RMC_POT user guide for version 2023.1

Orsolya Gereben

25.04.2023

I. The Reverse Monte Carlo algorithm

The Reverse Monte Carlo (RMC) computer simulation technique is capable of building 3-dimensional structural models in agreement with the experimental (mainly diffraction) data. Only a very brief description of the algorithm is given below, more details can be found elsewhere[1],[2],[3]. Several computer version of RMC exist, one of the newest implementation (RMC++)[4] was written in C++ (RMC++_new) and was parallelized and improved with new capabilities (RMC++_multi)[5]. These later software were used as the starting point of the present development, RMC_POT. Here the usage of the program will be described.

The starting point for RMC_POT was the unification of the codes RMC_new and RMC_multi codes. Now compiler options regulate the compilation of the code. Parallelization was achieved first by using the POSIX standard (which can also be used under Windows) and can make execution faster on multiprocessor computers. Later this was changed to C++11 standard, but the program can run only with one thread as well. New features were also sometimes depending on the compilation, as for example introducing artificial neural network potential usage, local invariance calculation, non-periodic boundary conditions, advanced geometric constraints and vibrational amplitude calculation.

The RMC_POT program was written in C++, using only the standard statements as far as it was possible. There were however few cases, where operation system-specific statements have to be used, these are located in *altern.h* and *altern.cpp* files. The program was tested on Windows and Linux platform, if other platform is used, some alteration might be necessary!

RMC can fit neutron [later on denoted as *S(Q)*], X-ray [*F(Q)*] and electron diffraction [*F(g)*] structure factors, X-ray intensity [*I(Q)*] and EXAFS data [*E(k)*]. It can also fit partial and total *g(r)* functions as well, but it is discouraged to transform the experimental data and use this for *g(r)* fitting, this feature is recommended only if the *g(r)* is coming from MD simulation to test the validity of a force field additionally to diffraction data.

A. The brief description of the RMC algorithm

The idea of the RMC algorithm is the following. We have a cubic simulation box containing fixed number of atoms having number density $\rho=N/V$. The box dimensions are between -1 →+1 in reduced units in each direction. From this initial configuration the histogram can be calculated. The count in the kth histogram bin is the number of atoms between $k*dr$ and $(k+1)*dr$ from the central atom, where dr is the size of the histogram bin. The RMC histogram only contains counts coming from unique atoms pairs to save time during the calculations, so the pure partial RMC histogram has half of the counts of a normal histogram. Let's consider a multi-component system. Here partial functions have to be defined between the atom types. The number of the partials is $ntypes\cdot(ntypes+1)/2$ (where *ntypes* is the number of atom types), so the mixed partials are just defined once. Order of the partials is row-continuous following the order of the elements in an upper diagonal matrix: (for example for 3 types: 1-1, 1-2, 1-3, 2-2, 2-3, 3-3). The order of the RMC atom types is defined by the order they are given in the configuration file. The partial pair correlation function (ppcf) which can be called radial distribution function (prdf) as well, can be calculated from the partial histograms by normalization:

$$g_{ij}(r)=\frac{n_j(r)}{4\pi r^2\Delta r\rho_j} \qquad \text{E 1}$$

where n_j is the number of atoms of type j at a distance between r and $r+\Delta r$ from a central atom of type i, averaged over all atoms of the central type i. ρ_j is the number density of the type j atoms.

The partial structure factor can be calculated by Fourier transformation from the *prdf*:

$$S_{ij}^{RMC}(Q)=S_{ij}(Q)-1=\frac{4\pi\rho}{Q}\int_0^\infty r\left[g(r)_{ij}-1\right]\sin(Qr)\,dr \qquad \text{E 2}$$

As the neutron scattering length does not depend on scattering vector, *Q*, the neutron diffraction will be used to demonstrate the RMC algorithm. In case of the normal X-ray diffraction and electron diffraction the atomic scattering factors (or form factors) *f(Q)* and *f(g)* are *Q* or *g*-dependent, so the formulas below should be modified accordingly. From version 2.0 there is a possibility that in case of X-ray diffraction not only the *Q*-dependent part, *f(Q)* of the scattering factor could be used but the real part of the *E*-dependent component, *f'(E)* as well. This mostly has meaning in case of anomalous X-ray diffraction, AXS, see sections I.W.2-I.W.4.

The total neutron structure factor can be given as

$$S_T^{RMC}(Q) = \sum_{i=1}^{Ntypes} \sum_{j=i}^{Ntypes} c_i c_j \bar{b}_i \bar{b}_j S_{ij}^{RMC}(Q) = \sum_{i=1}^{Ntypes} \sum_{j=i}^{Ntypes} coeff_{ij} S_{ij}^{RMC}(Q) \qquad \text{E 3}$$

where

$$coeff_{ii} = \frac{c_i^2 \bar{b}_i^2}{\sum_{i=1}^{Ntypes} \sum_{j=i}^{Ntypes} c_i c_j \overline{b_i b_j}}$$

$$coeff_{ij} = \frac{2 c_i c_j \overline{b_i b_j}}{\sum_{i=1}^{Ntypes} \sum_{j=i}^{Ntypes} c_i c_j \overline{b_i b_j}}$$

where *Q* is the scattering vector, c_i and c_j are the molar fractions, b_i and b_j s are the (neutron) scattering length of the components, and *coeff*-s are the coefficients. It has to be noted that in RMC the partial and total structure factors are defined the way that they oscillate around zero at large *Q* value! For the remaining of this document the RMC superscript is dropped, but it is implied. For X-ray and electron diffraction the formula is very similar, only instead of *b* the *Q or g*-dependent atomic scattering factors *f(Q)*.or *f(g)* should be used, therefor the coefficients will be *Q* or *g*-dependent as well!.

The calculated and the measured data are compared by calculating the χ^2 of the data set *i*:

$$\chi_i^2 = \frac{\sum_k \left(S_i^C(Q_k) - S_i^E(Q_k) \right)^2}{\sigma_i^2} \qquad \text{E 4}$$

where *k* is going through the points of the data set and σ_i is the weight factor of the data set. This formula is only valid, if no renormalization is used, see section 0 for details.

Then a chosen number of atoms are moved in the configuration. It has to be noted, then the more atoms are moved the less likely that the move will be accepted due to the possible violation of the hard sphere cut-off and the largest change in the calculated data, but if it is accepted then the configuration changes more rapidly mapping quicker the available configuration space. The number of moved atom could be arbitrary using the old format **.dat* file, but only 1 in case of the free format **.dat* file when running normal, atomic RMC (either for atomic systems or for molecules kept together with FNC), in case of molecular RMC it has to match the number of atoms in the molecule, see section I.D. The new χ^2 is calculated, and if it is lower, than the old one, then the move is accepted. If not, then the move is only accepted with $\exp[-(\chi_{new}^2 - \chi_{\mathrm{old}}^2)]$ probability. If there are too-close atoms in the configuration (atoms closer, than their hard sphere cut-off values), and the move is moved them above the hard sphere cut-off, or at least their distance increased, then the move accepted regardless the change in the χ^2.

Repeating this procedure, we arrive at a configuration hopefully producing a calculated data very close the experimental one, and therefore we have a 3-dimensional configuration, which is the good representation of the real system. The simulation length can be controlled by a time limit, or the total number of generated or accepted moves.

Neutron, X-ray, electron diffraction and EXAFS experimental data can be used for the fitting, and total or partial *g(r)* functions as well, as many as we like. The EXAFS data calculation is a bit different from the others, will be discussed below.

The histogram calculation uses reduced units (meaning that the distance is divided by the half box length), so the maximum value of the histogram calculation can go up to √3. The maximum reduced distance, x_{max} should/can be given (depending on whether the old or new file format is used) in the parameter file **.dat*. It has to be mentioned that if x_{max} >1, then the statistics is slowly deteriorating, as the sphere centred on the central particle starts protruding from the simulation box. This effect becomes worse after x_{max} >√2, so it is better to use larger simulation box with more atoms to increase the largest distance than use x_{max} >√2.

B. The units used

The RMC program is traditionally using Angstrom to measure distance and Å^{-1} for the inverse space. Angstrom is displayed in file headers and printouts. Regardless of this, data can be given in other units, but the direct and inverse space data has to be consistent. If potential is used, then the non-bonded LJ sigma parameter has to be given in Angstrom in the **.dat* file, but due to the fact that the GROMACS topology file format is adopted, in the **.top* and **.itp* files the GROMACS units, namely nm for distance has to be used in the topology and include topology files.

Energy related parameters are given in kJ/mol. For the tabulated potential the distance is given in Angstrom and the potential in kgÅ^2/s^2 .

Parameters, which should be given for all type should be given in the order the atoms are represented in the **.cfg* file.

Parameters, which should be given for all partials should be given in RMC order: for a 3 component system 1-1 1-2 1-3 2-2 2-3 3-3, which is the row-continuous order in the upper triangular matrix form.

C. Cut-off distances, ‘tooclose’ atoms, moveout option

In RMC the atoms are represented as hard spheres having a hard sphere cut-off distance specified in the **.dat* file for each atom type pair (partials). From version 2.3 there is a possibility to prompt the program to set the cut-off automatically to the smallest occurring distance for each partial, if the CUT-OFF = MIN or the CUT-OFF = MIN-ALL is used in the free format **.dat* file. The difference between them is only in case of flexible molecules, see below. From this version the CUT-OFF key word is not mandatory in the free format **.dat* file, if not given automatic determination is assumed.

It can happen, if the cut-off distances manually given that the initial configuration does not satisfy the cut off, and some of the atoms are closer to each other, than their cut off distance. These are the “tooclose’ atoms. To facilitate the elimination of these atom pairs, the moveout option in the **.dat* file can be switched on. In this case the moved atoms are chosen in TOO_CLOSE_FRACTION*100 % of the moves from the ‘tooclose’ atoms, and not from the entire configuration. The TOO_CLOSE_FRACTION constant is located in the *units.h,* the default value is 0.5. If a ‘tooclose’ atoms is moved, then the move is accepted regardless the χ^2, if the distance of the ‘tooclose’ atom pair(s) moved above the cut-off, or at least their distance increased. If the pair is not a ‘tooclose’ pair, or the moveout option is not used, then change of the χ^2 decides as it described in I.A., whether the move is accepted or not. When all the ‘tooclose’ atoms have been eliminated the simulation continues normally.

It has to be noted that in case of FNC constraints only the ‘tooclose’ atom pairs, which are not FNC pairs will be counted as ‘tooclose’ atoms, as with do not want to move away the FNC pairs from each other. The same logic applies to the automatic cut-off determination in this case, the smallest distance not belonging to an FNC-pair will be used as cut-off.

In case of potential runs with bonded interactions similarly only those ‘tooclose’ atoms pairs are counted as ‘tooclose’ pairs, which are not involved in a bond.

In this case the automatic cut-off determination using MIN the minimum of only those distances determined, which are not bonds, and not involved in 1-3 interaction of an angle. (Dihedral interactions are usually much weaker, and the range of the 1-4 pair distance is wide due to configurational or conformational isomerism, so they are not considered as exceptions.) This means that the cutoff will be the minimum of the intramolecular distances starting with 1-4 interactions and further, and the intermolecular distances. In case of MIN-ALL the smallest distance of atom pairs occurring for a given partial will give the cutoff regardless whether the atom pair is involved in a bonded interaction, or not. When the cutoff is checked during the simulation regardless the cutoff determination method, then bonded pairs and atom involved in 1-3 interaction of an angle are accepted regardless their distance (using the analogy of the normal FNC constraint), relying on the associated bond and angle potential to keep these atom pairs in an acceptable range. Therefore the difference between MIN and MIN-ALL is in the acceptance of the intermolecular and 1-4 and above intramolecular interaction distances due to the fact that MIN-ALL will determine much lower cutoffs than MIN. It is up to the user to decide that for the given system with the applied other parameters (most importantly the scaling of the contributions of the different potential components to each other and to the regular data sets) which option would give the desired result. In case of molecules where H-bonding can occur the setting of a higher cutoff with MIN will definitely restrict the available configuration space to the one available in the initial configuration, which might not be desirable, as a starting RMC configuration is only one equilibrium MD configuration, and cannot represent the average!

D. The molecular move

In the normal RMC algorithm one or more atoms are moved during a RMC step. However for molecular system it can be more advantageous to move a whole molecule together. This can be done by choosing the molecular move option in the **.dat* file. But in this case, the program has to be compiled supplying an adequate molecular constructor and a makemove function located in the *makemovecus.cpp* file. Obviously the molecular move for different molecules can be very different, so no standard molecular move can be written.

As a default the molecular move for the CCl_4 molecule is present in the program, but two other *makemovecus.cpp* files for C_2Cl_4 and H_2O are supplied in the Source code directory. If one of it is should be used, then the existing *makemovecus.cpp* has to be substituted with the new one. Even if normal atomic move is used some *makemovecus.cpp* file has to be present at compilation time, as the program is looking for it!

If molecular move for other type of molecules should be used, then it has to be written by the user following the guidelines visible form the existing molecular move files. The basic concept is that at each move a molecule is randomly chosen for moving with a random distance to a random direction, and then depending on the molecular symmetry, some rotations and individual atomic moves are performed as well. The custom makemove function has to perform the molecular move, and some chosen parameters (like the maximum rotational angle, maximum atomic moves…) can be set in the **.cus* file, which parameters can be changed without recompiling the program, see the example in CCL4_mol in the validation suite.

It is advisable to use the molecular move in case of simple molecules, as it results in quicker change in the system during the simulation, although the acceptance ration will most probably be lower.

E. Gridding of the simulation box

Gridding means that the simulation box is divided into a given number of sub-cells in each direction. If we know about each particle, in which grid cell it is located, and we want to calculate certain properties for only those particles that are not farther from the chosen particle than a given distance, then it is enough to calculate only for those particles, which are located in the same, and a given number of neighbouring grid cells. As checking, whether the move can be acceptable based on the satisfaction of the cut off distances falls in this category, calculation can be quicker, if the grid-based cut off check is performed before entering the lengthy histogram change calculation. The number of particles in the grid cell is regulated by this parameter. Based on this, the average number density of the sample and an additional safety parameter, SAFE_ADD located in the *units.h*, the program calculates the number of grid cells in each direction t. As due to inhomogeneity in the sample these numbers can vary, the program always checks, whether the maximum is not exceeded, before attempting to write into the arrays. If the maximum was exceeded, the program gives a message, resizes the necessary arrays automatically, and continues execution.

Gridding can only increase speed, if the number of particles in a grid cell is chosen adequately! It is preferable to set the desired maximum number of particles in a grid cell in the **.dat* file to a relatively low value (5-10) depending of course on the system size to ensure at least 5 or preferably more grid cell in each direction. The actual number of grid cells is calculated by the program and printed on screen during the initialisation period, or can be found in the **.grid* file. It has to be kept in mind that even if the largest cut off distance is smaller than the length of one grid cell, not just the cell containing the central particle, but all its closest neighbour cells are checked, as the central particle can be close to the edge. This way normally at least 27 cells are checked. Speed increase due to gridding can only be expected, if the number of grid cells to be checked is smaller, than the total number of grid cells!

F. The EXAFS data fitting

EXAFS stands for the extended X-ray Absorption Fine Structure spectrum.

XAS (X-ray absorption spectroscopy) is an element specific method to investigate the bond angles, bond lengths and coordination numbers. During the experiment the material under investigation is targeted with monochromatic X-ray beam (produced by synchrotron radiation). Some of the X-ray photons are absorbed by the material, and the rate of the absorption is measured versus the X-ray photon energy.

The X-ray absorption coefficient of a homogenous material can be given by

$$\mu(E) = \frac{1}{d}\ln\left(\frac{I_0(E)}{I(E)}\right) \qquad \text{E 5}$$

where E is the photon energy and d is the thickness of the sample. Generally, the absorption of X-ray is decreasing with the increasing energy of the X-ray photons, but distinct spikes corresponding to a drastic increase of the absorption can be detected at some energy. These are the absorption edges, and they correspond to the binding energies of the inner-shell electrons (K, L, M). As each chemical element has specific, well-defined binding energies, it is possible to select an energy range for the X-ray beam sweeping specifically only an absorption edge (and the following) region of a selected element. This way information of the neighbourhood of the atoms of this chosen chemical element can be obtained.

XAS spectrum can be divided into different parts based on the energy range of the X-ray beam compared to the absorption edge, but there is no consensus in the literature neither in the number nor in the limiting energy values of the ranges. A possible division is given here.

1.) Directly before the absorption edge can be found the pre-edge region, where no ionisation occurs, only transition to higher, non-completely filled or empty orbits.

2.) In the edge region, where photon energy, $E \leq E^0$+10 eV (E^0 is the ionisation energy) the XANES (X-ray Near Edge Structure) is observable.

3.) NEXAFS (Near-Edge X-ray Absorption Fine Structure) region is between $E^0+10<\mathrm{E} \leq E^0$+50 eV.

4.) EXAFS region is where $E > E^0$+50 eV.

It has to be noted that sometimes there is no division between the XANES and NEXAFS region, and the two acronyms are used as synonyms, although NEXAFS is usually used in connection with organic molecules and surfaces. The absorption edge itself sometimes is not considered to be in the XANES region, and the beginning of this region is set at $E > E^0$+5 eV. Sometimes the division between the XANES-NEXAFS and EXAFS region is set around 150 eV. From now on the acronym XANES will be used for the description of the XANES-NEXAFS range.

In the EXAFS region the kinetic energy of the photoelectron is higher and consequently the wave length and the scattering amplitude is smaller, so mainly single scattering of the photoelectron by the neighbouring atoms takes place.

The scattering amplitude and phase shift caused by the backscatterer depends on of the neighbour’s type, and the phase and the amplitude of the backscattered wave depends on the inter-atomic distance between the absorber and the backscatterer as well.

After the ejection of the photoelectron the absorber atom will be in an exited state due to the core hole. Relaxation can occur when an electron occupying a higher energy level jump down into the core hole. The energy difference between the levels is either emitted as a fluorescence photon (only rarely occur), or absorbed by another higher-level electron, which is emitted from the atom (Auger-electron).

The wave vector of the photoelectron (k) can be calculated from the X-ray photon energy, E and the ionization energy, E^0 by:

$$k = \sqrt{\frac{2m}{\hbar^2}(E - E^0)}$$ E 6

The oscillating absorption coefficient is normalized by the smooth atomic absorption background, μ^0 defining the EXAFS signal, $\chi(k)$:

$$\chi = \frac{\mu(k) - \mu^0(k)}{\mu^0(k)}$$

E 7

F.1. RMC specific remarks

It has to be noted that in RMC χ^2 is applied for denoting the difference between the experimental and calculated data, so it should not be confused with $\chi(k)$. For fitting EXAFS data in RMC, the values of k and $\chi(k)$ has to be given in the EXAFS data file as the input experimental EXAFS data (see II.E.5).

The EXAFS experimental data used in the fitting is denoted by $E(k)$ and it is calculated by the formula

$$E^E(k) = \chi^E(k)\frac{k^n}{k_{max}^n}$$ E 8

after reading the k and $\chi(k)$ data from the file, where k_{max} is the largest used k value of the data set. The power factor n has to be given in the **.dat* file, and it is a small integer usually between 1 and 3. The reason for using this weighing is to compensate for the amplitude decay, fitting $E(k)=(k/k_{max})^n\chi(k)$ with n=1, 2, 3 results in a physically more realistic configuration than in case of n=0.

The coefficients, $c(r,k)$ for the Fourier-transformation of the r-space information to k-space are r,k dependent, and has to be given in the coefficient file. The coefficients can be calculated for example by program FEFF, see some information how to do it in the document exafs_feff_rmc.*pdf* written by Pál Jóvári available on from the EXAFS page of the RMC_POT web site. Care has to be taken that the same r-points has to be used during the coefficient calculation as used in the RMC simulation. Keep in mind that in RMC always the middle of the bin is used for the given bin! It is possible that only a range of the r-dependent coefficients given in the coefficient file should be used during the fitting, this can be specified in the **.dat* file.

Different ranges can be specified for the different partials (all the possible atom type pairs of the edge particle type) for an absorption edge. The minimum and maximum indices of the r points (columns) for the partials are given in the **.dat* file, in the order $rmin_1$, $rmax_1$, $rmin_2$ $rmax_2$, ...$rmin_N$, $rmax_N$, where the indexing refers to the partials contributing to the given edge. For example for a 3-component system of As, Se, I for the As edge there are three partials (As-As, As-Se, As-I), for which the r index data has to follow in RMC order for the partials.

