, i.e. about five points over the sharpest features of the experimental data. Note that the Q mesh, used internally for reducing the 3D reciprocal lattice to one dimension *is* defined at equidistant points as give by parameter `dqp`, cfm. 4.4.13.

The minimum Q value that can be modelled is given by $Q_{\text{min}} = 2\pi/L$, where L is the longest edge of the configuration cell. If you try to fit to smaller Q values then the effects are unpredictable. For example, if a much smaller Q_{min} is used then this can lead to distinct density fluctuations of period $2\pi/L$ in the configuration. Note that weak fluctuations of this period can be seen in many simulations, not just in RMC.

5.5 Smoothing the diffuse data

Due to the discrete nature of the way diffuse scattering is computed in reciprocal space it is necessary to smear this contribution over a corresponding linear Q extension in order to approximate the experimental data. The intensity of each diffuse Q point is then simply uniformly distributed over a range of length ΔQ_{av} around Q . ΔQ_{av} is taken as the approximate linear size $2\pi/V_{\text{cc}}^{1/3}$ of a reciprocal cell.

In order to further smooth the model diffuse scattering, and hence to avoid fitting the statistical errors, it is also possible to apply Savitsky-Golay smoothing before the experimental resolution is convoluted. The Q width of the smoothing interval is then determined by the parameter `swdt` as

$\text{swdt} \cdot 2\pi/V_{\text{uc}}^{1/3}$, i.e. as the swdt fraction of the linear size of a cell of the reciprocal lattice of the *unit* cell. We have found that in general a value of $\text{swdt} \leq 0.3$ is appropriate.

5.6 Experimental error

The RMCPOW algorithm assumes that we have only statistical errors. These can be a) read with the experimental data or b) estimated as proportional to the square root of the data or c) taken as constant. A real experimental structure factor $I^E(Q)$ will contain both statistical and systematic errors, but the whole procedure is not thereby invalidated. A three dimensional structure that is consistent with the experimental data within some measure of the error can still be produced, though this measure is now less well defined.

While one might expect statistical errors to be small where $I^E(Q)$ is large, and vice-versa, in practice the requirement to perform container and background corrections in many experiments means that statistical errors are often quite uniformly distributed. For these cases it is usually simplest to assume a constant value of σ at all Q , though σ may differ between different data sets. However there have been cases in which large values of σ have been used in particular Q ranges where it is known that there were errors in the data. By setting $\sigma(Q)$ at an extremely large value these data points can effectively be ignored. From the above discussion it is clear that the precise value of σ is not known in any particular case; it may therefore be considered as a parameter of the simulation.

5.7 Refinable parameters

Experimental data will normally contain small errors in the form of multiplicative and additive constants. It is possible to take account of such errors within the RMCPOW algorithm. During the RMCPOW process, the real experimental total structure factors, $I^E(Q)$, can be considered as simple transformations of the structure factors actually measured, $I^{E,m}(Q)$:

$$I^E(Q) = \frac{1}{\beta} \left(I^{E,m}(Q) - \sum_{l=0}^{n_B} \alpha_l Q^l \right) \quad (33)$$

where n_B is the degree of a background polynomial. In addition the calculated structure factor can be dependent of a number of other parameters p_1, p_2, \dots , e.g. resolution parameters. The expression that is actually minimised in the simulation procedure is then accordingly (for the n th experiment)

$$\chi_n^2 = \sum_{i=1}^{m_n} \left(\beta I_n^C(Q_i, p_1, p_2, \dots) + \sum_{l=0}^{n_B} \alpha_l Q_i^l - I_n^{E,m}(Q_i) \right)^2 / \sigma_n(Q_i)^2 \quad (34)$$

The multiplicative and additive parameters can be refined using a linear least-squares fit and so can be done, with a minor cost of computing time, after each attempted move (atomic positions and spin orientations are refined by the Monte Carlo procedure and are considered as fix parameters in this minimisation step). The other parameters in general require a non-linear approach and are therefore more expensive. Full non-linear fits are therefore performed only at each *iprint* attempted move, and inbetween only linear parameters are refined. The non-linear fit is a modified version of the Marquardt-Levenberg algorithm based on the code in Numerical Recipes [12].