The calculation of $E^C(k)$ data is performed according to the following formula:

$$E^C(k) = \sum_{ip}^{np}\sum_{ir}^{nr} f * H(ip,ir)\frac{c(ir,k)}{N_e}\frac{k^n}{k_{max}^n}$$ E 9

where f is 2 for the pure partials and 1 for the mixed partials, as the RMC histograms contain counts only coming from unique pairs, $H(ip,ir)$ is the value of the ip-th partial histogram ir-th histogram bin, N_e is the number of the particles with the absorption edge. The first sum is going through all the partials containing the particle type producing the absorption edge.

The squared difference for the ith, EXAFS data set χ_i^2 is calculated as

$$\chi_i^2 = \frac{\sum_{j=1}^{N}\left(a_i E_i^E(k_j) + b_i - E_i^C(k_j)\right)^2}{\sigma_i^2}$$

E 10

only using constant and multiplication factor for renormalizations, σ_i is the weight factor for the data set.

F.2. Coefficient correction with E^0 shift

As there can be some error regarding the coefficient calculation, and the measurement, there is a possibility to try to calculate the *E(k)* with a somewhat shifted coefficient matrix from version 1.8. A maximum ΔE_0 shift can be chosen, which would mean

$$\Delta \boldsymbol{k}_{lmax} = \boldsymbol{k} - \sqrt{\boldsymbol{k}^2 - \frac{2\boldsymbol{m}_e \Delta \boldsymbol{E}^0}{\hbar^2}}, \; \Delta \boldsymbol{k}_{rmax} = -\boldsymbol{k} + \sqrt{\boldsymbol{k}^2 + \frac{2\boldsymbol{m}_e \Delta \boldsymbol{E}^0}{\hbar^2}} \qquad \text{E 11}$$

maximum left and right *k* values, as the relationship between ΔE^0 and *k* is not linear, see E 6. To make the calculation quicker, the shift cannot be any value between $-\Delta \boldsymbol{k}_{lmax}$ and $\Delta \boldsymbol{k}_{rmax}$. A symmetric, equidistant 2**N*+1 point grid in the energy space will be created between $-\Delta E^0 \rightarrow +\Delta E^0$, and the k values for each *E*-grid points will be calculated according to **E 12**.

$$\boldsymbol{k'}_{i,n} = \sqrt{\boldsymbol{k}_i^2 - \frac{2\boldsymbol{m}_e \Delta \boldsymbol{E}_{0,n}}{\hbar^2}} \qquad \text{E 12}$$

where the positive $\Delta \boldsymbol{E}_{0,n}$ values correspond to the left (smaller *k*) part of the *k* interval and the negative $\Delta \boldsymbol{E}_{0,n}$ to the right part.

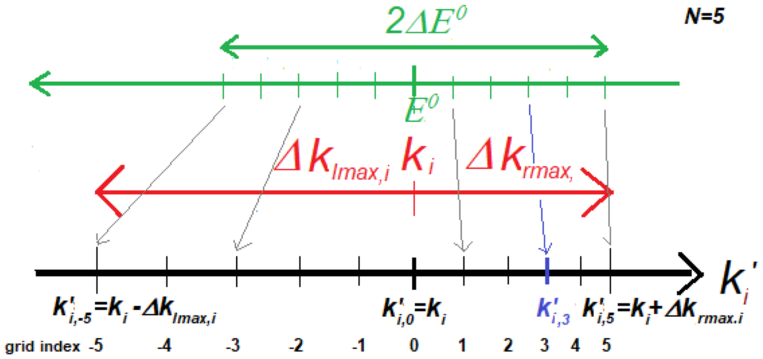


Figure 1. The relationship between ΔE^0 and the equidistant E grid, $\Delta \boldsymbol{k}_{lmax}$ *and* $\Delta \boldsymbol{k}_{rmax}$, and the non-equdistant *k*-grid values in case of N_{grid}=5, and chosen grid index 3.

The distance of the grid points are different in the left and right direction, and it is obviously *k* dependent. The values of the coefficient matrices for each partial is calculated with linear interpolation for each grid point, $k'_{i,n}$, and this coeff($k'_{i,n}$,r_j) will be used instead of the original coeff(k_{i},r_j) for the calculation of *E*(k_i). It has to be mentioned that due to the linear interpolation there will be some data point loss at both end of the *k*-range!

At every EXAFS-SHIFTSTEPth generated move, no atomic move is performed (similarly to the scalable background correction or AXS in case of *I(Q)* data sets) , but a grid index between $-N_{grid} \rightarrow +N_{grid}$ is randomly chosen in case of each EXAFS data set, where E^0 shift is performed. The grid index is the same for each partial of a data set. The EXAFS data E(k_i) will be calculated using the coeff($k'_{i,n}$,r_j) values. The χ^2 is calculated, and the move accepted or rejected the usual way. If the move is rejected, then the old grid index is used for the subsequent steps, if accepted, then the new one till the following EXAFS-SHIFTSTEPth generated move. The starting grid index is zero meaning no shift by default.

This feature is only accessible using the free format input file. In the [GENERAL] section optionally the default value of the EXAFS-SHIFTSTEP can be modeified. For each EXAFS data set in the [EXP] section optionally the DELTAE0_NGRID key word can be given, expecting the maximum E^0 shift in eV and the number of grid points in one direction.

Using this option might indicate that only one segment of the grid points is used. In a following simulation therefore we might one to constrain the number of used grid points to this segment, and even like to continue the simulation with a grid point with non-zero grid index, which means using an asymmetrical grid. To make this possible, an other keyword, E0GRID-FROM_TO_START was introduced expecting 2 parameters for the first and last grid point to use and a third, optional value for the grid index to start the simulation with. If this keyword is given for a data set, then DELTAE0_NGRID should be also given for it, as this latter specifies the grid.

The shifted coefficient matrices are only calculated for the used grid points. The data points loss is determined based on the whole symmetrical grid given by DELTAE0_NGRID.

For example if the following lines are present for a EXAFS set:

DELTAE0_NGRID = 2 5

E0GRID-FROM_TO_START = -3 1 -2

then only the blue range given in Figure 2 will be used representing E0 shift values 1.2 →-0.4 eV, starting with 0.8 eV shift.

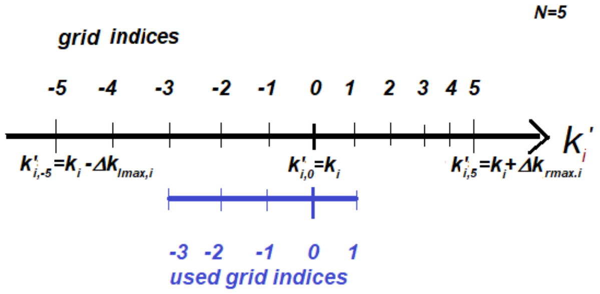


Figure 2: The usage of the asymmetrical E0 shift grid, where the original grid has 5 shifted grid points in one direction, and only the -3 → 1 range is used.

G. The renormalization of the data

The *g(r)*, *S(Q)*, *F(Q*, *F(g)* and *E(k)* data sets can be renormalized to correct some errors due to the measurement. If renormalization is used, then not simply the squared difference between the calculated and experimental data set divided by sigma square of the data set is calculated during the χ^2 calculation, but the experimental data is renormalized the following way:

$$\chi_i^2 = \frac{\sum_{j=1}^{Np_i}\left(a_i A^E_{i,j}(x_{i,j}) + b_i + c_i x_{i,j} + d_i x^2_{i,j} + e_i x^3_{i,j} - A^C_{i,j}(x_{i,j})\right)^2}{\sigma_i^2} \qquad \text{E 13}$$

where N_{pi} is the number of data points for data set i, $A^E_{i,j}(x_{i,j})$ is the experimental, $A^C_{i,j}(x_{i,j})$ is the calculated data, for neutron and X-ray fitting $x_{ij}=Q_{i,j}$ for electron diffraction $x_{ij}=g_{i,j}$ is the *j*th data point of the *i*th set, a_i , b_i , c_i d_i and e_i are the renormalization coefficients. The coefficients can be calculated by linear regression explicitly. Up to version 1.2 only up to quadratic renormalization (coefficients a_i , b_i , c_i d_i) could be used. From version 1.2 cubic renormalization was added to the formula.

Whether to use renormalization is controlled by the parameters given in the **.dat* file. In case of old, fixed format the vary amplitude, constant, linear, quadratic and cubic switch in the **.dat* file is used, and displayed during the simulation and written to the **.fit* file. To maintain downward compatibility for the **.dat* files, cubic renormalization can only be featured in the **.dat* file, if 1 is given for use_cubic switch after the renormalization switch for the given data series. If it is not given then old format is assumed, and the quadratic parameter is read as the last for the data set from the **.dat* file. In case of free format data input the RENORM key word switches the *a* parameter, the others are represented by the POLY-BACK-FLAGS key word for each data set.

For *g(r)* data sets $x_{ij}=r_{ij}$ and only a_i exists in the old format, but from version 1.2 constant, linear, quadratic and cubic renormalization can be used as well, if 1 is given for use_cubic as described above. It has to be noted that this new renormalization is only recommended, if the *g(r)* is somehow defective (not tending to one). Using constant is especially not recommended, as it can make *g(r)* smaller than zero, which is in contradiction with its definition. (See the structure of the **.dat* file in II.C).

The same formula is used for *E(k)* fitting, except that $x_{ij}=k_{ij}$ and only the a_i and b_i coefficients used and featured in the **.dat* file.

The program determines the coefficients, where the difference between the calculated and the experimental sets is the minimum. (for the sake of clarity see the Quadratic background correction within RMC by László Temleitner and the addition of the Cubic correction on the RMC web site).

In case of X-ray data sets, optionally not only the structure factor, but instead the intensity, *I(Q)* can be fitted as well. As the intensity can be expressed for the *j*th data point as

$$I(Q)_j=\langle f(Q_j)\rangle^2 F(Q_j)^T+\langle f^2(Q_j)\rangle+B(Q_j)\,\exp(-\alpha_i\cdot Q_j^2)=\langle f(Q_j)\rangle^2 F(Q_j)^T+A(Q_j)+B(Q_j)\,\exp(-\alpha_i\cdot Q_j^2), \qquad \text{E 14}$$

where is for $f(Q_j)$ is the atomic scattering factor (see I.I), $B(Q_j)$ is the Compton scattering term. $A(Q_j)$ and $B(Q_j)$ is calculated by the program (see I.I). The exponential term with α_I is responsible for damping the Compton term.

For *I(Q)* fitting χ^2 is defined as

$$\chi_i^2=\frac{\sum_{j=0}^{Np_i}\left(a_i I_{i,j}^E+b_i-I_{i,j}^C\right)^2}{\sigma_i^2} \qquad \text{E 15}$$

no linear, quadratic and cubic term is available here. Unfortunately the value of α_i is somewhat uncertain, and can be fitted as well (so the otherwise certain simulated data $I^C_{i,j}$ can contain a renormalizable term too), and the renormalization coefficients a_i, b_i and α_i in this case cannot be calculated by linear regression, it has to be determined by non-linear regression.

For *I(Q)* fitting only the following cases are implemented:

- a_i is fitted, b_i and α_i custom (linear regression)
- a_i and b_i is fitted, α_i =0 (linear regression)
- α_i is fitted, a_i and b_i custom (non-linear regression)
- a_i and α_i is fitted, b_i =0 (non-linear regression)
- a_i, b_i and α_i is fitted (non-linear regression)

Unfortunately non-linear regression can only be done by iterative methods, RMC uses the Levenberg-Marquardt method (implemented according to the Numerical Recipes, The art of scientific computing, 3rd edition https://e-maxx.ru/bookz/files/numerical_recipes.pdf) for optimizing the required parameters. The iterations can be time consuming, and might not converge, so be careful using these options. The parameters (fudge lambda, factor to change lambda, maximum number of iteration, limit for convergence) governing the iterative process have the default values suggested by the manual, but all of them can be adjusted by the user as well, as described in section II.C.2, as the *I(Q)* fitting is only available if the free format parameter file is used.

Another unusual feature of the *I(Q)* fitting is that at certain cases non-trivial custom parameters can be given in the **.dat* file, these are described in section II.C.2 as well. (The trivial values are a_i=1.0, b_i=0.0 and α_i=0.0.

H. Neutron scattering coefficient calculation

From version 2.3 the neutron scattering coefficients no longer have to be given by the PARTIAL-COEFFS key word in the [EXP] section of the free format **.dat* file. If it is not present, and the CHEMICAL-SYMBOLS key word is given in the [GENERAL] section, then it can be calculated assuming natural abundance for each component. It has to be noted, that the CHEMICAL-SYMBOLS keyword can contain non-standard element symbols and ions used by X-ray coefficient calculation, (see I.I for details), but the program determines the corresponding standard element symbol, and this is used in connection with the neutron coefficient calculation.

If different isotope composition is required for an element, then in the [EXP] section of the given data set the ISOTOPE-COUNT_SYMBOLS_RATIOS key word should be given. First the number of isotopes for this component should be specified followed by the isotope symbols in the format Cl35 (chemical symbol followed by the mass number) and then the ratios. The sum of the ratios should be 1!. ISOTOPE-COUNT_SYMBOLS_RATIOS key word can occur as many times in the [EXP] section of a data set as the number of component types.

If the PARTIAL-COEFFS key word is (also) present for a neutron data set, then the given coefficients will be used regardless.

The bound coherent scattering lengths are from V. F. Sears, Neutron News (1992) 3, Nr. 3, 26-37.

I. X-ray atomic scattering factor and Compton scattering calculation

From the 2.0 version the X-ray atomic scattering factors (form factors) are calculated by default by the program. For this the CHEMICAL-SYMBOLS key word with the symbols for the constituents in the same order as in the **.cfg* file should be included in the [GENERAL] section. The extended Waasmaier-Kirfel[6] table contains the parameters for the calculation of *f(Q)* for the atoms up to atomic number 103. The H atom parameters calculated by L. Temleitner were added to the table. The parameters for ions of most of the atoms came from the same source[6]. Help is given by the program about the implemented symbols (see III.D).

The program expects the chemical symbols in standard format (like As). There are a few non-standard symbols: for hydrogen beside the normal H (which refers to the atomic Schröedinger-equation solution) there is a HS as well, which refers to the hydrogen in a hydrogen molecule. There is CV and SiV which are the valence states (sp^3) for carbon and silicon.

If custom values should be used for the atomic scattering factors then the READ-COEFFS=1 key word should be given in the [EXP] section of the given data set. As the file format can be different in case of *I(Q)*-fitting depending on whether the *f(Q)*-s are read or not and whether scalable background correction is used or not, see the file format in II.E.3.

Also from version 2.0 the Compton term used in case of *I(Q)* fitting is calculated by the program using the Balyuzi[7] method and tables extended with parameters for hydrogen as well. Here only atomic parameters are supplied for atoms with atomic number 1-95,102. If ions are given after the CHEMICAL-SYMBOLS key word, then the parameters for their neutral atom will be used. The program gives help about the implemented elements and their parameters (see III.D). Parameters for HS and CV implemented here as well.

If Compton scattering with different parameters should be used, then it is possible giving $Compton(Q)*\exp(-\alpha Q^2)$ as the background correction using the I(Q)BACKG_MU_DMU-MAX = 1 1.0 0 line for the given data set and giving 0 for the Nstep_μ. In this case no changing of the background will be performed, and technically $a(I^E(Q)- Compton(Q)*\exp(-\alpha Q^2))+b+I^C(Q)$ will be fitted. No scalable background correction can be used for this data set in this case.

J. R_w calculation

The R_w .value used usually in crystallography to characterize the goodness of the fit is also calculated for the *g(r), S(Q)* , *F(Q)* and *E(k)* data series, according to the formula:

$$R_{w,i} = \sqrt{\frac{\sigma_i^2 \chi_i^2}{\sum_{j=1}^{Np_i} (a_i A_{i,j}^E + b_i + c_i x_{i,j} + d_i x_{i,j}^2 + e_i x_{i,j}^3)^2}} \tag{E 16}$$

where i means the ith data series, $x=r$, (for old format **.dat* files $b=c=d=e=0$, see section 0 for details) for the *g(r)* data, $x=Q$ for the X-ray and neutron data ($e=0$ for old format), $x=g$ for electron diffraction data ($e=0$ for old format), $x=k$, $c=d=0$ for EXAFS data, A^E_{ij} is the experimental. R_w is printed on screen together χ^2 and given in the **.hst* file for the end of the run.

K. The cosine distribution of bond angles constraint

The angles between the bonds of three atom types can be given as a constraint in the **.dat* file (see II.C. and II.F for the syntactic). Any number of constraints can be given, and RMC_POT makes a distinction between the types of the neighbours, so for a two component system, if the middle atom denotes the central type, then for a type A central A-A-A, B-A-A and B-A-B type different cosine distributions can be calculated (the A-A-B would be the same as B-A-A.) Take care that in the **.dat* file after the mode indicator first the central then the two neighbour types has to be given!

The constraint can work in 4 different modes depending on the method given:
0: The theoretical distribution is calculated as a step function;
1: The theoretical distribution is calculated as a Gaussian distribution.
2: No angles are required in a certain range
3: Experimental distribution should be read from a file.

In case of method 0-2 the spacing of the cosine distribution of bond angles histogram is determined by the *dcos(θ)* parameter given in the **.dat* file.

A constraint can be "positive", which means that bond angles are needed in the given region, and "negative" meaning that angles are not wanted in the given region. Positive constraint can be set up by specifying either 0 for the calculation method indicating a step function, or 1 meaning Gaussian for the shape of the constraint. Step function means having uniform distribution with integral 1 between angle - *wcontrol ->angle* + *wcontrol*, (both given in degrees), 0 otherwise. Gaussian indicates a normal distribution with integral 1 having the maximum at angle (given in degree), which is converted to radians and the cosine of it is calculated. This cos(θ) value will give the peak position, and the width is controlled by *σ=wcontrol*, where *wcontrol* is used as it is given in the **.dat* file, so it is a *dcos(θ)* value. So it is a good approximation that the theoretical distribution will span the *cos(θ)-3*wcontrol* → *cos(θ)+3*wcontrol* range in *cos(θ)* units, which corresponds to the confidence interval *σ*=3 (99.7 % of the data is inside the range). The size of the confidence interval is regulated by the CONF_INT constant in *units.h*, the default is 3.

In these cases the theoretical distribution is calculated for all the bins spanning the range -1=<*cos(θ)*=<1, so only one positive constraint (desired angle) can be meaningfully given for a neighbour1-central-neighbour2 triplet, as more than one constraint would ruin each other's effect.

Negative constraints that no angles desired in a given region can be set up using calculation method=2. In this case the constraint is for the given region and not for all the bins, as it could interfere otherwise with a "positive" constraint for the same particle types. A normal distribution curve similarly to method=1 is calculated from that part of *-CONF_INT*wcontrol* → + *CONF_INT*wcontrol*, which is in the -1 → +1 range, centred on the not desired angle. The calculated curve is used during the χ^2 calculation to provide an additional, *cos(θ)*-dependent weight by multiplying the calculated distribution with it, and calculating the product's squared difference from 0. Make sure that the interval of a negative constraint does not overlap with a positive constraint for the same particle triplet!

In case of method 3 the experimental distribution is read from a file. The *cos(θ)* values have to span the whole –1 → 1 interval, has to denote the middle of the bins, and have to be equidistant! See the file structure in chapter II.F. This constraint can be used together with other type of cosine distribution of bond angles constraints, and in this case, the spacing for the experimental data does not have to be the same as for the other constraints!

L. Coordination number constraint

It was discussed in the literature that RMC tends to produce the most disordered structure consistent with experimental data. So it can be a good idea to use some extra knowledge about the structure, as for example the preferred coordination number. This means that we can specify that around an atom of the central type between r_{min} and r_{max} what should be the preferred coordination number of the atoms of the neighbour type for the desired fraction of the central atoms. It is possible to have more, than one neighbour type for the same constraint. For example in case of 3 atom types it is possible to define a constraint that the central atom of type1 should have 3 atoms belonging to type2 and type3 between r_{min}[type2] → r_{max} [type2] and r_{min}[type3] → r_{max}[type3] respectively, which means that there are several possibilities satisfying this constraint (type2-type2-type2, type2-type2-type3, type2-type3-type3 and type3-type3-type3). Coordination number constraint contributes to the χ^2 according to the formula below:

$$\chi^2 = \sum_{m}^{n_{cc}} \frac{\left(\frac{N_m^S}{N_m^C} - N_m^f \right)^2}{\sigma_m^2} \qquad \text{E 17}$$

where σ_m, is the respective standard deviations, n_{cc} is the number of coordination number constraint, N_m^C is the number of central atoms and N_m^S is the number of atoms satisfying the *m*th coordination constraint, N_m^f is the desired fraction of the *m*th coordination constraint.

There can be more, than one constraints specified especially in a multi-components system. If multiple constraints are necessary between the same type of atoms, which only differ in the desired coordination number and fraction, then only one constraint has to be specified but with more than one sub-constraint. Each sub-constraint will have their own sigma (see the II.C how to specify the constraint). Using sub-constraints instead of using separate coordination constraints save memory and time, so it is highly advisable to use them.

The central atoms with the number of neighbours are saved into the **.cnc* file. From version 2.4.1 detailed information with the indices, types and distances of the neighbours can be saved to the **.cncd* file, if WRITE-CNC-DETAIL = 1 is given in the [COORD] section of the constraint in the free format **.dat* file.

M. Average coordination constraint

Sometimes it is better to constrain not the coordination number of each individual central atom, but the average coordination number of the neighbour type atoms around the central type atoms between r_{min} and r_{max}. The χ^2 contribution is calculated as

$$\chi^2 = \sum_{n}^{n_{ac}} \frac{\left(\frac{N_n^{sn}}{N_n^C} - N_n^d \right)^2}{\sigma_n^2} \qquad \text{E 18}$$

where σ_n, is the respective standard deviations, n_{ac} is the number of average coordination number constraint, N_n^C is the number of central atoms and N_n^{sn} is the total coordination number for the *n*th average coordination constraint, N_n^d is the desired average coordination number. See the II.C how to specify the constraint.

N. The Fixed Neighbour Constraint (FNC)

The FNC is invented to keep molecules together in a relatively rigid way. For this a **.fnc* file (see the structure later in II.G) has to be supplied, where for each atom of the configuration the number of neighbours, their indices, and their FNC constraint type has to be given. For each FNC constraint type the minimum and maximum distances between which the distance of the atom pairs belonging to this constraint should stay is specified in the header of the file. In the original implementation the simulation can only be started, if the constrained distances are already in the range, which is not entirely practical. This corresponds to the FNC switch fnc=1.

To remove the necessity that the initial configuration should satisfy the FNC constraint, two other possibilities are included. To be able to start the simulation, even if not all the constrained pairs are in range by resetting the maximum and minimum constrained FNC distances can be achieved by using fnc=2. If we do not want to widen the FNC range, then fnc=3 could be used, in this case at the beginning warning is generated about the out-of-range pairs, and only those moves are accepted, where the new distance is closer to or in the desired range, than the old one, hopefully eliminating all the out-of-range pairs.

There is another possibility for keeping molecules together, which is using fnc=4, flexible, MD-like molecules, kept together by forces (see Q.1.1(2)).

O.Advanced geometric constraints

For the advanced geometric constraint the code has to be compiled with _ADVANCED_GEOM_CONST compiler option.

O.1. Common neighbour constraint

Let’s consider two atoms 1 and 2 from type A and B with distance between d_{min} and d_{max} from each other. The number of these pairs is N^P in the configuration, and they will be called $primary_1$ and $primary_2$. We would like to have N^T number common neighbours (secondary atoms) in a given distance range from the primaries from types B or C for a given fraction of the primary pairs.

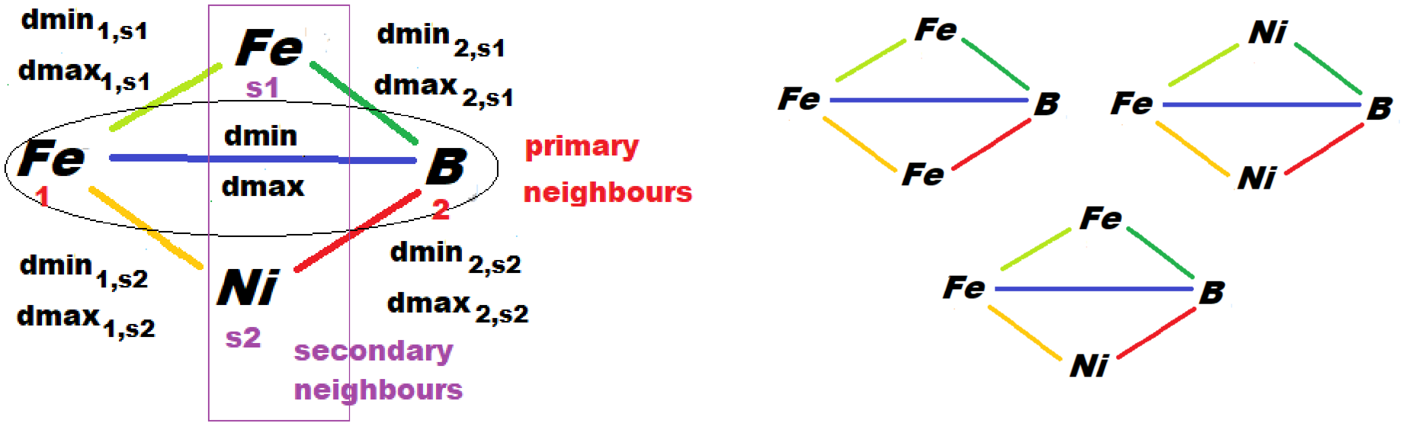


Figure 3. Demonstration of the common neighbour constraint.

In Figure 3 a common neighbour constraint can be seen with Fe and B primary atoms, which should have two common Fe or Ni neighbours in the given range for let’s say 80 % of the Fe-B pairs with distance between d_{min} and d_{max}. This means that this particular constraint can be satisfied by three different ways, as it is visible on the right part of the picture. The number of secondary types can be freely chosen, maximum is the number of atom types in the configuration. The syntax of the constraint can be found later (page 50). The χ^2 contribution for all the CONC constraints is calculated as

$$\chi^2 = \sum_n^{n_{com}} \frac{\left(\frac{N_n^S}{N_n^P} - f_n\right)^2}{\sigma_n^2}, \tag{E 19}$$

where N_n^S is the number of atom groups satisfying the n*th* constraint from the N_n^P primary pairs, f_n is the desired fraction and σ_n is the weight parameter for this constraint.

O.2. The second neighbour or Q^n constraint

In the case of this constraint we are looking central atoms of type *i* each having *n* first neighbour of type *j* between d_{min1} and d_{max1} Å and through this neighbour at least one second neighbour of types k_1-k_N between $d_{min2,1}$-$d_{min2,N}$ and $d_{max2,1}$-$d_{max2,N}$ Å for *f* fraction of all the central atoms. In other words the central atoms of type *i* should be in the Q^n coordination state regarding the *j* type first neighbours, which have another neighbour of types k_1-k_N (excluding the central atom in case of $i=k_m$). The constraint should be satisfied for *f* fraction of the central atoms. The *n* target coordination number, the fraction, the weight parameter, the atom types and the distance ranges are the required parameters.

The χ^2 contribution coming from the SNC constraints is calculated

$$\chi^2 = \sum_n^{n_{sec}} \frac{\left(\frac{N_n^S}{N_n^C} - f_n\right)^2}{\sigma_n^2},$$

where N_n^S is the number of central atoms satisfying the n*th* constraint from the N_n^C central atoms, f_n is the desired fraction and σ_n is the weight parameter for this constraint.

This constraint was introduced in version 1.7.1, and from this version the new features are ONLY available in the free format **.dat* file!

For example in SiO_2 100 % of Si should be in the Q^4 coordination state with O first neighbour and Si second neighbour, see Figure 4.

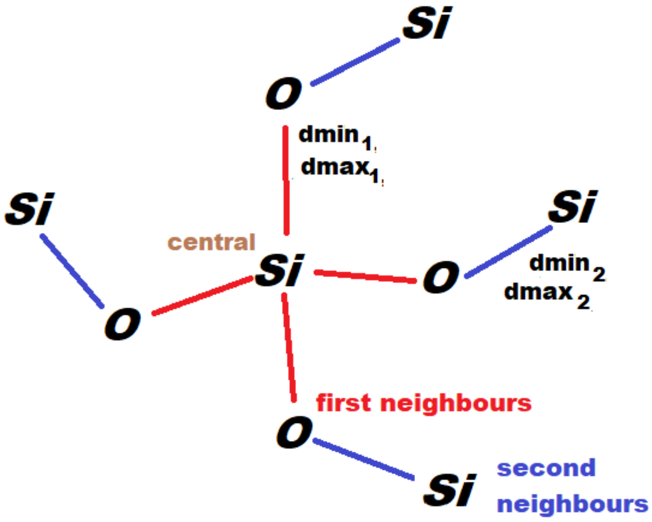


Figure 4. Demonstration of the second neighbour constraint with Q^4 configuration, where the central Si atom has 4 O-Si first and second neighbours.

Another example is the borosilicate glass, where the B central atom should be Q^3 coordinated with O first and Si or B second neighbours let's say for 80 % of the B atoms, see Figure 5. It has to be noted that in case of multiple second neighbour types there can be several ways, how the constraint can be satisfied, in this case four different configurations (BO_3Si_3, BO_3Si_2B, BO_3SiB_2 or BO_3B_3) can satisfy the constraint.

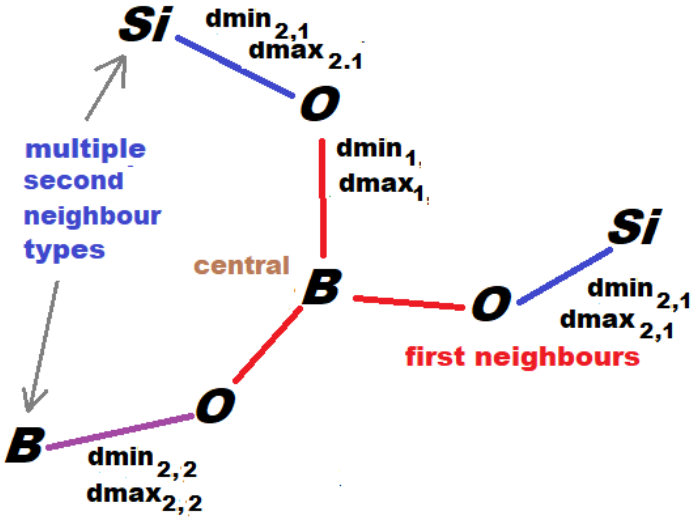


Figure 5. Demonstration of the second neighbour constraint with Q^3 configuration and multiple second neighbour types for borosilicate glass, where the central B should have 3 O first neighbour with either Si or B second neighbours.

O.3. Bond valence sum constraint

From version 2023.1 the possibility of the bond valence sum constraint (BVS) was included in the program, somewhat similarly to what is written in RMCProfile Manual v6.7.4. This feature is only available, if the program is complied with the _ADVANCED_GEOM_CONST compiler option and can be used only in case of the free format **.dat* file, giving the parameters after the [BVS] tag (see II.C.2 for the available key words). This constraint can only be used for atomic systems (no localized bonds), but non-bonding potential can be used together with it.

The bond valence sum of an atom can be calculated according to E 20:

$$V_i=\sum_j \ exp\left(\frac{R_0^{ij}-R_{ij}}{b_{ij}}\right)$$

E 20

where V_i is the bond valence sum of atom i, j is running through the neighbours with the desired types inside the cutoff, R_0^{ij} is the expected bond length of atom pairs of these types, and b_{ij} is a parameter controlling the shape of the curve. The R_0 and b parameters depend on the oxidation state of the central and neighbour atoms, default values based on the table compiled by I. David Brown (Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario Canada, idbrown@mcmaster.ca) are included in the program. If default value is needed for R_0, or R_0 and b, they will always be supplied from the table, if available, if only default value for the b parameter is needed, 0.37 Å (I. D. Brown and D. Altermatt, Acta Crystallogr., Sect. B, 1985, 41,244) is used.

The bond valence is calculated for every atom of the given central type, for every neighbour of desired types inside the specified cutoff.

The χ^2 contribution of the constraint is calculated according to E 21:

$$\chi^2_{k,BVS}=\sum_{i=1}^{N_i}\frac{\left(V_i-V_k^{ideal}\right)^2}{\sigma_k^2}$$

E 21

where i is running through the central atoms, the V_k^{ideal} is the desired bond valence for central atoms of the k-th constraint, V_i is the actual bond valence sum of atom i, and σ_k^2 is the weight parameter. There can be several neighbour types for a central type, with their appropriate R_0 and b parameters.

If an atom appears in a system in different oxidation states connected to BVS constraints (either as central or neighbour type), then each oxidation state has to have its own atom type in the configuration, and the appropriate BVS constraints accordingly.

The BVS for each central atom for each BVS constraint is written into the text type **.bvs* file.

P. The local invariance

The local invariance was introduced based on the work of M. J. Cliffe et al[8]. In their work an additional cost term increasing with the deviation of the local *g(r)* of each atom from the experimental *g(r)* was used. For practical reasons a coarse-grained approach was used, instead of storing all the pair distances, a local histogram is calculated and stored for each atom. Storing all the distances even for a system consisting of few thousand atoms would have too large memory requirement, and for a lager system sometimes used in RMC would have been completely impossible to do. There are two possibilities for the calculation of the local χ^2-contribution. The first one is the bin-based approach, where the deviation of the local histogram from the normal, average histogram is calculated according to E 22:

$$\chi^2_{loc}=\sum_{a=1}^{Nt}\sum_{i=1}^{N(a)}\sum_{b=1}^{Nt}\sum_{j\neq i}^{N(b)}\sum_{k=0}^{N_{bin}}\frac{\left(H^i_{ab}(k)-\frac{\overline{H_{ab}(k)}}{N(a)}\right)^2}{\sigma^2 dV(k)}$$

E 22

where N_t is the number of atom types, N(a) and N(b) are the number of atoms of type a and b, N_{bin}=int((*loc_ratio*$_2$-*loc_ratio*$_1$)**d*/2/*dr*$_{loc}$)+1 is the number of local histogram bins involved in the calculation where d is the size of the simulation box in Å, dr_{loc} is the width of the local histogram bin in Å, *loc_ratio*$_1$ and *loc_ratio*$_2$ are the minimum and maximum distances for the local histogram calculation in reduced units. $H^i_{ab}(k)$ is the kth bin of the ith atom's local histogram for the ab type partial, $\overline{H_{ab}(k)}$ is the average histogram for the ab partial's kth bin, $dV(k)$ is the volume of the kth histogram bin and σ^2 is the weight parameter.

The other approach is distance based. This is closer to the original implementation of Cliffe. The χ^2 is calculated as the squared difference of the distance (which in this coarse grained approach is the middle value of the histogram bin which it can be found) of the jth neighbour atom of type b of a central atom i from type a compared to the average distance of the jth neighbour of type b for all the central atoms of type a. The jth neighbour of a central atom means the one with the jth largest distance among all the b-type neighbours. In case of the distance-based approach it is possible to divide the interval to several consecutive parts with their own weight parameters to be able to make the contributions of the different intervals separate. The χ^2 contribution of the *n*th interval is calculated as

$$\chi^2_{loc_n} = \sum_{a=1}^{Nt} \sum_{i=1}^{N(a)} \sum_{b=1}^{Nt} \sum_{j=J_n(b)}^{J_{n+1}(b)} \frac{\left(I^i_{ab}(j) - \overline{I_{ab}}(j)\right)^2 dr^2_{loc}}{\sigma^2_n dV\left(\overline{I_{ab}}(j)\right)} \qquad \text{E 23}$$

where *j* is running through the neighbours of type *b* of atom *i* of type *a* arranged according to increasing distance from *i* starting with index $J_n(b)=int(loc\_ratio_n*N(b))$ and ending with $J_{n+1}(b)= int(loc\_ratio_{n+1}*N(b))-E$ where loc_ratio_n and loc_ratio_{n+1} are the ratio of neighbour atoms for the interval boundaries, and E is 0 for the last interval and 1 otherwise. $I^i_{ab}(j)$ is the index of the local histogram bin in which the distance of the *j*-th type *b* neighbour from central atom *i* of type *a* is found, and $\overline{I_{ab}}(j)$ is the average bin index for the *j*-th type *b* neighbours' distance for all the central atoms of type *a*, dr_{loc} is the width of the local histogram bin in Å, $dV(\overline{I}_{ab}(j))$ is the volume of the average bin and σ^2 is the weight parameter. In this case the local histogram is calculated and stored for the maximal range of 1.732 reduced distance, as it is not known, how much the local atomic environments differ, but only a portion of this regulated by loc_ratio_1 and loc_ratio_{Nloc+1} is used for the χ^2 calculation. Basically the different intervals correspond to different spherical shells centred on the central atom, whose boundaries are determined through the *loc_ratio*. The average distance of the first and last neighbours in Ångstrom for all the partials is given as part of the screen output during the run.

The bin size of the local histogram calculation can be different from the normal histogram used for the calculation of the data sets. As statistic is not an issue here, it can be chosen finer, than the normal histogram. The smaller the bin size is, the most this coarse grain approach will mimic the original idea. The limiting factor for decreasing the local histogram bin size is the larger memory requirement. The bin size of the local histogram can be given in the **.dat* file.

This necessitates the storage of the local histogram (*Nt·Nbin* elements for each atom), which can have large memory requirements for larger systems. The local histogram is saved to the **.lhgm* file, saving and loading can take noticeable time, sometimes more, than recalculation!

The local invariance calculation is only performed, if the code is compiled with the _LOCAL_INV compiler option. In this case from a new line after the number of threads to use first the number of local invariance intervals are read, then the weight parameters for the local invariance intervals. After that the calculation mode can be given, and then $N_{loc}+1$ real values for the atom ratios to use (where N_{loc} is the number of intervals), loc_ratio_n. In case of the bin-based calculation (where only one interval can exist) the two real values mean the starting and limiting reduced distances, for which the local invariance is calculated, the histograms are stored only between them. In case of the distance-based approach, the *n.* and *n+1.* ratio sets the boundaries of the *n*th interval they represent the fraction of neighbour atoms (the same fraction value resulting in different atom number for each type according to the different number of atoms/type) to begin and end the local invariance calculation with. Check the actual boundaries in the **.hst* file, and in the screen output. So in both case it is possible to calculate the local invariance only for a portion of the range the normal histogram is calculated!

In case of the free format **.dat* file there should be as many LOC-INT-FROM_SIGMA… key words as intervals setting the beginning and the sigma for the interval (for bin-based method only one!), and one LOC-INT-TO to set the end of the last interval.

Only compile the code with this option, if you really want to calculate the local invariance, as it is much slower, than the normal RMC run!

Q.The potential-related features

RMC can use different potential-handling methods, pair-potentials and many-body potentials as well. The pair potential-based approach, where the potential is divided into non-bonded and bonded potentials is implemented based on the GROMACS molecular dynamics software (see I.Q.1).

The many-body potential approach is based on AENET which is the *Atomic Energy Network* software to train and to use Artificial Neural Network potentials (ANN). RMC can only use the ANN potentials previously trained by AENET. Presently AENET uses three-body potentials, but if new versions of this software will be available and can use more than three-body interactions, as far as the potential file format does not change, RMC will be capable of handling it. In this case there is only one total potential component (no separation to bonded and non-bonded components), and optionally the separate atomic energies can be outputted as well. As the total energy is usually negative, algorithmically it is handled similarly to the non-bonded potential (see I.Q.2).

The potential energy of the system will make a contribution to the χ^2, so guiding the system to reach an energetically more favourable state. More precisely, if non-bonded interactions (or ANN) are present it contributes to a second, potential-related χ_P^2. The contribution is defined, as the V_{NB}/σ_{NB}^2 and therefore can be negative, this is the reason that it is not added to the normal χ^2, as it could wipe out the deviation of the data set.

Q.1. GROMACS-type non-bonded and bonded potential

In this section the molecular-dynamics simulation-related pair potential based interactions will be discussed. There is a much simpler way, if only non-bonding interaction is needed, by using tabulated potential, which is described in section I.Q.1.3.