During the non-linear minimisation step it is possible to apply linear constraints on the various parameters. Using the refinement codes, *cindex*_{*j*}, the *i*th constraint can be given in the general form

$$\sum_{j=1}^{n_p} \text{ccoeff}_{i,j} \text{cindex}_j = \text{constant} \quad (34)$$

where n_p is the number of refinable parameters, though only terms containing non-zero *ccoeff* has to be given. The actual values of the codes *cindex* are just labels and can be anything in the range

[0,100], the value 0 however indicating a fix parameter. Two or more codes can have the value 1, indicating that they all correspond to free and independent parameters. No two parameters can have the same code > 1, such values being reserved for constrained parameters.

5.7.1 Renormalisation and background

The application of linear constraints permits the refinement of normalisation factors (β) and sloping backgrounds (α_0, \dots) in $I_m^E(Q)$. Optimisation of any combination of the parameters is possible. It is recommended, however, that the normalisation factor (β) be refined only when the best possible fit without changing it could be achieved. The values of the above parameters, together with the χ^2 value, provide a sensitive test of data quality, particularly in terms of systematic errors. For an experimental structure factor reduced to correct absolute units [barns or electron units] β should be 1, and $\alpha_i = 0$.

In addition to a polynomial background it is also possible to read supplementary data files containing fix, Q -dependent backgrounds, $I_{fix}(Q)$. Since such a background can be adjusted with a refinable scale factor, **betaf**, it can be used to e.g. subtract a secondary phase with known structure but unknown concentration. An additional term **betaf** $\times I_{fix}(Q)$ is then added in eq. 29. Such background data must be defined at the same Q bins as the experimental data.

Concerning magnetic neutron scattering the effective moments, **mueff**, effectively determine the particle magnetic scattering strengths and should, similar to nuclear scattering lengths, be kept constant during the simulation. The application of an overall magnetic scale factor, βm , is however often practical in order to obtain a proper amount of model magnetic scattering. βm is here defined relative to the total scale factor β so the expected value is 1 (if moments do take on their nominal values) regardless of β . Note also that the magnetic scattering goes as **mueff**² and so the nominal moments scale as $\sqrt{\beta m}$.

5.7.2 Experimental resolution

One of the most important features of the RMCPOW program is the ability to include and fit the experimental resolution. In the standard RMC approach, where a model $g(r)$ is computed and Fourier transformed to be compared with experimental data, it is not possible to account for the resolution in a straightforward way. With the RMCPOW method the calculated structure factor can be separated into Bragg and diffuse contributions and both contributions can easily be convoluted with the experimental resolution.

In principle any of the standard resolution functions in use for Bragg profile refinements could be applied also in the RMCPOW algorithm. The current version however only includes four versions for CW data and one for TOF data. The CW profiles are

1. The Gaussian:

$$I(2\theta, 2\theta_0) \propto \frac{1}{H} \exp\left(-4 \ln 2 \left(\frac{2\theta - 2\theta_0}{H}\right)^2\right) \quad (31)$$

2. The Lorentzian:

$$I(2\theta, 2\theta_0) \propto \frac{eta}{H} \left(1 + 4 \left(\frac{2\theta - 2\theta_0}{H}\right)^2\right)^{-1} + \frac{(1 - eta)}{H} \exp\left(-4 \ln 2 \left(\frac{2\theta - 2\theta_0}{H}\right)^2\right) \quad (31)$$

3. The splitted Lorentzian, as 2. but with terms for a 2nd wavelength.
4. A Gaussian convoluted with a rectangular distribution, as occuring on instruments with no or coarse Soller collimation:

$$I(2\theta, 2\theta_0) \propto \frac{1}{2W} \left\{ \operatorname{erf} \left[2\sqrt{\ln 2} \left(\frac{2\theta - 2\theta_0 + W/2}{H} \right) \right] - \operatorname{erf} \left[2\sqrt{\ln 2} \left(\frac{2\theta - 2\theta_0 - W/2}{H} \right) \right] \right\} \quad (31)$$

where $W = d\theta_i \cdot [2 \tan \theta / \tan \theta_{mon} - 1]$, $d\theta_i$ being approx. the monochromator incident divergence and θ_{mon} the monochromator Bragg angle.