Both non-bonded and bonded potential can be calculated. For the non-bonded potential calculation the potential switch in the **.dat* file has to be set to larger than one (see later) instead of 0. The non-bonded potential can be calculated for both atomic and molecular systems. In each case a GROMACS-type topology (**.top*) and if necessary, additional include topology (**.itp*) files have to be supplied, with some additional parameters, discussed later.

The potential energy of the system will make a contribution to the χ^2, so guiding the system to reach an energetically more favourable state. More precisely, if non-bonded interactions are present it contributes to a second, potential-related χ_P^2. The contribution is defined, as the V_{NB}/σ_{NB}^2 and therefore can be negative, this is the reason that it is not added to the normal χ^2, as it could wipe out the deviation of the data set. If only bonded potential is used, but there is Ryckaert-Bellemans dihedral potential among them, then the potential energy can also be negative, in this case a fake non-bonding potential (very large sigma to make the non-bonded contribution small) should be applied, as this way all the potential related terms contribute to χ_P^2.

It is optional, whether the whole systems' potential will make only one contribution or the potential of the RMC partials of a multi-component system can each make their own contribution with their individual sigma value to χ_P^2, so weighed separately from one another. This is decided by the weight mode parameter following the potential switch. In this case a separate sigma value for all the partials has to be given in the **.dat* file.

After it is checked that the hard sphere cut off distances are not violated, the histogram and together the non-bonded potential is updated.

The non-bonded interaction consists of a dispersion and repulsion term making up the van der Waals (vdW) term, and a Coulomb term.

Non-bonded interactions are calculated up to a certain cut off radius, which not necessarily coincide with the r_{max} value of the histogram calculation. The cut offs can be set in the **.dat* file separately for the vdW and the Coulomb interaction, the tabulated potential or the ANN potential.

Bonded interactions can be calculated for molecules. This feature was essentially introduced to provide an alternative, more flexible way to keep molecules together instead of the rather rigid FNC constraint. Hopefully it will provide a physically more realistic distribution of the bond lengths, angles and dihedral angles. These interactions can be calculated only if the Fixed Neighbour Constraint (FNC) switch is set to 4, so the normal FNC constraint and the new flexible molecule handling cannot be used together. Furthermore, as the same class is handling the potential and the conventional FNC in the program, normal FNC (1, 2, 3) cannot be used, together with non-bonded potential calculation. The molecular topology has to be given in the **.top* (or **.itp*) file.

The bonded interactions consist of bond stretching, angle bending and three differently defined dihedral potentials. The contributions of the bonded interactions are almost exclusively positive, as the potential functions are defined the way that the minimum value will be zero. Therefore if only bonded interactions are present, then they give a contribution to normal χ^2, the larger their energy is the larger the χ^2 so guiding the system to reach configurations closer to the favourable equilibrium value. However lately it came to light that in case of Ryckaert-Bellemans dihedrals the GROMACS-related dihedral potential for some dihedral types can be negative, so in this case depending on the configuration and the bonded potential weighing it can happen that the potential terms have a negative total chi2 contribution. This is not allowed, so the program terminates. In this case use non-bonding potential as well (as this way all the potential terms contribute to $\chi_P^{2)}$, even with so large sigma values that the non-bonded contribution does not influence the simulation.

If non-bonded interaction is present, then the bonded interaction contribution is added to the χ_P^2 forming a purely energy-based contribution, if not, then to the normal χ^2. Their contribution is calculated similarly to the non-bonded interaction ($\chi_B^2 = V_B/\sigma_B^2$). It is possible to have only one contribution coming from the given interaction type (for example bonds), if for each of the different bond types the same sigma value is given in the **.top* (or **.itp*) file. If different sigma values are given for the different bond types, then each will give its own contribution to χ^2 and displayed separately.

Although in GROMACS the interaction parameters like the LJ sigma and epsilon or the force constant and equilibrium values for the bonded interactions come usually form data base files and do not have to be specified in the topology for RMC they have to be specified directly in the topology file. This way other force field parameters can be used easily, but it have to be kept in mind that the parameters have to be consistent.

If non-bonded interactions are present, then first the χ^{P2} is calculated from the potential related contributions, and based on this it is decided, whether the move is acceptable or not. If χP2 decreased, then the move is accepted, if not then it is accepted with exp[-(χPnew2 –χPold2)] probability. If there are "tooclose" atoms, and the move decreased its number or at least increased the distance between "tooclose" atoms, then the move is accepted regardless the χP2 change, similarly as in case of the normal χ2 based acceptance procedure. If the move passed this test, then the data set and constraint calculations and the normal χ^2 calculation follows, and a second acceptance test based on the normal χ^2.

As usually potential-using RMC calculations are started from equilibrium MD configurations, our aim is not to decrease further the total potential energy, but to prevent it to move away from its equilibrium value. Therefore it is possible to give a lower limit for the χ_P^2 under which the total potential energy cannot decrease. This can be done by specifying negative value for weight mode in line 64 of the example **.dat* file, and give a fraction value (χ_P^2*_low_lim)* in line 69. The given fraction of the initial value of the total potential related chi square, χ_P^2 will be used to calculate the lower limit of χ_P^2 the following way:

if $\chi_P^2 > 0$, the lower limit will be χ_P^2_low_lim*χ_P^2;

if $\chi_P^2 < 0$, then χ_P^2 +(1-χ_P^2_low_lim)*χ_P^2 ensuring that χ_P^2 will not decrease under this value.

The substantial increase of the χ_P^2 can be prevented by adequately small sigma parameter(s) for the potential related components. On the other hand the sigma parameters has to be large enough to let an appropriate number of potential increasing moves be accepted, as otherwise, the simulation will not be able to proceed.

See C about the 'tooclose' atom counting for potential runs applying FNC=4 (flexible molecules). It has to be noted that in case of fnc=4 moveout option has to be applied with care, as starting from md configuration, a lot of 'tooclose' atom can occur (you can see it in the **start.tca* file), which would prevent the decrease of the χ^2, so either do not use moveout option in this case, or set the cut-offs smaller.

From version 2.3 there is a possibility to promt the program to set the cut-off automatically to the smallest occurring distance for each atom pair, if the CUT-OFF = MIN is used in the free format **.dat* file. It has to be noted that this automatic CUT-OF determination should be used with great care in case of flexible molecule runs, as setting the cut-off distances too high would exclude a large portion of the configuration space.

Q.1.1. The interaction functions

(1)Non-bonded interactions

The non-bonded interactions potential consists of a van der Waals term and a Coulomb term. Presently only Lennard-Jones 6-N potential is included for the van der Waals term, *N* can be specified in the **.dat* file, if not given, the default 12 is used.

The potentials are calculated according to the formulas given below:

$$V_{LJ}(r_{ij}) = 4\varepsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^N - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^6\right) = \frac{C_{ij}^{(N)}}{r_{ij}^N} - \frac{C_{ij}^{(6)}}{r_{ij}^6} \qquad \text{E 24}$$

$$V_{Coulomb}(r_{ij}) = f\,\frac{q_1 q_2}{r_{ij}}$$

where ε_{ij} (kJ/mol) and σ_{ij} (Å) or alternatively $C_{ij}^{(N)}=4\varepsilon_{ij}\sigma_{ij}^N$ are the LJ parameters, which should be supplied for all different GROMACS-type of atoms in the **.dat* file, $f=1/(4\pi\varepsilon_0)$.

(2)Bonded-interactions

For molecules the bond stretching potential between atom *i* and *j* can be calculated as a harmonic potential:

$$V_b(r_{ij}) = \frac{1}{2}k_{ij}^b\left(r_{ij} - b_{ij}\right)^2 \qquad \text{E 25}$$

where the k_{ij}^b is the force constant in kJ/mol/nm^2, *b* is the equilibrium distance in nm should be given in the topology..

The harmonic angle bending potential for atoms *i*, j and *k* (*j* is in the middle) is defined as:

$$V_a(\theta_{ijk}) = \frac{1}{2}k_{ijk}^a\left(\theta_{ijk} - \theta_{ijk}^0\right)^2 \qquad \text{E 26}$$

where the k_{ijk}^a is force constant in kJ/mol/rad^2, θ^0 is the equilibrium angle in radian.

Dihedral angle can be proper or improper. Proper dihedral angle is defined according to the IUPAC/IUB convention as the angle between the planes *ijk* and *jkl* (see Figure 6) with zero corresponding to the cis conformation.

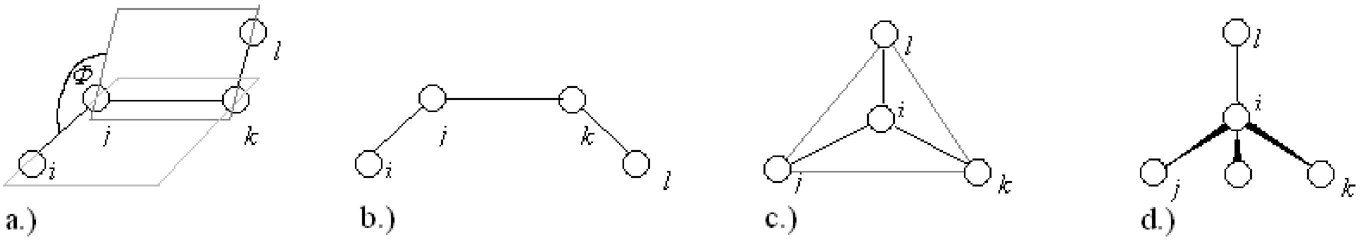


Figure 6: a.)Proper dihedral angle, b.) improper dihedral for rings, c.) planar group and d.) chiral centre.

Improper dihedrals are used to keep planar groups planar, or to prevent chiral groups to transform to their mirror image.

Three different dihedral angle definition is possible based on GROMACS, the periodic, harmonic, and Ryckaert-Bellemans dihedral definition. It depends on the force field, how the proper and improper dihedrals are calculated.

Usually periodic dihedral is used to handle the proper dihedral interaction, and calculated as:

$$V_{pd}(\Phi_{ijkl}) = k^{\phi}\left(1 + \cos\left(n\Phi_{ijkl} - \Phi^s\right)\right) \qquad \text{E 27}$$

where k^{Φ} is the force constant in kJ/mol, *n* is the multiplicity and Φ^S is the equilibrium angle in degree.

Improper dihedrals are defined as harmonic dihedrals, and calculated as:

$$V_{hd}(\xi_{ijkl}) = \frac{1}{2}k_{ijkl}^{\xi}\left(\xi_{ijkl} - \xi^0\right)^2 \qquad \text{E 28}$$

where k^{ξ} is the force constant in kJ/mol/rad^2, and ξ^0 is the equilibrium angle value.

For alkanes the Ryckaert-Bellemans dihedral definition is often used:

$$V_{RBd}\left(\psi_{ijkl}\right) = \sum_{n=0}^{5} C_n(\cos(\psi_{ijkl}))^n \qquad \text{E 29}$$

where $\psi_{ijkl}=\phi_{ijkl} - 180°$ 0 corresponding to the trans conformation,□ and C_n in kJ/mol are the RB constants.

The OPLS force field is using Fourier dihedrals with ϕ=0 corresponding to cis:

$$V_{Fd}\left(\phi_{ijkl}\right) = \frac{1}{2}\left[F_1(1+\cos(\phi)) + F_2(1-\cos(2\phi)) + F_3(1+\cos(3\phi)) + F_4(1-\cos(4\phi))\right] \qquad \text{E 30}$$

which can be converted into RB dihedrals, as
$C0=1/2(F_1+F_3)+F_2$
$C_1=-1/2(F_1-3F_3)$
$C_2=4F_4-F_2$
$C_3=-2F_3$
$C_4=-4F_4$
$C_5=0$
so the literature values has to be treated accordingly. The program internally uses RB dihedrals.
In case of the OPLS force field improper dihedrals are not used for chiral atoms, and for planar groups the periodic E 27 function is used as improper dihedral angle function.

(3)Force fields in RMC

The parameters for the bonded and non-bonded interactions are determined by the force field used in the simulation. Only all-atom force fields can be used for RMC simulations! There are several force fields supported by GROMACS, the list is changing with the GROMACS version. RMC mainly focused and still focuses on OPLSAA, Encadv and Encads, the last two is no longer supported natively by GROMACS. Presently the following force fields are recognized (all the all-atom force fields supported before and the new ones):

NAME	Combination rule	Generate pairs for 1-4 interaction
OPLSAA	3	yes
encads, encadv	3	yes
AMBER*	2	yes
CHARMM*	2	yes
UNKNOWN_GENERATE_PAIRS	X	yes
UNKNOWN_READ_PAIRS	X	no

Table 1. The force fields recognized by RMC.

The main characteristics of a force field is what kind of potentials it is using (currently only LJ and Coulomb is implemented for non-bonding potential in RMC), it is regulated by the potential=1 option in the **.dat*. The other important parameter is how the valued of the non-bonded potential for the mixed partials is calculated, this is regulated by the combination rule (also given in the **.dat* file), see later in Q.1.2(5).

How the LJ potential parameters for the 1-4 interactions are handled (see Q.1.2(3)) is decided by the generate_pairs option in the [defaults] section of the topology for GROMACS. RMC does not read this directive, and does not need and handle the force field files. Whether to generate pairs is decided based on the name of the force field, which is in the include file of the force field (for example #include “oplsaa.ff/forcefield.itp” or in the old format “ffoplsaa.itp”. If no known force field is used (which is possible, as the non-bonding parameters are read anyway from the **.dat*), you should include either “ffUNKNOWN_GENERATE_PAIRS.itp” or “ffUNKNOWN_READ_PAIRS.itp”.

Q.1.2. The description of the GROMACS-type topology, and the connection to RMC

The same format text type RMC configuration file **.cfg* and/or the binary format **.bcf* files are used, as before. Here the concept is that for a multi-component system (which can be atomic or can contain molecules) usually one RMC-type depicts a chemical element, so there are as many RMC-types, as the number of different chemical elements. The coordinates of the atoms belonging to the same RMC-type are making up a block, and these blocks follow each other consecutively in the order the types are given in the file header. The partial *ppcf*-s are calculated based on this type selection. When a bit more complicated system is used, (for example propanol), it might be necessary to use more than one RMC-type for the same chemical element depending on the location of the atoms in the molecule to be able to hold the molecule together properly with a Fixed Neighbour Constraint (FNC).

In molecular dynamics simulation programs (as GROMACS) a lot of different type (referred to as GROMACS-types) are used for the same chemical element depending on the environment of the atom, as the non-bonded parameters are strongly dependent on the atomic environment.

Even in that case we might want to use only one RMC-type for a chemical element, as the neutron or X-ray or electron diffraction properties are the same. Therefore, the rendering of the GROMACS-atom indices to the RMC atom indices has to be given in the topology file. The parameters for the non-bonded interactions for each GROMACS-types has to be given in the **.dat* file, and the weight parameter for the potential-based χ^2 contribution as well.

There is a difference between GROMACS and RMC in handling systems with more than one molecule type, if the order of the molecule types after the [molecules] directive differs from the order the molecule types are described in the topology by the [moleculetype], [atoms] ..(included from *itp* files). While using GROMACS the order of the molecule types in the **.gro* file has to match the order after the [molecules] directive and this will be the order used by the program, so does not matter, in which order the **.itp* files are included, RMC will store the GROMACS types in the order read from the [moleculetype], [atoms] ..(the order of the **.itp* files). This is very important, as the vdW parameters has to be given in **.dat* following this order! The order RMC stored the GROMACS types can be seen from the screen output and the **.hst* file as well, so it is a good idea to check it. The best way however to use the same order in **.gro*, after the [molecules] directive and the order the molecules described in the topology to avoid confusion!

As we use the GROMACS molecular dynamics suite, the same format topology files for describing the molecules are used in RMC_POT with some additional parameters, but the additions are always given at the end of a line, so this extended topology file can be used for both GROMACS and RMC, which simplifies our work. It might be useful to consult the GROMACS manual, which is available from the GROMACS web site http://www.gromacs.org/Documentation/Manual, as in this documents only the basics of the topology is given. It is advisable to start a simple MD GROMACS simulation with your newly created topology, as GROMACS performs more extensive checks regarding the integrity of the topology file than RMC does!

(1)The concept of charge groups

Non-bonded interactions are calculated up to a cut off value. Therefore for molecules containing atoms with partial charges it would create a problem, if some of the atoms would fall below cut off and other above it. This is why nearby atoms are rendered to charge groups. Preferably the net charge of the charge group should be 0, but this not always possible. For example the three atoms of a water molecule belong to the same charge group with 0 net charges. For larger molecules it is not possible to render all the atoms of the molecule to the same charge group, here a chemically meaningful part of a molecule form a charge group, as for example a methyl ($-CH_3$) or methylene ($-CH_2$) group. The atoms of a charge group are not separated from each other by the cut off during non-bonded interaction calculation, the geometric centre of the charge group decides for all the atoms of the group, whether it is inside or outside cut off. The atoms of a charge group has to follow each other in the molecule's **.top* or **.itp* file having the same charge group number.

(2)The exclusions

The interactions of the atoms connected to each other by 1-2 chemical bonds are handled quantum-mechanically by the bond stretching (1 bond) or angle bending (2 bonds). Therefore they have to be excluded from the normal non-bonded interaction. This is achieved by setting the **number of exclusions** parameter in the topology file, and handled by RMC exactly the same way, as by GROMACS.

After the introduction of the virtual sites the handling of the exclusions also by the [exclusions] directive was implemented as well. If an exclusion is defined by the molecule connectivity with the number of exclusions and later in the [exclusions] section as well it will be included only once in the exclusion list in the program. Exclusions involving virtual sites has to be given explicitly in the [exclusions] section, as these cannot be determined from the connectivity!

(3)The 1-4 interactions

For the atoms separated by 3 bonds (1-4 neighbours) usually the normal non-bonded interactions are too strong. The handling of these pairs depends on the parameterisation of the force field, but usually a different set of parameters are used for the 1-4 interactions (as in GROMOS force field) or the normal parameters are scaled as in OPLS. Whether to use 1-4 interactions together with dihedral angle interaction also depend on the applied force field and the dihedral interaction type, for example for GROMOS force field if the periodic dihedral potential is used, then a special 1-4 interaction has to be included (the LJ1-4 parameters has to be given for RMC in the topology file after the [pairs] directive), if RB dihedrals are used for alkanes, then no 1-4 interaction has to be calculated at all. In either case, the 1-4 interaction has to be excluded from normal non-bonded interaction calculation by setting the number of exclusions to 3. If OPLS-aa force field is used, then all the 1-4 interactions (both the vdW and the Coulomb) is scaled down by a factor of 2, so no additional parameters are required and together with this RB dihedrals has to be used. The 1-4 atom pair indices has to be given after the [pairs] directive, if 1-4 interaction has to be calculated either by using special parameters or by scaling!

I believe that in case of a virtual site is carrying the charge or the van der Waals parameters instead of an atom involved in 1-4 interactions, then the virtual site has to be included in the pair list as well, although no information was available about this in the GROMACS manual.

If scaling is used, then it can be given as the vdW and Coulomb fudge in the **.dat* file for RMC, for OPLS it has to be 0.5 for both Lennard-Jones and Coulomb interaction.

(4)The virtual sites

The program can handle virtual sites implemented in GROMACS for handling polarization, see the GROMACS manual for details! In RMC_POT virtual sites can be used to carry charge or have van der Waals parameters! The virtual sites has to be given in the [exclusions] section if necessary, as they cannot be deduced from the connectivity!

The virtual sites has to be included in the [atoms] section of the molecular topology with their own GROMACS type, and their charge. For GROMACS they always have to have zero mass! RMC indices for them have to be set to 0, as this indicates to RMC_POT that these are not real atoms! Then they have to be described in the [virtual_sitesX] (X=2,3,4) section of the topology, as for GROMACS.

Take care in which order the atoms building the virtual sites are specified in the [virtual_sitesX] section! As Figure 4.18 showing the different virtual sites in the 2016 edition of the GROMACS manual did not put indices on the pictures, see Figure 7 below. For detailed description of the virtual sites consult the GROMACS manual!