Parameters determining the peak width can be refined as desired but it is recommended to keep them fixed at the nominal instrumental resolution and only allow variation at the end stage of a simulation. The expression used for the 2θ FWHM is

$$H^2 = U \tan^2 \theta + V \tan \theta + W + IG / \cos^2 \theta \quad (30)$$

where parameters U , V , W and IG are given in units of degrees squared. The range (in FWHM units) over which each Bragg or diffuse point is calculated is determined by the parameter $rwtd$.

Finally, the sole TOF profile (so far) is

5. the TOF profile function 1 in the GSAS manual [13]:

$$I(TOF, TOF_0) \propto N [e^u \operatorname{erfc} y + e^v \operatorname{erfc} y] \quad (31)$$

where N, u, v and z are dependent on $TOF - TOF_0$ and the instrumental width parameters αp_0 , αp_1 , β_0 , β_1 , σ_0 , σ_1 and σ_2 (some parameters in the GSAS definition are neglected here).

Note that thermal diffuse scattering in general occurs as broad peaks below the Bragg peaks. Such diffuse features could easily be incorporated into the Bragg peaks in traditional powder refinements, where diffuse scattering is not considered. The Bragg peaks would then appear to have a more Lorentzian shape and increasing widths with increasing temperature. Thermal parameters would also be affected since too much intensity is assigned to Bragg scattering. Since RMCPOW includes modelling of both thermal and static diffuse scattering resolution parameters should in principle not be affected by these considerations. From these arguments it also follows that if crystallites sizes are small so that there is a significant broadening then it can be difficult to distinguish thermal/static disorder from size effects, when using the RMCPOW method.

Since the scattering amplitudes are not dependent on the resolution these parameters can be changed at any stage of a simulation. However in order to get a well-defined value of the calculated scattering at Q_{max} , the maximum Q fitted, the intensity of peaks in some range $qwmax$, determined by the resolution, outside Q_{max} has to be included. So that it should not be necessary to calculate new amplitude sums during the simulation, the maximum expected $qwmax$ is read from `name.dat` rather than being estimated from the resolution parameters. Normally, for $Q_{max} \approx 10 \text{ \AA}^{-1}$, $qwmax > 0.5 \text{ \AA}$.

As a further possibility to determine the resolution, a separate file with FWHM's can be read by setting `wext > 0`. These widths [degrees] are then added (squared) to the Cagliotti expression (eq. 30).

5.7.3 Other refinable parameters

For the `ijob = 2` or `3` modes, see 3.6 and 4.4.13, it is possible to refine the mean square displacements, parameters `u2`, and average ordered components of magnetic moments, parameters `muz`. For `ijob = 3` the Debye temperature `TD` can also be refined. Future versions of RMCPOW are planned to include lattice parameter refinement.

References

[1] RMCA Version 3, R.L. McGreevy, M.A. Howe and J.D. Wicks, (1993), available at <http://www.studsvik.uu.se>

- [2] Reverse Monte Carlo modelling of neutron powder diffraction data. A. Møllergård and R.L. McGreevy. *Acta Cryst. A* 55 (1999) 783
- [3] V.F. Sears, *Neutron News*, Vol.3, No. 3 (1992) 29
- [4] E.J. Lisher and J.B. Forsyth, *Acta Cryst A* 27 (1971) 545
- [5] P. J. Brown, Magnetic formfactors, chapter 4.4.5 in *International tables for crystallography* vol. C (A. J. C. Wilson, ed.), pp. 391-399
- [6] *International Tables of Crystallography*, volume C
- [7] B.E. Warren, *X-ray Diffraction*, Reading, Mass: Addison-Wesley (1969)
- [8] P. Suortti, *Ann. Acad. Sci. Fenn. A6*, 240 (1967) 1
- [9] PGPLOT Copyright © 1983-2001 by the California Institute of Technology. <http://astro.caltech.edu/~tjp/>
- [10] A. Møllergård, R.L. McGreevy, A. Wannberg, and B.Trostell. *J. Phys.: Cond. Matter.* 10 (1998) 9401
- [11] I.D. Brown, in *Structure and Bonding in Crystals*, vol.II, eds. M. O'Keeffe and A.Navrotsky, Academic Press, New York, USA, pp 1-30.
- [12] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes (Fortran)* (1989), Press Syndicate of the University of Cambridge, Cambridge, USA
- [13] Allen C. Larson & Robert B. Von Dreele, *GSAS Technical Manual*, Los Alamos National Laboratory