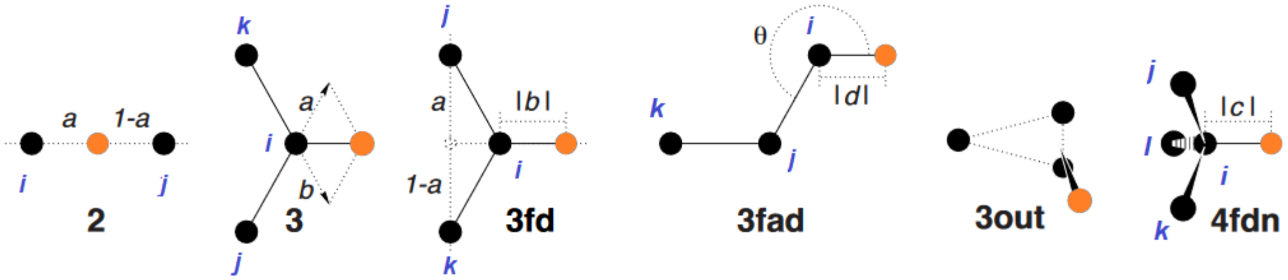


Figure 7: The 6 different virtual sites implemented in RMC.

The virtual sites has their own GROMACS type (there should be more than one GROMACS type for them, if they have different van der Waals parameters, if only the charges are different, then one GROMACS type for them is enough), so the number of GROMACS types has to be set in the **.dat* file accordingly, and their van der Waals parameters have to be given there as well.

In RMC the potential contributions are divided into RMC partials during the calculations, so each virtual site has to have an RMC type assigned to it, as its potential contribution will be included in those partials, which the RMC type specifies! Normally a virtual site serves as a shifted position for one of the atoms (for example for oxygen in the water models) to account for polarization. If no RMC type is given, then the GROMACS serial index of the first atom building the virtual site (atom *i*) is assumed. If other RMC type is preferred, then its atom index should follow the semicolon at the end of the normal entries of the line in the [virtual_sitesX] section. Obviously for the total potential energy it does not matter, which partial the virtual site contributes to.

(5)The combination rule, and the non-bonded parameters for the mixed partials

There are different ways the non-bonded parameters can be specified similarly to GROMACS. For the LJ interaction always two parameters are required for an atom type, (denoted generally by *V* and *W*), which can be either the σ - ε pair, or the $C^{(6)} - C^{(N)}$. The parameters for the mixed partials can either be directly specified if their calculation does not follow any rule, or calculated according to some formulas, where either arithmetic or geometric averages can be used. The combination rule given in the **.dat* file for RMC decide about the meaning of the given parameters, and how the vdW parameters for the mixed partials has to be calculated, if they are not supplied. These are the following options:

Combination rule	*Supply V, W*	Formula
0	$C_{ij}^{(6)}$ [kJ·Å^6/mol], $C_{ij}^{(N)}$ [kJ·Å^{12}/mol]	given for all the partials
1	$C_i^{(6)}=4\varepsilon_i\sigma_i^6$ [kJ·Å^6/mol] $C_i^{(N)}=4\varepsilon_I\sigma_i^N$ [kJ·Å^N/mol]	$C_{ij}^{(6)}=\left(C_i^{(6)}C_j^{(6)}\right)^{1/2}$ $C_{ij}^{(N)}=\left(C_i^{(N)}C_j^{(N)}\right)^{1/2}$
2	σ_i [Å], ε_i [kJ/mol]	$\sigma_{ij}=\frac{1}{2}\left(\sigma_i+\sigma_j\right)$ $\varepsilon_{ij}=\left(\varepsilon_i\varepsilon_j\right)^{1/2}$
3	σ_i [Å], ε_i [kJ/mol]	$\sigma_{ij}=\left(\sigma_i\sigma_j\right)^{1/2}$ $\varepsilon_{ij}=\left(\varepsilon_i\varepsilon_j\right)^{1/2}$

Combination rule 1-3 are the same in GROMACS (only the unit for distance is nm), 0 only exist in RMC to be able to handle force fields (like gmx or gromos), where the pair parameters are explicitly given, in this case the $C_{ij}^{(6)}$ and $C_{ij}^{(N)}$ values for all the GROMACS type partials has to be supplied in the **.dat* file in the right order. (Similarly as for RMC partials, but using the GROMACS types, for example 1-1, 1-2, 1-3, 2-2, 2-3, 3-3).

(6) The topology (.top) and include topology (*.itp) file*

The topology file contains the information about the system, how many different molecule types can be found, and how many molecules exist for each type. This has to be in the **.top* file. The definition of the different molecule types can go into their own **.itp* file, or if there is just one molecule type into the **.top* file itself. The **.itp* files can be included into the **.top* file with #include *itpfilename.itp*. All the C-Preprocessor directives can be used in a topology file for GROMACS, but only the following ones are recognised by RMC: #include, #ifdef, #ifndef, #else, #endif. The last three can help to create topology files, where the active content can be regulated by passing or not passing compiler options through the **.dat* file. For example if a part of a topology file is

```
[ atoms ]
1    opls_209  1  DMeS  CA    1  -0.013            ;  1   2
#ifdef _DEUTERIUM
2    opls_140  1  DMeS  DA1   1   0.06  2.014102   ;  31  34
3    opls_140  1  DMeS  DA2   1   0.06  2.014102   ;  32  35
4    opls_140  1  DMeS  DA3   1   0.06  2.014102   ;  33  36
#else
2    opls_140  1  DMeS  HA1   1   0.06             ;  31  34
3    opls_140  1  DMeS  HA2   1   0.06             ;  32  35
4    opls_140  1  DMeS  HA3   1   0.06             ;  33  36
#endif
```

If _DEUTERIUM is passed to the topology file, then the blue part using deuterium atoms are used, if not then the green part using normal hydrogen atoms. Arguments can be passed to the topology file after the FNC switch preceded with –D like -D_DEUTERIUM.

Topology file has to be given even for atomic or ionic system, in this case only the [moleculetype] and [atoms] directive has to be used for the molecule description. The available directives are discussed below.

(7)The GROMACS directives

GROMACS topology files are directive driven, the directives has to be specified in square brackets. The order of some directives is important, while others can be arbitrary. The lines following a directive are supposed to be of the type indicated by the directive and treated accordingly, while other directive is found. Empty lines can be anywhere, and they are ignored. RMC uses only some of the existing GROMACS topology directives, the others are ignored. It has to be noted that the semicolon is treated by GROMACS as text qualifier, and everything after it is ignored, so additional information which is required by RMC are sometimes given after that, as RMC, if expecting an additional entry for that line ignores the semicolon, but at the beginning of a line treats it as text qualifier, and do not try reading after it. Comment lines beginning with text qualifier can be anywhere in the file.

The directives recognised by RMC are the following:

Recognised GROMACS directives	Format
[moleculetype]	**molecule name**, **number of exclusion, (index offset)**
[atoms]	serial number**, type**, residue number, residue name, **atom name**, **charge group**, **charge (e)**, mass, **first index**, **second index**
[pairs]	**atom index i**, **atom index j, type, {$V^{(cr)}$}, {$W^{(cr)}$}**
[bonds]	**atom index *i*, atom index *j*, type, *b* (nm), k^b(kJ/mol), (sigma)**
[angles]	**atom index *i*, atom index *j*, atom index *k*, type, θ^0(degree), k^a (kJ/mol), (sigma)**
[dihedrals]	**atom index *i*, atom index *j*, atom index *k*, atom index *l*, 1, Φ^S (degree), k^Φ (kJ/mol), multiplicity, (sigma)**
[dihedrals]	**atom index *i*, atom index *j*, atom index *k*, atom index *l*, 2, ξ^0 (degree), k^ξ (kJ/mol/rad^2), (sigma)**
[dihedrals]	**atom index *i*, atom index *j*, atom index *k*, atom index *l*, 3, C_0, C_1, C_2, C_3, C_4, C_5, (kJ/mol), (sigma)**
[exclusions]	**atom index for the exclusion centre, one or more indices for the excluded atoms**
[virtual_sites2]	**vsite index, two atom indices building the site, function type:always 1, parameter a, *host atom index***
[virtual_sites3]	**vsite index, three atom indices building the site, function type: 1 for type “3”,parameter a, b, *host atom index***
[virtual_sites3]	**vsite index, three atom indices building the site, function type: 2 for type “3fd”,parameter a, d (nm), *host atom index***
[virtual_sites3]	**vsite index, three atom indices building the site, function type: 3 for type “3fad”,parameter theta (deg), d (nm), *host atom index***
[virtual_sites3]	**vsite index, three atom indices building the site, function type: 4 for type “3out”,parameter a, b, c (nm^{-1}), *host atom index***
[virtual_sites4]	**vsite index, four atom indices building the site, function type: 2 for type “4fdn”,parameter a, b, c (nm), *host atom index***
[system]	system name
[molecules]	**molecule name, number of molecules**

Bold underlined parameters are only used by RMC, has to be preceded by a semicolon, if the topology is used for GROMACS. If it is in (), like the sigma parameter, then has to be given only for the first occurrence of the given interaction type. For the index offset it is necessary only for the second and higher indexed molecule types, if there is more, than one molecule type. ***Bold underlined italic*** parameters are optional for RMC.

Bold parameters are used both by GROAMCS and RMC**.**

Parameters with normal letters are only used by GROMACS, but value matching the data type of the parameter has t be given for RMC as well, only the value does not matter.

Parameters in {} are needed only depending on the force field type, for OPLS-aa it is not required.

Some of the parameters are explained below:

[moleculetype]

This can be in the **.itp* file.

- The **molecule name**'s size is set by the NAME_SIZE define statement in the *units.h*, and the default value is 50, in the GROMACS manual the maximum size for GROMACS was not found, but 50 characters is accepted as well.
- Atoms separated by the number of bonds**(= number of exclusions**) are excluded from the normal LJ interactions.
- Index offset is the total number of atoms before this molecule type according to the GROMACS configurations. This way the same **.itp* with the same first and second index can be used either it is alone in the system or together with other molecule types. First and second index has to be changed only if the number of molecule is changed.

[atoms]

This can be in the **.itp* file.

- **type** is the GROMACS atom type of the atom, for RMC it can be anything, but has to be the same for the same type of atoms;
- **atom name** is the unique identifier of the atom max 5 character;
- index of the **charge group;**
- first index is the RMC index of the atom in the first occurrence of the molecule;
- second index is the RMC index of the atom in the second occurrence of the molecule;

[pairs]

This can be in the **.itp* file.

Here the atom pairs for which 1-4 interaction has to be calculated can be given:

- **atom index i** and **atom index j** are the GROMACS serial indices of the atoms in the topology file;
- **type** is the GROMACS interaction type, has to be set to 1 for both RMC and GROMACS for the normal 1-4 interaction discussed above;
- $\mathbf{V^{(cr)}}$**,** $\mathbf{W^{(cr)}}$ **are vdW parameters** according to the combination rule**,** see (5).

There are usually predefined bonds, angles, dihedrals between given type of atoms, check the force field's *ff*bon.itp* file for the parameters.

[bonds]

This can be in the **.itp* file.

- **atom index *i***, **atom index *j*** are the GROMACS serial indices of the atoms in the topology file;
- **type** is the GROMACS interaction type, should be 1 for harmonic bond potential, RMC try to proceed, even it is not;
- ***b* (nm)** is the equilibrium distance, not necessary for GROMACS as it comes from force field files, but has to be given before or after semicolon for RMC;
- $\mathbf{k^b}$ **(kJ/mol)** is the force constant**,** not necessary for GROMACS as it comes from force field files, but has to be given before or after semicolon for RMC;

- **sigma** parameter to weight the contribution of the bond type for χ^2 calculation, has to be given only for the first occurrence of a bond type; Bonds with the same equilibrium bond and force constant belong to a given bond type.

[angles]

This can be in the **.itp* file.

- **atom index *i*, atom index *j* and atom index *k*** are the GROMACS serial indices of the atoms in the topology file (*j* being the middle atom);
- **type** is the GROMACS interaction type, should be 1 for harmonic angle potential, RMC try to proceed, even it is not;
- θ^0 **(degree)** is the equilibrium angle, not necessary for GROMACS as it comes from force field files, but has to be given before or after semicolon for RMC;
- k^a **(kJ/mol)** is the force constant**,** not necessary for GROMACS as it comes from force field files, but has to be given before or after semicolon for RMC;
- **sigma** parameter to weight the contribution of the angle type for χ^2 calculation, has to be given only for the first occurrence of an angle type. Angles with the same equilibrium angle and force constant belong to a given angle type.

[dihedrals]

This can be in the **.itp* file.

- **atom index *i*, atom index *j*, atom index *k* and atom index *l*** are the GROMACS serial indices of the atoms in the topology file (atoms following each other in the chain in alphabetical order);
- **type** is the GROMACS interaction type, 1 for periodic, 2 for harmonic and 3 for Ryckaert-Bellemans dihedral potential, RMC quit, if it is neither of them.
- θ^0 or ξ^0 **(degree)** are the equilibrium angle depending on the dihedral type, not necessary for GROMACS as it comes from force field files, but has to be given before or after semicolon for RMC;
- k^Φ **(kJ/mol)** or k^ξ **(kJ/mol/rad^2)** are the force constant depending on the dihedral type**,** not necessary for GROMACS as it comes from force field files, but has to be given before or after semicolon for RMC;
- C_0**,** C_1**,** C_2**,** C_3**,** C_4**,** C_5**, (kJ/mol)** are the parameters for the Ryckaert-Bellemans potential, not necessary for GROMACS as it comes from force field files, but has to be given before or after semicolon for RMC;
- **sigma** parameter to weight the contribution of the dihedral type for χ^2 calculation, has to be given only for the first occurrence of a dihedral type; Dihedral angles with the same type, and other parameters belong to a given dihedral angle type.

[exclusions]

This can be in the **.itp* file

- **atom index is** the GROMACS serial index for the atom we are giving the exclusions for,
- one or more GROMACS serial indices for the **excluded atoms**

[virtual sites2]

- GROMACS serial index of the virtual site in the [atoms] section
- **atom index *i*** and **atom index *j*** are the GROMACS serial indices of the atoms building the virtual site
- **function type**, can only be 1
- parameter a (scale factor for $\mathbf{r}_{\mathbf{ij}}$, no unit)
- Optional: GROMACS serial index for **host atom**, whose RMC type defines, which partials the potential contributes to, if not given, then RMC type of atom index *i* is assumed!

[virtual sites3]

- GROMACS serial index of the virtual site in the [atoms] section
- **atom index *i*, atom index *j* and atom index *k*** are the GROMACS serial indices of the atoms building the virtual site

- **function type 1 for type “3”, 2 for type “3fd”, 3 for type “3fad”, 4 for type “3out”**
- parameters: for function type 1: a, b (scale factors, a for $\mathbf{r}_{ij}$, b for $\mathbf{r}_{ik}$ no unit)
  for function type 2:a (scale factor), d (nm)
  for function type 3:theta (deg), d (nm)
  for function type 4:a, b (scale factors a for $\mathbf{r}_{ij}$, b for $\mathbf{r}_{ik}$), c (scale factor for the cross product of $\mathbf{r}_{ij} \times \mathbf{r}_{ik}$ nm^{-1})
- Optional: GROMACS serial index for **host atom**, whose RMC type defines, which partials the potential contributes to, if not given, then RMC type of atom index *i* is assumed!

[virtual sites4]

- GROMACS serial index of the virtual site in the [atoms] section
- **atom index *i*, atom index *j*, atom index *k* and atom index *l*** are the GROMACS serial indices of the atoms building the virtual site
- only **function type 2** for type “4fdn” is implemented
- parameters: a, b (scale factors a for $\mathbf{r}_{ik}$, **b for** $\mathbf{r}_{il}$), c (nm)
- Optional: GROMACS serial index for **host atom**, whose RMC type defines, which partials the potential contributes to, if not given, then RMC type of atom index *i* is assumed!

[system]

Has to be in the **.top* file, the molecule type definitions should precede this.

- system name is not relevant for RMC

[molecules]

Has to be in the **.top* file following the [system] directive.

- **molecule name** has to be the same declared with the [moleculetype] directive.
- **number of molecules**

(8)Rendering the RMC configuration to the topology

The structure of the text type RMC configuration file was not changed. As a reminder the structure of the **.cfg* file follows the simple rule that the total number of atoms, the number of atom types, the half-length of the simulation box and the number of atoms for each type are specified in the header of the **.cfg* file, then all the coordinates of the first type of atoms (each a separate line), then the coordinates of the second and so on atom types can be found consecutively. Because of this, if there are molecules consisting of different type of atoms, then the coordinates of the atoms belonging to the same molecule can be found in non-consecutive lines of the file. Even if there are more than one atom from the same RMC type in the molecule, they not necessarily located consecutively, but sometimes in blocks, depending on the structure of the **.cfg* and the accompanying **.fnc* file, if no potential is used. The only important thing is that has to be logic in the arrangements of the atoms, so if we know the atom indices of the atoms belonging to the first and second molecule, we can calculate the atom indices of the atoms in any molecule. The coordinates are reduced, going between –1 and +1 and the box half-length and other distance related properties are given in Ångstrom in RMC.

For example there are two different representations, two sets of **.cfg* with their matching **.fnc* file for a normal simulation, where no topology and potential is used of SnI4, see the atom indices in Table 2.

Table 2: The indices of the atoms in the two differently structured **.cfg* and **.fnc* file for SnI4.

Atoms	First representation		Second representation	
	Atom indices		Atom indices	
	first mol.	second mol.	first mol.	second mol.
Sn	1	2	1	2
I	1001	1005	1001	1002
I	1002	1006	2001	2002

I	1003	1007	3001	3002
I	1004	1008	4001	4002

On the other hand in GROMACS the distances and coordinates are given in nm, and the simulation box goes from 0→ box length in the **.gro* file, but this is not relevant for RMC.

In GROMACS atoms have a name and a type. It has to be emphasized that the GROMACS atom types should not be confused with the atom types in RMC. In GROMACS atoms of the same chemical type can have different GROMACS types, as the GROMACS type have to reflects the chemical role, bonds, hybridisation state of the atom causing it to have different Lennard-Jones parameters or charge. Each force field has its own defined different GROMACS atom types. You have to use one of the atom types of the force field you chose, if you want to use the topology for GROMACS as well, so check the *ff*.atp* file of the force field for the available types in the top directory of the GROMACS installation. If the topology is only used for RMC, then of course one does not have to stick to the atom types of a force field, any type name can be used, but always use a consistent set of parameters! The atom names can be freely chosen (max 5 character).

Q.1.3. Using tabulated non-bonding potential

If only non-bonding potential is needed, then it can be accomplished without using a topology file. Tabulated potential for all, or only for some selected RMC partials can be supplied. The further advantage that this way interaction functions currently not included in RMC_POT can be used as well. Obviously either the GROMACS-like OR the tabulated non-bonding potential can be used!

For the potential switch 10 has to be specified (see line 64 in **.dat*), and the *r* [Å] – *U(r)* [kJ] values for an RMC partial has to be given in a file, see section for the details of the file structure. The maximum number of the tabulated potential files is the number of RMC partials. Each tabulated potential partial has its own sigma parameter, and displayed separately during the simulation and in the **.chi* and **.hst* files. The U/kT for each partial is also displayed during the run, but does not play an algorithmic role. Therefore the temperature in K has to be supplied in the **.dat* file (see line *66b*).

Similarly to the previously described GROMACS-like non-bonding potential, the tabulated potential is contributing to the χ_P^2see section I.Q for details.

Tabulated potential cannot be used with fnc=1, 2, 3, similarly to the normal topology based non-bonding potential.

On the other hand tabulated potential can be used together with flexible molecules (fnc=4), but it has to be done with caution. It has to be kept in mind that in case of the tabulated potential there are no charge groups, which causes that some atoms of a molecule can be inside the cut off and the remaining outside, which results in a total artificial charge for the system part inside the cut off, and numerically there would be a difference from the correct one beside the tabulation. The bigger problem is that in this case no exclusions are considered, and the 1-2, 1-3 and partly the 1-4 interaction (the later depending on the force field) are calculated twice, once normally during the bond, angle and dihedral interaction calculation, and then faultily during the tabulated potential calculation, which can cause an enormous error in the potential, even the sign of it can change.

For example in case of bis(methylthio)methane C-C, C-S and C-H partials the correct LJ potential taking into account the exclusions up to 3 bonds (and not calculating the non-bonded interactions for them) gives -7150.6, -7426.4 and -1221.8 kJ while the tabulated potential calculation results in 118740.3, 53155170.8 and 2868724376.9 kJ. In this case no error came from the lack of charge groups, as the cut off for the potential calculation was larger than the half box length.

So it is advisable to avoid the usage of tabulated potential together with flexible molecules, it is only applicable for atomic systems!

Q.2. Artificial neural network potential

The description of the ANN potential can be found on the AENET website, http://ann.atomistic.net/ here only the usage of this potential in RMC will be discussed. Here only mentioned, that the potential decomposes to potential components by atoms, and therefore easily parallelizable. In case of a parallel RMC run the potential calculation is divided among the threads.

The potential should be created (trained) by AENET. RMC should be compiled with the _AENET compiler option to be able to handle the ANN potential, in this case the static fortran libaenet.a library and two others are linked to create the executable. As AENET only runs on linux, the RMC linked with the AENET library will only be usable on linux platform.

The ANN potential calculation only available using the free format input file, section [AENET] should be included into the **.dat* file to give the ANN potential-related parameters to RMC. These are the following:

```
[ AENET ]
IND_ANN-FILE_TYPE-NAME = 2 O.15t-15t.nn O
IND_ANN-FILE_TYPE-NAME = 1 Ti.15t-15t.nn Ti
CUTOFF-AENET = 8
WRITE-ATOMIC-ENERGY = 1
AENET-STEP=1
SIGMA = 1
```

With the mandatory IND_ANN-FILE_TYPE-NAME key word first the atom type is give then the name of the file containing the ANN potential for this atom type, and then optionally the chemical symbol can be specified. This symbol is passed to the aenet library which handles the potential, as this is used to differentiate between the atoms of different types in AENET. If the symbol is not given here, then the standard chemical symbol of the element will be extracted from the symbols given after the CHEMICAL-SYMBOLS key word in the [GENERAL] section. For example as for the TiO2 system above Ti4+ and O2- is given for the CHEMICAL SYMBOLS, and Ti and O is extracted by the program, if nothing is given here.

With the optional CUTOFF-AENET keyword the cut-off for the ANN potential in Å can be specified, if smaller value should be used, as was used with the potential training. If nothing is given, then the default value given in the potential file is used.

The optional WRITE-ATOMIC-ENERGY key word controls whether the energy for the individual atoms are written into the **.en* file.

As the potential calculation is relatively time consuming, the option not to recalculate the potential energy in every step was included, this is regulated by the optional AENET-STEP, the default is presently 5. If 1 is given, then the potential is calculated in every step, as in case of the normal pair-potential implementation. It has to be mentioned, that any case only those atom’s potential is recalculated that were affected by the move. If the potential is not calculated in every step, then the potential of all the atoms, which were affected by the moves since the last energy calculation will be recalculated, so considerable speed increase can only be achieved with large steps, which on the other hand will make the simulation path not to precise in the energy space.

One of the SIGMA/SIGMA-SCALABLE/SIGMA-MASTER keywords has to be used to give the weight factor to scale the χ^2 contribution.

No other potential (bonded, flexible) can be used together with the ANN potential. As the ANN potential contains three-body correlations, coordination number and cosine distribution constraint should not be necessary to constrain the structure, but it is possible to use them.

Vibrational amplitude calculation is possible together with ANN potential if the program is additionally compiled with the _VIBR_AMP compiler option, but ANN potential cannot be used together with non-periodic boundary conditions and local invariance. Simulations with occasional no-atomic-move steps (AXS, *I(Q)* background correction and EXAFS shift) can be used with ANN potential. Also the _AENET compiler option can be used together with the _MUL_SCAT_VECTOR, _READ_EDIFF and _R_SWITCH_GR_MIN_1 compiler options.

R. Using RMC++ without periodic boundary

It is possible to simulate spherical samples (like nanoparticles) without the usage of periodic boundary conditions. The model in this case can be seen in Figure 8:

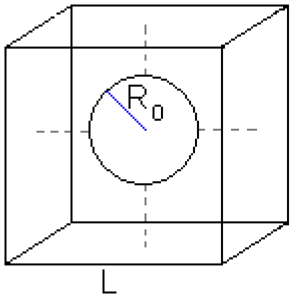


Figure 8: The location of the spherical sample with radius R_0 inside the simulation box with box length L for nanoparticle simulation without periodic boundary conditions.

To avoid the usage of periodic boundary conditions, the sample is centred in the middle of the simulation cell, and its radius is given by

$$\frac{L}{4} = R_0 - \frac{\Delta r}{4} \qquad \text{E 31}$$

where Δr is the size of the histogram bin.

In this case it has to be taken into account that the volume element for the normalization of the histogram is actually smaller for a considerable amount of particle, than the volume of the spherical shell, what is otherwise used. If near-homogenous distribution is assumed, than the normalizing volume can be analytically calculated, the details are given in Appendix 1. Applying the correct normalization the *g(r)* should behave normally, oscillating around and in larger distance tending to one.

The number density in this case is defined as $\rho = \frac{N}{V_{sample}}$, where N is the total number of atoms and V_{sample} is the volume of the spherical sample with radius R_0, so this density should be given in the **.dat* file. The radius of the spherical sample, R_0 has to be given in line 83 of the **.dat* file in Angstrom, see II.C. The half box length in the **.cfg* file should satisfy E 31, but if it is not, it will be rescaled, see below.

The program checks a few things at the beginning of the simulation, in the order they are given below, and makes adjustments, if necessary.

1. Only x_{max}=0.5 is allowed in this case, it is reset to this value if necessary.
2. It checks, if the sample radius R_0 corresponds to the number density given in the **.dat* file. If it is not, then the R_0 is reset accordingly.
3. Then it checks, whether R_0 divided by the size of the histogram bins gives a number, where the fraction corresponds to the bin shift specified in the **.dat* file. If bin shift=0, then R_0 has to be divisible by Δr.
4. Next it checks, whether the half box length satisfies equation E 31) and if it is not, it will adjust the half box length accordingly.
5. Then it checks, whether the coordinates of the atoms are centred inside the box, and centres them, if not.
6. Then it checks, whether the coordinates are inside the sample volume. If they are not, the program will rescale the coordinates according to the distance of the atom farthest from the centre, to set them inside the sphere of radius R_0. However, it DOES NOT CHECK, whether the sample has spherical shape at all! If it does not have initially spherical shape, then during the simulation after sufficient number of moves, the atoms are expected to occupy the whole sphere.

During the simulation only those moves are generated, which does not move the atoms outside the spherical sample.

S. Including the effect of the vibrational motions of the atoms

The atoms are constantly vibrating around their equilibrium position, which in crystals leads to the broadening of the diffraction peaks. Although in disordered solid materials not to mention liquids no periodic lattice positions can be assumed, the possibility of the decoupling of the structural modelling from the thermal vibrations was included in the program. If for the description of the atomic vibrations the harmonic approximation is used, then the distribution of the atomic displacements around their equilibrium positions can be described by a Gaussian function:

$$f(d) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{d^2}{2\sigma^2}\right) \qquad \text{E 32}$$

where d is the distance from the equilibrium position, and $\sigma^2 = \langle u^2 \rangle$, where $\langle u^2 \rangle$ is the mean square displacement of the atoms[9].

The program will expect the σ parameters for each atom type for the distribution in the **.dat* file (see line 84.

The correction can only take place, if the 4*σ for the narrowest Gaussian distribution is larger, than the half of the histogram bin, otherwise 99.99% of the distribution is inside the same histogram bin. If $4*\sigma > \Delta r/2$, then the original count of a histogram bin will be distributed among the neighbouring bins according to the Gaussian convolution function, which is calculated for the middle distance value of the histogram bins up to $\pm 5*\sigma_{max}$ (see Figure 9).

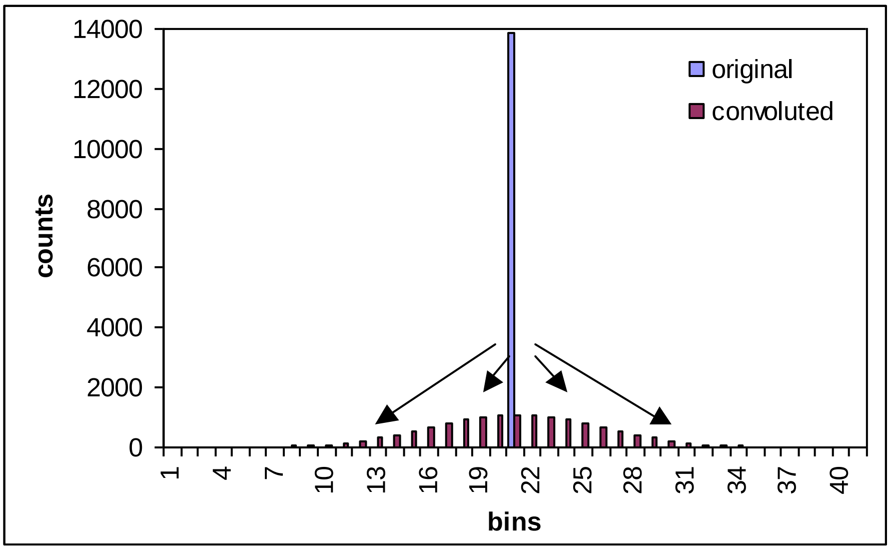
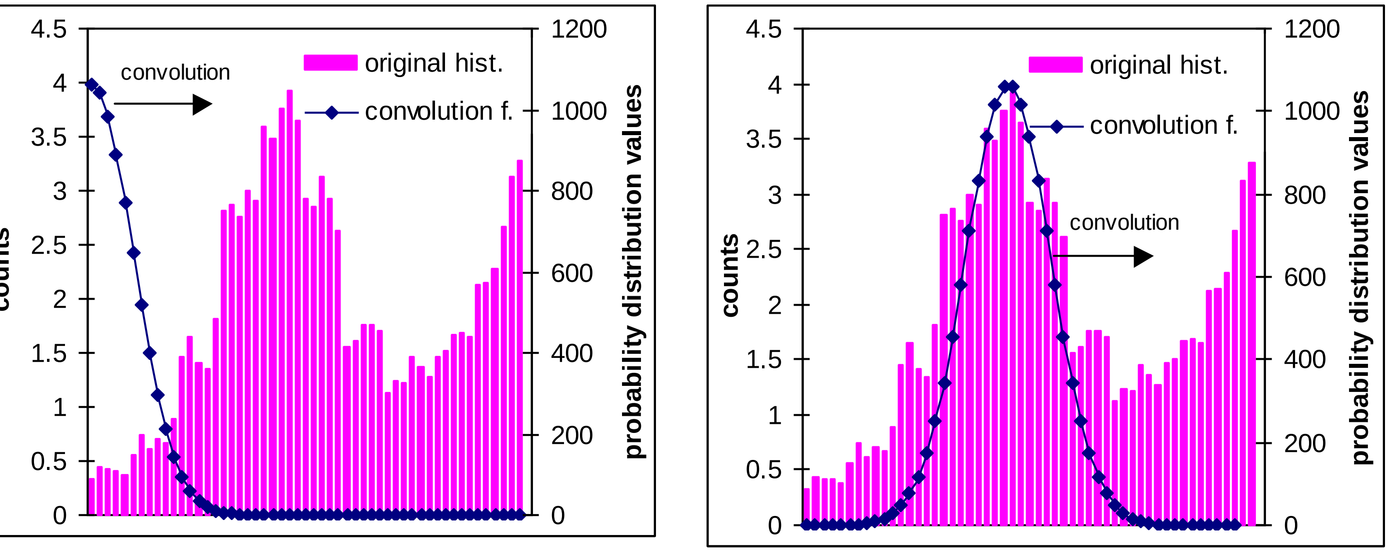


Figure 9: The distribution of the original count value of a histogram bin among the neighbouring bins.

The convolution window is sliding through all the bins creating the convoluted histogram (see Figure 10).



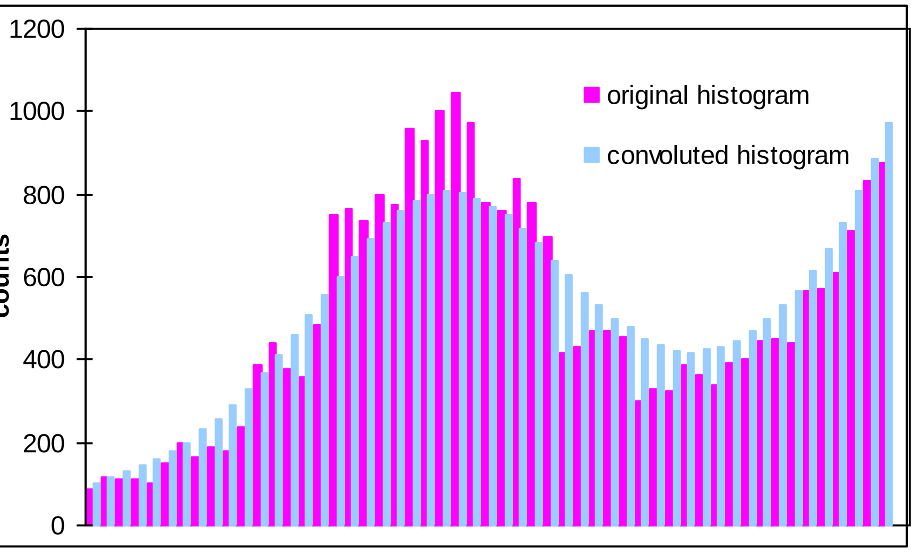


Figure 10: The convolution process.

If the convolution would put counts into non-existing histogram bins before the first (this is not likely because of the cut off values), and after the last bin, then those counts are discarded. As close to the end of the histogram the convoluted histogram bins would not receive as many counts as they should, due to the facts that counts from the non-existing histogram bins after the last bin should go into them, but this does not happen, and this would lead to the curving down of the radial distribution function (*rdf*) at the final few r values, $3^*\sigma_{max}/\Delta r$ bin will be discarded from the end of the convoluted histogram, and not included in the further calculations.

Both the original and the convoluted histograms are stored and saved to the **.hgm* file, as the original histogram is needed for the histogram update, and the radial distribution function is calculated from the convoluted histogram.

T. Increasing the contribution of the large r, Q, or g points

As the diffraction data and the pair correlation function both experiences a damping effect (the intensity of the oscillatory $g(r)$, $S(Q)$, $F(Q)$ and $F(g)$ decreases with the increasing r, Q, or g), sometimes it can be desirable to increase the contribution of the large r, Q, or g data points by fitting $Q\cdot S(Q)$, $Q\cdot F(Q)$, $g\cdot F(g)$ or $r[g(r)-1]$. (In case of $g(r)$ data fitting 1 has to be subtracted to make it oscillate around zero.)

This can be achieved for the $g(r)$ data sets, if the code is compiled with _R_SWITCH_GR_MIN_1 compiler option. In this case $r^N[g(r)-1]$ is assumed at reading for every $g(r)$ data sets, and this is calculated and used in the fitting. N can be set at compilation time, for LINUX it can be passed to the *Makefile* at compilation (section 0), for Windows it has to be set in *altern.h* (R_SWITCH_POWER). The default is 2.

To fit $Q\cdot S(Q)$, $Q\cdot F(Q)$, $g\cdot F(g)$ the code has to be compiled with _MUL_SCAT_VECTOR compiler option. In this case for all the used neutron, X.ray and electron diffraction data sets $Q\cdot S(Q)$, $Q\cdot F(Q)$, $g\cdot F(g)$ is assumed at reading, and this is calculated and fitted.

U. Custom histogram bin sizes for the different data sets

In some rare cases it can be advisable to use different histogram bin sizes for the different data sets. From version 1.6.1 this is possible, if a custom bin size is specified for the given data set after the number of data points used in the fitting in the **.dat* file, see section II.C for details. For each data set, where custom bin size is not specified, the normal bin size given in line 5 of **.dat* is used. Also from version 1.6.1 in case of $g(r)$ data sets the spacing of the data set points in the experimental data file is analysed, and if it is equidistant, the bin size and bin shift (see line 72 in the **.dat* file) of the data set is used for this data set overriding the default and even if it is given, the custom bin size. In this case no smoothing is required during the *ppcf* calculation from the histogram. In case of free format data file the key word R-SPACING in the [EXP] section of a data set can be used to specify the custom bin size for this data set.

V. Using Rw instead of the normally calculated χ^2 to guide the simulation

It can be desirable (mostly in case of EXAFS fitting) to use the Rw factor (section J) instead of the normally calculated χ^2 (section 0) to guide the simulation. This can be achieved by specifying optionally 1 after the weight parameter of a data set in the old format **.dat*, or giving the USE-RFACTOR =1 in case of the free format **.dat* file for the given data set. In this case Rw/sigma is used for this data set and added to the total χ^2. During the simulation the normal χ^2/data point is not displayed on the screen for this data set, only the Rw displayed as in case of normal fitting.

W. Fitting the intensity instead of the structure factor in case of X-ray data sets

As obtaining the structure factor from the measured intensity contains uncertainties it can be desirable to fit *I(Q)* directly instead of the *F(Q)*. This possibility is included in the program from version 1.8. The relationship between the total structure factor, *F(Q)* and the *I(Q)* is given by E 14. According to this now the calculated data will contain the exponential Compton damping factor with α, and its value is not exactly known, so it can be fitted as well.

As it was mentioned in 0, the following combination of the renormalization parameters described by E 15 is implemented:

- a is fitted, b and α custom (linear regression)
- a and b is fitted, α =0 (linear regression)
- α is fitted, a and b custom (non-linear regression)
- a and α is fitted, b =0 (non-linear regression)
- a, b and α is fitted (non-linear regression)

where a is the usual multiplying factor for the measured *I(Q)*, b is a constant additive factor, and α is given above. Where α optimization is involved, non-linear regression has to be used. Unfortunately non-linear regression can only be done by iterative methods, RMC uses the Levenberg-Marquardt method (implemented according to the Numerical Recipes, The art of scientific computing, 3rd edition https://e-maxx.ru/bookz/files/numerical_recipes.pdf, but some improvements were made ensuring the continuation of the simulation even in case of non-convergence) for optimizing the required parameters. The iterations can be time consuming, and might not converge, (although some modifications are implemented which would increase the possibility of convergence) so be careful using these options! The parameters (fudge lambda, factor to change lambda, maximum number of iteration, limit for convergence) governing the iterative process have the default values suggested by the manual, but all of them can be adjusted by the user as well, as described in section II.C.2II.C.2.

I(Q) fitting can only be applied, if the new, free format input file is used. To indicate for a given X-ray data set that *I(Q)* should be fitted instead of the structure factor, the I(Q)_A_B_ALPHA keyword should be included into the [EXP] section describing the data set. This key requires a mandatory 1 (fit *I(Q)*) or 0 (normal fit), but in the latter case it can be omitted as well.

Another unusual feature of the *I(Q)* fitting is that at certain cases non-trivial custom parameters can be given at the I(Q)_A_B_ALPHA keyword after the mandatory 0/1. Even if only b and α should be given, some value has to be supplied for a as well!

I(Q)_A_B_ALPHA only indicates that *I(Q)* fitting should be used for the given data set, but does not decide about which type of fitting. This can be regulated by the state of the data set's RENORM =0/1 for a parameter fitting, POLY-BACK-FLAGS = 0/1 for b parameter fitting and the new COMPTON-SUPR-COEFF=0/1 for α parameter fitting. If the state of the flags indicates a case which is not implemented, the program will terminate.

Considering the third option, where only α is fitted and a and b have custom values it has to be noted that these custom parameters, especially b should be chosen with care. Due to the nature of the $\exp(-\alpha Q_j^2)$ which only can have positive or close zero value correct solution only can be found if $a_i I_{i,j}^E + b_i - F_{i,j} - A_{i,j} \geq B_{i,j}\exp(-\alpha Q_j^2)$. Although in other cases convergence with some α parameter might be found, the model is statistically incorrect!

Some other keywords were implemented to make it possible to change the default values of the parameters governing the non-linear regression. These are universal for the whole program (for all the $I(Q)$ fitting data sets) and therefore can be given in the [GENERAL] section of the free format **.dat* file.
These are the following:
EPSILON-NONLIN: finish iteration, if abs(χ^2-χ^2_{old})< EPSILON-NONLIN
FACTOR-NONLIN: multiplication factor to increase-decrease lambda
LAMBDA-NONLIN: fudge factor
LOG-NONLIN-STEPS: write the iteration steps into the log file, only use for short runs if there is a problem with convergence, as writing to disc is slow and the log can be large.
MAX-NITER-NONLIN: stop iteration after this many steps even if convergence was not reached.
TERMINATE-NONLIN: Exit program, if this many successive move was rejected because of no convergence.

If iteration ended with convergence, the normal acceptance method is applied for the χ^2. If iteration ended without convergence, then that move will be rejected, and if TERMINATE-NONLIN successive moves are rejected because of no convergence, the program terminates.
The detailed description of the keywords can be found in section II.C.2 as well.

W.1. Using scalable background correction in case of *I(Q)* fitting X-ray sets

As the background correction is somewhat ambiguous, there is a possibility from version 1.9 to use scalable background correction of the experimental intensity for the *i*th data set *k*th data point in the from:

$$I(Q_k)_i^{\text{E}corr} = I(Q_k)_i^E - \mu_i^{act} IB(Q_k)_i \qquad \text{E 33}$$

where $I(Q_k)_i^{\text{E}corr}$ is the background-corrected experimental intensity and $I(Q_k)_i^{\text{E}}$ is the original experimental intensity, $IB(Q_k)_i$ is the Q-dependent uncertain background correction (supplied in the experimental data file), and μ_i^{act} is the scaling factor. The scaling factor is chosen randomly in the $\mu_i - d\mu_{max,i} \to \mu_i + d\mu_{max,i}$ range, μ_i and $d\mu_{max,i}$ are parameters specified by the user. $\mu_i - d\mu_{max,i}$ and $\mu_i + d\mu_{max,i}$ has to be between 0 and 1. This option is only available through the free format data file. If you want to use the scalable background correction, then the I(Q)BACKG_MU_DMU-MAX key word should be included in the [EXP] section of the data set in the free format **.dat* file with three parameters: first is an integer switch 0 (no background correction) 1 (use background correction) switching whether this option should be used, then if it is used first the μ_i then the $d\mu_{max,i}$ value could be specified, if not, default values will be used. If there is/are background correction-using *I(Q)* data sets, then for each in every Nstep$_\mu$ steps a new μ_i^{act} value is generated, and in this step no atomic move is performed similarly to the E0 shift generating steps (I.F.2) and the AXS (I.W.4). This way the χ^2 difference is only coming from the change in the background correction (and/or E0 shift and AXS). If the move is accepted, then the new μ_i^{act} value will be used in the following steps, if not then the program will revert to the old μ_i^{act} value. The Nstep$_\mu$ can be regulated by the I(Q)BACKG-STEP key world in the [GENERAL] section and therefore is the same for each data set using this option, if not given default value is used.

W.2. Some notes about the atomic scattering factor

The atomic scattering factor is composed of the following parts:

$$f(Q,E) = f(Q) + f'(E) + if''(E) \qquad \textbf{E 34}$$

When normal X-ray scattering data is simulated, then the coefficients are calculated from the first, Q-dependent *f(Q)* part. Sometimes, mainly in a multicomponent system with heavier atoms it cannot be avoided that the energy of the beam in a normal X-ray scattering experiment is close to one of the many absorption edges of a component. In this case it is advisable to use *f(Q)+f'(E)* for this component. This can be done applying the FPRIME-INDEX_FPRIME key word for this data set, see I.W.3.

If the energy of the beam is a bit lower than the absorption edge of a given atom type, anomalous X-ray scattering occur. In this case the energy-dependent part of the atomic scattering factor cannot be neglected. The absorption can be interpreted as a resonance phenomenon, and can be described by the imaginary component of the E-dependent part, $f''(E)$. At a bit lower energy than the absorption edge the absorption is somewhat smaller than at energies a bit above the edge. A bit below the absorption edge the real part of the scattering factor differs somewhat from the value further down form the edge, this contrast can be exploited during an anomalous X-ray diffraction experiment, see I.W.4. The change in the imaginary part is considered to be negligible, so it is omitted.

W.3. Using the fixed *f'(E)* for *X-ray data sets*

As $f'(E)$ is experimental setup and atom-type dependent, if $f'(E)_i$ should be used, it can be given in the [EXP] section describing the given data set. The FPRIME-INDEX_FPRIME key word followed first by the index of the atom type (starting with 1) and then the $f'(E)$ value can be used to give the necessary parameters. The value given here will be added to the $f(Q)$ values of the given atom type of this data set, and the coefficients will be calculated accordingly. There can be as many FPRIME-INDEX_FPRIME key words for a data set as atom types. This keyword can be used not only for $I(Q)$, but normal, $F(Q)$-fitting data sets as well, in the latter case only if the coefficients are calculated by the program!

W.4. Anomalous X-ray Scattering (AXS) in RMC

Ideally two experiments should be conducted with two different beam energy both below the absorption edge of the chosen atom type, one is close to the edge, the other is further away, and the difference should give the information. Unfortunately due to experimental errors simulating the difference curve was not feasible. Therefore an AXS simulation needs two X-ray $I(Q)$-fitting data sets measured on the same instrument one a bit below another further down the edge of the given element. The program obviously cannot decide, whether this condition is satisfied, only the number of data sets are checked. There can be an AXS set for each component of a multi-component system, so N_{AXS} AXS and N_{AXS} normal X-ray data sets are needed by the program. As usually those instruments which are capable of AXS measurement might not have a very large Q-range and there are other difficulties as well, it can be advisable to use another, wide Q-range X-ray set too, but this is not a requirement.

The measurement of the $f'(E)$ is problematic, so the initial value can be fitted as well by the program. This means, that in every $Nstep_{AXS}$ the value of $f'(E)$ is somewhat modified, and if the fit becomes better, then the new value is kept, otherwise discarded. In these steps no atomic move is performed as in case of E0 shift for EXAFS and scalable background correction for $I(Q)$ fitting!

To indicate, that the given X-ray set is an AXS set, the AXS-INDEX_FPRIME_FRACTION key word followed by the type of the near-edge component (starting with 1), the $f'(E)$ and the fraction $frac_p$ should be given. The new $f'(E)_{act}$ value is generated in the $f'(E)*(1-frac_p) \rightarrow f'(E)*(1+frac_p)$ interval. The default value of $Nstep_{AXS}$ can be modified by the AXS-SHIFTSTEP key word in the [GENERAL] section. It is possible to use fixed $f'(E)$ for the other atom types in the AXS set.

X. Scalable sigma parameters, leading series

If more, than one data series and constraints are used, it can be a lengthy and tiresome job to set the sigma (or other words weight) parameters properly, to ensure the proper convergence. It has to be kept in mind that the largest χ^2 data set will guide the system, and other data series or constraints with small sigma will not have virtually any effect on the outcome of the simulation. Therefore it is important to set the sigma values adequately to make sure that the χ^2 of all the series and constraints should make its contribution. For this point this could only be done by trial and error, restarting the simulation, as the deviation of the data sets and constraints and therefore their χ^2 cannot be known beforehand.

To make this easier, there is a possibility to choose a leading series, and to set the χ^2 values of the other series and constraints to the percentage of the χ^2 of the leading series. If we want to use this scaling feature, then for the sigma values for the data series and constraints to be scaled a negative fraction, which absolute value specifies the desired ratio of the given series per the leading series initial χ^2 should be given. The negative value only indicates that not normal sigma, but scaling should be used.

The index of the leading series is the first series of the simulation by default (the order of the data sets is the same as they appear in the **.dat* file, ($g(r)$, $S(Q)$, $F(Q)$, $F(g)$, $E(k)$ data, cosine distribution of bond angles, coordination number constraints including the sub-constraints, average coordination constraints, local invariance (and bonds, angles and periodic, harmonic and RB dihedrals, if no non-bonded potential is calculated)). The indexing starts with 1.

If non-bonded parameters are present, then a second χ^2 is calculated including the contributions of the non-bonded and bonded potential-related contributions as it is discussed in chapter I.Q. In this case another leading series index for this potential related contributions can be given in the **.dat* file (see line 54 in Table 7). The indexing for the leading potential series starts with 1 for the (first partial of the) vdW potential, and they follow in the order vdW partials, Coulomb partials, bonds, angles and dihedrals. 1-4 non-bonded interaction partials, and interactions with zero sigma parameter (see later) cannot be given as leading series index!

The χ^2 of the leading potential series can be scaled to the χ^2 of the normal, not potential related leading series, if the leading potential series' sigma value is given as a negative fraction. Those other potential related contributions, where the sigma is a negative fraction will be scaled to the leading potential series' χ^2.

It can be desirable that the ratio of all the bonded potential interactions a.) of a molecule type or b.) several molecule types should not change compared to each other. This can either be done by using the same fixed sigma for all the bonded interaction types in the **.top* or **.itp* file of the molecule in case of a.), or for the desired molecule types' interaction types in their respective section in **.top* or **.itp* file. But in this case no scaling can be applied.

The more convenient way is applying zero sigma values the following way: If not all molecule types should have the same sigma, then those which should not, have to come the last in the topology. For all those molecule types, which should have the same sigma one bond type of one of these molecule types has to be selected, and it has to have a non-zero (a fixed positive or scalable negative) sigma in the **.top* or **.itp* file. If this sigma is negative, than the χ^2 of this bond type will be scaled to the leading (potential) series' χ^2 giving a sigma for this bond type. For all the other bond types and angles and dihedrals of these molecule types zero sigma has to be given in the **.top* or **.itp* file, indicating that the sigma of the first bond type with non-zero sigma has to be used for them. (This is the reason, if there are other molecule types in the system, which we want to scale separately, they have to come after all the molecules types we want to scale together!) Even in case of fixed sigma it is better to use this 'one non-zero bond sigma all the other sigma is zero for the non-bonded interaction' formalism, as it is a lot easier to change all the bonded interactions' sigma in this case, if it is necessary. If there is a system with more, than one molecule type, and we want some of them to have all their bonded interactions scaled together, but separately from other molecule types' bonded interactions, them for all of them set one of their bond interaction with nonzero sigma, and set the other bonded interactions of these molecule types to zero. The order, in which the program determines in case of zero sigma, which should be the lead sigma, is first to look for a non-zero sigma among the bonds of the same molecule type's bonds, and only if none found, then trying to find the first bond interaction with non-zero sigma. It is possible, although it might not make much sense, to use more non-zero sigma values for a molecule beside the zero sigma values. Always the first non-zero bond sigma will be used for setting the zero values, the other non-zero values will be used normally (meaning that their sigma will not be set to the selected bond type's sigma, but their own sigma value will be used for them). After setting the sigma values, start the simulation and check on the screen output or in the **.hst* file, whether the desired effect was achieved.

Similarly we might want to ensure that the ratio of the non-bonded interactions potential does not change compared to each other. This can be done similarly to the method described above: The NB weight mode has to be set 0 or 3, (ensuring that all the vdW or Coulomb partials will have the same sigma and will be displayed together, one vdW and one Coulomb interaction). Then the desired (fixed or scalable) sigma has to be given for the vdW interaction, and 0 for the weight of the Coulomb interaction, indicating that the vdW weight should be used for this too.

If the ratio of all the potential interactions shouldn't change compared to each other, (the total potential energy of the system should be used), than the simplest way to do this (only if non-bonded potential is used), is to specify 2 for the NB weight mode. In this case the screen and the history display will be collapsed (only one item for each interaction type, similarly to NB weight mode 0). Only one sigma parameter has to be given for the vdW interaction, this can be positive, or negative. No other sigma will be read (though the line for the Coulomb weight has to be present). This sigma (after it was calculated, if it was scalable, negative) will be used for all the potential-related interactions.

For example let's consider the following simulations with the data sets, constraint and potential components given in Table 3. The leading series index will be 2, meaning the X-ray data series, and its sigma has to have an explicit value. The initial χ^2 of the neutron data series will be set to 10 % of the initial χ^2 of the X-ray series, the second cosine distribution of bond angles constraint's initial χ^2 to 5 % and the local invariance's to 1 %. The leading potential series (with index 3) will be the 3rd vdW partial in this case, and in itself will be scaled to the leading series having 20 % initial χ^2 compared to this. The other potential-based series will be scaled to this 3rd vdw partials χ^2, except the angle, and the first RB dihedral, which have fixed sigma. The sigma values used during the run for series to be scaled are calculated to satisfy the ratio between the initial χ^2 of the series to initial χ^2 of the leading series and it is constant during the run.

Table 3: Data sets, constraint and potential components for a simulation example demonstrating the usage of χ^2 scaling. The lines with yellow shading are the normal data series and constraints contributing to the normal χ^2, the blue shaded potentials contribute to the χ_P^2. The leading series and leading potential series is printed in bold.

serial index	**series/constraint**	**sigma**
1	neutron	-0.1
2	**X-ray**	**3e-5**
3	cosine distr. of bond angles	2e-6
4	cosine distr. of bond angles	-0.05
5	local invariance	-0.01
1	vdW partial 1	-0.8
2	vdW partial 2	-1.0
3	**vdW partial 3**	**-0.2**
4	Coulomb partial 1	-0.5
5	Coulomb partial 2	-0.4
6	Coulomb partial 3	-0.4
7	bond type 1	-0.2
8	bond type 2	-0.1
9	angle	1e-3
10	RB dihedral 1	2e-7
11	RB dihedral 2	-0.2
12	RB dihedral 3	-0.3

Y. Handling of vacancies

The possibility of handling vacancies was included from version 2.3. The vacancy is an atom type in the **.cfg* file, which will not contribute to scattering. Before version 2.3, for each data set the appropriate coefficients should have been set to zero. This was somewhat tedious for X-ray data and not possible for the Compton scattering contribution, if it was used. So the 'dummy' chemical symbol was included in the X-ray, neutron and Compton calculation tables, and if dummy is given (in this format!) for a type after CHEMICAL-SYMBOLS in the [GENERAL] section, and it is chosen, that the coefficients are calculated by the program, then the appropriate coefficients will be set to zero.

dummy atoms can have size to contribute to cut-off, and can be moved as well.

They should not be confused with virtual sites (I.Q.1.2(4)) occurring for potential runs!

Z. Exact continuation of the run

It might be desirable to be able to continue a simulation exactly. If no potentials are present, then this could be done earlier, since the binary **.bcf* file was present, only the previous history (**.hst*) file was overwritten, and the number of generated, tried and accepted moves reset at each new start.

In case of the potential, however as it is represented by double type variable, exact continuation of the run is not possible without the binary **.pot* file, as the potential at the end of a run can never be the same as the one recalculated at the beginning of the next run based on the **.bcf* file due to the rounding errors and the different order the contributions to the potential components are calculated.

An additional problem is, if any of the sigma values were scaled in the original run, because without saving and reloading the originally calculated sigma values new sigma values would be calculated when the run is restarted, based on the χ^2 values at the end of the first run, and the proportion of the different χ^2 components would change.

Therefore a new command line option was created (**cont**), which would start the exact continuation of the run. For this the **.bcf* , the binary **.state* file and in case of potential the binary **.pot* file is necessary. The syntactic of the command line will be discussed later (III.D).

II. The RMC_POT program

A. The structure of the program

The program is written in C++ and consists of several header and source files listed below.

Table 4: Header files of the RMC_POT program

Aenet.h	Header file for the AENET, holding the declaration of the Aenet class for ANN potential calculation. Contains ‘C’ interface for fortran routines. Only add to the project, if AENET should be used and there is suitable environment for compiling it.
altern.h	Header file including the necessary built-in headers, define statements for Windows based compiler regulating the building of the code, declaration of the functions differing according to the options chosen.
classes1.h	Declaration of basic classes: ExptsData, SimpleCfg, CoordNumbConst, AvCoordConst, CosDistrConst , RunParams, FNC_POT, HistoSet, and if _ADVANCED_GEOM_CONT compiler option is used, for BondValenceSumConst, CommonNeighConst and SecondNeighConst and the classes’ destructors.
classes2.h	Declaration of the classes involved in the calculation: DataMat, PPCFSet, CalcPart, CalcData, ChiSquared, History and the classes’ destructors
data.h	Declarations for the numerical tables.
global.h	Global file name, file extension and potential related declarations.
Move.h	Declaration of the Move class responsible for making the movement and the class’ destructor.
Neighbourlist.h	Declaration of the NeighbourList class related to the gridding of the simulation box, and in case of some constraints handling the neighbour list. It contains the class’ destructor as well.
Threads.h	Declaration of the Threads class handling the multi-threading and the class’ destructor.
Topology.h	Declaration of the Topology class handling the molecular topology and the class’ destructor.
units.h	Declaration of the used constants.
utilities.h	Declaration of the auxiliary functions and definition of the function templates.

Table 5: Source files of RMC_POT

Aenet.cpp	Definition of the Aenet class for ANN potential calculation. Contains calls to ‘C’ interface functions for fortran routinesrunning only under Linux. Only add to the project, if AENET should be used and there is suitable environment for compiling it.
altern.cpp	Definition of the function differing according to the chosen options
AvCoordConst.cpp	Definition of the AvCoordConst class describing the average coordination number constraint.
BondValenceSumConst.cpp	Definition of the BondValenceSumConst class holding and calculating the bond valence sum constraint, only compiled, if _ADVANCED_GEOM_CONT compiler option is on.
CalcData.cpp	Definition of the CalcData class holding and calculating the calculated data comparable with the experimental.
CalcPart.cpp	Definition of the CalcPart class holding and calculating the partial *g(r)*, *S(Q*, *F(Q)* and *E(k))*.
ChiSquared.cpp	Definition of the ChiSquared class holding and calculating difference between the calculated and the experimental data.
CommonNeighConst.cpp	Definition of the CommonNeighConst class holding and calculating the common neighbour constraint, only compiled, if _ADVANCED_GEOM_CONT compiler option is on.
CoordNumbConst.cpp	Definition of the CoordNumbConst class describing the coordination number constraint.
CosDistrConst.cpp	Definition of the CosDistrConst class describing the cosine distribution of bond angles constraint.
data.cpp	Contains the numerical tables used by the program
DataMat.cpp	Definition of the DataMat class containing the conversion table for the normalization of the *g(r)*, and the Fourier transformation matrices.
ExptsData.cpp	Definition of the ExptsData class containing the experimental data.
FNC_POT.cpp	Definition of the FNC_POT class handling the normal FNC and the potential.
globals.cpp	Contains the tables with the existing tags and keywords for the free format data input
History.cpp	definition of the History class gathering and outputting information about the run
HistoSet.cpp	Definition of the HistoSet class containing and calculating the histogram.
makemove.cpp	Containing the makemove function creating the movement of the atoms.
makemovecus.cpp	Containing the custom makemove function creating the movement of the molecules.
Move.cpp	Definition of the Move class, regulating the movements of the atoms, functions connected with the ‘tooclose’ atoms, other functions of the Move class.
NeighbourList.cpp	Definition of the NeighbourList class containing the gridding of the simulation box, and in case of cosine

	distribution of bond angles constraint handling the neighbour list.
PPCFSet.cpp	Definition of the PPCFSet class, handling the ppcf calculation and storage.
RMC_POT.cpp	The main program for RMC_POT regulating the run, creating the instances of the classes, containing the simulation loop.
RunParams.cpp	Definition of the RunParams class describing the parameters of the simulation, calculating some necessary parameters.
SecondNeighConst.cpp	Definition of the SecondNeighConst class holding and calculating the second neighbour constraint, only compiled, if _ADVANCED_GEOM_CONT compiler option is on.
SimpleCfg.cpp	Definition of the SimpleCfg class containing the coordinates of the atoms the charge group centres.
Threads.cpp	Definition of the Threads class containing the variables and function necessary for multi-threading, but a basic Thread class without multi-threading-specific variables is used even in case of consecutive compilation.
Topology.cpp	Definition of the Topology class reading and handling the molecular topology.
utilities.cpp	Definition of the auxiliary functions.

B. Files used by RMC_POT

These are the files used by the simulations performed by RMC_POT program for input and/or output.

- The input files written in bold are mandatory, but it is enough, if either the text or the binary coordinates files is given, text type has precedence over binary, if both is given and they are not compatible.
- If neither INPUT nor OUTPUT specified, then the file is used for both of them.
- The software can be started without giving any *experimental* data, in this case a hard sphere simulation is performed, and constraints can be present.
- The files given in brackets are produced only at the beginning of a whole simulation process.
- * means the general file name of the run, and shared by all the file containing the *.

Table 6: The files used by RMC_POT

***.dat**	INPUT: the data file containing the parameters of the run, can be fixed or free format
***.cfg**	The text type configuration file containing the coordinates.
***.bcf**	The binary type configuration file containing the coordinates.
anyname	INPUT: The files containing the experimental data, tabulated potential, custom cosine distribution, custom surface table, name can be free choice.
**.cus*	INPUT: The parameters for molecular move.
**.fnc*	INPUT: The Fixed Neighbour Constraint file.
**.top (*.itp)*	INPUT: The topology and include topology files, to describe the molecular topology.
**BOND.fnc,* **ANGLE.fnc,* **DIHEDRAl.fnc*	OUTPUT: Only if fnc=4: The RMC index lists of the bonds, angles and dihedrals.
**.hgm (*start.hgm)*	Partial histograms containing counts from unique atom pairs.
**lhgm (*start.lhgm)*	Only if compiled with _LOCAL_INV switched on: partial local

	histograms.
*grid (*start.grid)	Gridding of the simulation box.
*.cnc (*start.cnc)	The coordination number constraint.
*.acn (*start.acn)	The average coordination number constraint.
*.conc (*start.conc)	Common neighbour constraint (only in case of _ADVANCED_GEOM_CONST compilation)
*.snc (*start.snc)	Second neighbour constraint (only in case of _ADVANCED_GEOM_CONST compilation)
*bvs (*start.bvs)	OUTPUT: Bond valence sum constraint text file (only in case of _ADVANCED_GEOM_CONST compilation)
*bbv	Bond valence sum constraint binary file for the exact continuation (only in case of _ADVANCED_GEOM_CONST compilation)
*.tca (*start.tca)	The indices of the 'tooclose' atoms in case of the moveout option.
*.bin	The central bin index for each atom in case of _NO_PERIODIC compilation.
*.state	The binary state file needed for exact continuation of the run (the program started with the **cont** command line option).
*.pot	Only if non-bonded and/or bonded potential used: the binary potential file.
*.cos (*start.cos)	OUTPUT: The result of the cosine distribution of bond angles constraints.
*.calcdat	OUTPUT: The initial calculated $g(r)$, $S(Q)$, $F(Q, E(k)$ data.
*.fit (*start.fit)	OUTPUT: The total calculated and renormalized experimental $g(r)$, $S(Q)$, $F(Q, E(k)$ data, (_*start.fit_ only if compiled with _TEST_MODE compiler option).
*.expt	OUTPUT: The experimental data is saved to it for checking.
*.ppcf	OUTPUT: The partial radial distribution function $g(r)$ with the same spacing as the histogram.
*.pgr	OUTPUT: The calculated $g(r)$ partials on the same r points, as the 'experimental' $g(r)$.
*.psq	OUTPUT: The calculated $S(Q)$ partials for the neutron data sets.
*.pfq	OUTPUT: The calculated $F(Q)$ partials for the X-ray data sets.
*.pfg	OUTPUT: The calculated $F(g)$ partials for the electron diffraction data sets.
*.pek	OUTPUT: The calculated $E(k)$ partials for the EXAFS data sets.
*.hst	OUTPUT: The history file containing information about the run.
*.chi	OUTPUT: The initial χ^2 for each data set.
*.datmat	OUTPUT: The conversion tables of the DataMat object (only if compiled with _TEST_MODE compiler option).
*.out	OUTPUT: The total and partial calculated and total renormalized experimental $g(r)$, $S(Q)$, $F(Q, E(k)$ data in RMCA format, only if OLD_OUT=1 is used at the end of the _*.dat_ file, or keyword CREATE-OUT is included in the free format _*.dat_ file..
*.nei	OUTPUT, only in case the code was compiled with _NEI switched on: containing the neighbour list, and in case compiled with _NEIE the squared distances as well
*.excl	OUTPUT, only if non-bonded potential is calculated: the indices of the excluded atoms for each atom.
*avm	OUTPUT, only in case of _AV_MOVE, contains the distance for the starting position for each atom.
*.tabp	OUTPUT: for the tabulated potential
_*.en_	OUTPUT: for the ANN potential, only if compiled with _AENET
*.free	OUTPUT: free format data file containing all the parameters of the run
*.freer	OUTPUT: reduced free format data file containing only the non-default parameters of the run before sigma scaling

**.freers*	OUTPUT: reduced free format data file containing only the non-default parameters of the run after sigma scaling, only generated, if scalable sigma parameters were used
**.log*	OUTPUT: in case of LOG-NONLIN-STEPS key word (only available through the free format **.dat* writing the non-linear regression steps
filename#ORI	OUTPUT: from version 2.3:if any input text file was read, which did not contain the right line ending for the operating system, then the original text file renamed to this name, and the content of the file will be converted to the right format with the original filename.

C. The structure of the *.dat file

The main parameters of the simulation can be found in this file. Originally the file has a fixed format in the sense that the order of the lines and the order of the parameters in a line is fixed. Introducing new features into the code and therefore putting new parameters into the data file was difficult to keep up downward compatibility and keep the parameters logically ordered as well.

László Temleintner introduced the concept of free format data input into version 1.4, but it was not completed. The fully operational version of free format input is available since version 1.6.2, and from this point new features will be available **only** using free format input, though the fixed format will be kept as a legacy using only the features already available till this point.

As the file name is the same *(*.dat*), the program differentiate between the two data input by looking for #XXX (where XXX are the version of the **.dat* file. The first free format **.dat* file has version number #002 in the beginning of the first line. The version #003 free format **.dat* file was introduced in RMC version 2.2 and became the default for the program, but the others can be used as well. If #XXX is found, then free format data input is assumed. In case of the fixed format the first line contains the title of the run, so DO NOT USE #002 or #003 as the beginning of the title!

From RMC version 2.0 the possibility of the X-ray coefficient calculation was introduced with default behaviour calculating the coefficients. This caused downward incompatibility. Therefore in RMC version 2.2 the #003 format was introduced to tell the program that from this **.dat* file version the calculation of the X-ray coefficients will be the default both for normal and *I(Q)* fitting. Default means, that if nothing is given regarding the choice between calculation versus reading, then calculation will be assumed. But for this the chemical symbols in RMC order has to be given with the newly created CHEMICAL-SYMBOLS key word. Therefore the previous (version #002) free format **.dat* files could not have been used without modification. As so far downward compatibility for the previous **.dat* file format was maintained this necessitated the introduction of the version #003 **.dat* format. Visually the only difference between version #002 and #003 is the version number, but while in case #002 the default handling of X-ray coefficients will be reading as before, in case of #003 the default will be calculation promoting this choice as it is more precise and saves the user of the trouble of calculating the X-ray coefficients outside the program. In case of the fixed format **.dat* file reading the coefficients is the default and only option. The usage of the READ-COEFFS key word in both free format versions can regulate the choice of the X-ray coefficients handling, if not the default behaviour is preferred (see more about the key word and usage in section II.C.2 and II.E.3).

In case of the free format input a lot of parameters (a lot more, than is case of the fixed format input) have default values, so their presence is not needed to successfully run the program. The program outputs all the parameters of the simulation in free format in the **.free* file, and only the non-default values into the reduced **.freer* and **.freers* files. The last one is only created, if scalable sigma parameters are used, as in this case the **.freer* contains the scalable sigma and the **.freers* the scaled, actual sigma values. Any of these files –renamed- can be used to start the simulation. They contain comments as well, the description of the keywords and the alternatives if there are any.

C.1. Fixed format data input

In the fixed format **.dat* file the majority of the lines has to be always present in the file, but if certain options are used (either regulated by compiler option or by a switch in the **.dat* file), additional lines at certain place of the file has to be included. These lines will be highlighted with colours in the example.

All lines will begin with data, and after the data remarks can follow. The remarks does not have to begin with a exclamation mark, it just helps to mark its beginning. The data can be separated by any number of spaces and/or tabs. If multiple data has to be given in one line, they have to be in the same line, and not split into several lines, although this was possible with earlier versions. No empty lines can be present, except the end of the file. The file does not refer to any actual simulation; it features all the possible line types.

Table 7: The structure of the fixed format **.dat* file. The data given underlined and italic means that they are optional, and do not have to be specified, the values given here are the default values used in case nothing is given, except in some cases, where they reflect the actual layout of this example file, here they are bold as well. The default vales are 1 in these cases! Lines underlined only included at certain cases, but then the values are mandatory! The first column is just a serial index to reference the lines, it is not part of the file! Only the lines without shading mandatory in each case, the lines with coloured shading are only necessary, if the data set, constraint or option they are referring to are used, or if the code was compiled with some compiler options specified in the description of the line.

1	test_run	
2	0.076605545	! number density in (1/A3)
3	1.3 2.2 1.5	! cut-offs for each RMC partials in A
4	0.1 0.2	! maximum moves for each types in A
5	0.1 *0.1*	! r spacing (A), *r spacing local inv. (A)*
6	.true.	! whether to use moveout option (0 or .false.: false, 1 or .true. : true)
7	10 *1*	! no of configurations to collect, *frequency*
8	1000	! step for printing
9	120 30	! time limit, step for saving in minute
10	1 1 1 1 1 *1*	! no. of *g(r)*, neutron, X-ray, electron diffraction and EXAFS data series, *leading series index*
11	mydata.gr	! *g(r)* file name
12	1 305 *def_dr*	! range of r points (start with 1), *custom rspacing*
13	0.00	! constant to subtract
14	0.2 0.5 0.3	! partial coefficients in RMC order
15	1e-5 *0*	! standard deviation*, whether to use Rw instead of* χ^2 *for this set in the fitting (1 if wanted)*
16	.false. *0*	! whether to vary amplitudes (0 or .false.: false, 1 or .true. : true) *use_cubic switch (0 or .false.: false, 1 or .true. : true)*
16b	0	! whether to vary constant (0 or .false.: false, 1 or .true. : true), only read if use_cubic is given
16c	0	! whether to vary linear (0 or .false.: false, 1 or .true. : true), only read if use_cubic is given
16d	0	! whether to vary quadratic (0 or .false.: false, 1 or .true. : true), only read if use_cubic is given
16e	0	! whether to vary cubic (0 or .false.: false, 1 or .true. : true), only read if use_cubic is given
17	neutron.sq	! file name for the neutron data
18	10 101 *def_dr*	! range of Q points (start with 1), *custom rspacing*
19	0.000	! constant to subtract
20	0.2 0.1 0.7	! partial coefficients in RMC order
21	3e-12 *0*	! standard deviation, *whether to use Rw instead of* χ^2 *for this set in*

		the fitting (1 if wanted)
22	0 *0*	! whether to vary amplitudes (0 or .false.: false, 1 or .true. : true) *use_cubic switch (0 or .false.: false, 1 or .true. : true)*
23	1	! whether to vary constant (0 or .false.: false, 1 or .true. : true)
24	0	! whether to vary linear (0 or .false.: false, 1 or .true. : true)
25	0	! whether to vary quadratic (0 or .false.: false, 1 or .true. : true)
25b	0	! whether to vary cubic (0 or .false.: false, 1 or .true. : true), only read if use_cubic is given
26	xray.fq	! name of the X-ray file
27	1 139 *def_dr*	! range of Q points (start with 1) , *custom rspacing*
28	0.000	! constant to subtract
29	3e-12 *0*	! standard deviation, *whether to use Rw instead of χ^2 for this set in the fitting (1 if wanted)*
30	.true. *0*	! whether to vary amplitudes (0 or .false.: false, 1 or .true. : true) *use_cubic switch (0 or .false.: false, 1 or .true. : true*)
31	.true.	! whether to vary constant (0 or .false.: false, 1 or .true. : true)
32	.true.	! whether to vary linear (0 or .false.: false, 1 or .true. : true)
33	.false.	! whether to vary quadratic (0 or .false.: false, 1 or .true. : true)
33b	.false.	! whether to vary cubic (0 or .false.: false, 1 or .true. : true), only read if use_cubic is given
34	*Ediff.fg*	! name of the electron diffraction file
35	2 96 *def_dr*	! range of *g* points (start with 1) , *custom rspacing*
36	0.000	! constant to subtract
37	5e-6 *0*	! standard deviation, *whether to use Rw instead of χ^2 for this set in the fitting (1 if wanted)*
38	.true. *0*	! whether to vary amplitudes (0 or .false.: false, 1 or .true. : true) *use_cubic switch (0 or .false.: false, 1 or .true. : true)*
39	0	! whether to vary constant (0 or .false.: false, 1 or .true. : true)
40	.true.	! whether to vary linear (0 or .false.: false, 1 or .true. : true)
41	.false.	! whether to vary quadratic (0 or .false.: false, 1 or .true. : true)
41b	0	! whether to vary cubic (0 or .false.: false, 1 or .true. : true), only read if use_cubic is given
42	21 31 *21 31 21 31*	! range of histogram bin points to use for EXAFS (r space), one first used-last used pair has to be given at least*. If it differs for the partials, then ntypes npartials pairs has to be given*
43	3	! chi(k) power
44	exafs.ek	! name of the k-E(k) file
45	1 444	! no. of k data points (starting with 1)
46	60 260 *def_dr*	! range to be used in fitting, *custom rspacing*
47	1	! type of absorbing particle
48	exafs_coeff.dat	! coefficient file for EXAFS
49	-0.0005 *0*	! weight parameter, *whether to use Rw instead of χ^2 for this set in the fitting (1 if wanted)*
50	1	! whether to vary amplitudes (0 or .false.: false, 1 or .true. : true)
51	0	! whether to vary constant (0 or .false.: false, 1 or .true. : true)
52	4 *0.05*	! number of cosine distr. of bond angles constraint, *dcos_theta (spacing in cos(theta) space)*
53	0 2 1 1 2.1 2.1 2.7 2.7 110.5 20 0.001	! calc method (0:step, 1:Gauss, 2:no angle, 3:file),central, neigh1, neigh2 type, dmin1, dmin2, dmax1, dmax2, 0-2:(angle, wcontrol) 3:filename, weight
54	1 2 1 1 2.1 2.1 2.7 2.7 110.5 0.17 0.001	! calc method (0:step, 1:Gauss, 2:no angle, 3:file),central, neigh1, neigh2 type, dmin1, dmin2, dmax1, dmax2, 0-2:(angle, wcontrol)

		3:filename, weight
55	2 3 1 2 2.3 3.0 3.0 3.8 41 0.2 0.0001	! calc method (0:step, 1:Gauss, 2:no angle, 3:file),central, neigh1, neigh2 type, dmin1, dmin2, dmax1, dmax2, 0-2:(angle, wcontrol) 3:filename, weight
56	3 1 2 2 2.1 2.1 2.7 2.7 SeAsSe.cos 0.0002	! calc method (0:step, 1:Gauss, 2:no angle, 3:file),central, neigh1, neigh2 type, dmin1, dmin2, dmax1, dmax2, 0-2:(angle, wcontrol) 3:filename, weight
57	2 ***1 2 2 1***	! no of coordination constraints, *number of neighbour type/constraint, number of sub-constraints/constraint*
58	2 1 2.1 2.7 2 3 0.4 0.6 0.00015 0.00025	! central type, neighbour type(s), rmin[first_neightype] … rmin[last_neightype], rmax[first_neightype] … rmax[last_neightype], desired coordination number[first_subconst]... desired coordination number[last_subconst], fraction[first_subconst]...fraction[last_subconst], sigma[first_subconst]...sigma[last_subconst]
59	3 1 2 2.4 3.0 2.95 3.5 2 1.0 0.00025	! central type, neighbour type(s), rmin[first_neightype] … rmin[last_neightype], rmax[first_neightype] … rmax[last_neightype], desired coordination number[first_subconst]... desired coordination number[last_subconst], fraction[first_subconst]...fraction[last_subconst], sigma[first_subconst]...sigma[last_subconst]
60	1	! number of average coordination constraints
61	1 1 34.0 40.0 12 0.004	! type of the central atom, type of neighbour, min dist, max dist, desired average coord numb, sigma
62	1 **2**	! number of common neighbour constraint, number of secondary types/constraint ONLY available, if the code was compiled with _ADVANCED_GEOM_CONST compiler option!
63	*3 4 1.8 2.0 2 3 0.9 1.52 1.5 2.1 1.1 1.72 1.7 2.3 2 0.8 0.01*	*! type of primary1, type of primary2, rmin_primary, rmax_primary, secondary neighbor type(s), rmin[primary1-secondary1]...rmin[primary1-secondaryN], rmin[primary2-secondary1]...rmin[primary2-secondaryN], rmax[primary1-secondary1]...rmax[primary1-secondaryN], rmax[primary2-secondary1]...rmax[primary2-secondaryN], number of desired common neighbours, fraction, sigma.* ONLY available, if the code was compiled with _ADVANCED_GEOM_CONST compiler option!
64	1 1 3 0.5 0.5 7 *12*	! whether to use a non-bonded potential, weight mode, combination rule, vdW fudge, Coulomb fudge, lead series index2, *LJ_power*
65	0.9 1.1	! vdW and Coulomb cut-off in A
66	4	! number of different GROMACS types
67	3.5 2.5 3.6 3.5	! First potential parameter depending on the combination rule
68	0.276144 0.12551 0.148532 0.276144	! Second potential parameter depending on the combination rule
69	1e-3 2e-3 3e-3 4e-3 5e-3 6e-3 7e-3 8e-3 9e-3 1e-4 1.1e-4 1.2e-4 1.3e-4 1.4e-4 1.5e-4 1.6e-4 1.7e-4 1.8e-4 1.9e-4 2.0e-4 2.1e-4	! vdW weights, same used for 1-4 interactions, if they are present, (maximum: number of RMC partials); χ_P^2 lower limit fraction in case of negative weight mode in line 64
70	1e-5 2e-5 3e-5 4e-5 5e-5 6e-5 7e-5 8e-5 9e-5 1e-6 1.1e-6 1.2e-6 1.3e-6 1.4e-6 1.5e-6 1.6e-6 1.7e-6 1.8e-6 1.9e-6 2.0e-6 2.1e-6	! Coulomb weights, same used for 1-4 interactions, if they are present, (maximum: number of RMC partials)

64b	10 *1*	! whether to use nonbonded-potential, *lead series index2*
65b	**15**	**! cutoff for the tabulated potential (Angstrom)**
66b	293	**! T (K)**
67b	**1**	**! number of tabulated potential files, then as many blocks of 68b - 70b as the number of tabulated potentials**
68b	**Potential_1_2.pot**	**! name of the tabulated potential file**
69b	**2**	**! which partial to use it for**
70b	**0.5**	**! weight parameter**
71	4 *-D_FF_OPLS -D_FLEX –DORI_ANGLE -DORI_BOND*	! FNC switch, *define options for the topology in case of fnc=4*
72	0	! initial bin shift
73	1.0	! xmax used in the run (Å)
74	1	! number of atoms moved in a single move
75	50	!size of the history buffer (0->no history record, 200 is good)
76	10	! number of saves between each history buffering
77	0	! indicator of custom move (0:no, 1:yes)
78	1	! whether to load the histogram (and coord. numbers, cos distr. local hist, if they are used), if the files are available (0 or .false.: false, 1 or .true. : true)
79	14	! maximum number of atoms in a gridcell
80	0.3 1	! fraction of swaps; *ntypes**(*ntypes*-1)/2 entry: 0 (not allowed) or 1 (allowed) for the possible mixed partial (in order 1-2, 1-3,... 2-3...)
81	2	! total number of threads to use
82	2 -0.3 –0.1 1 0.0 0.2 0.4	! number of consecutive local invariance intervals (N_{loc}), sigmas (N_{loc} entry), mode, loc_ratio (N_{loc} +1 entry). ONLY available, if the code was compiled with _LOCAL_INV compiler option!
83	25 *0*	! the radius of the spherical sample in Å for calculation of nanoparticles without periodic boundary conditions. Optionally: recentre flag (0 or nothing: not recentred, 1: recentred. ONLY available, if the code was compiled with _NO_PERIODIC compiler option!
84	0.05 0.07	! sigma values for the vibrational amplitude distribution for each atom type in A ONLY available, if the code was compiled with _VIBR_AMP compiler option!
85	*OLD_OUT=0*	*Whether to write old format output *.out (0, 1)*
86	*SUM_PPCF=0*	*Whether to sum the ppcf and write it to the end of the *.ppcf file (0, 1)*

Detailed description of some of the parameters, the parameters will be referenced by the serial number in the first column.

1. Title of the run, maximum 80 characters.

2. Number density for the system in Å^{-3}. If this does not correspond to the half box length and number of atoms in the **.cfg* file, than the simulation box will be rescaled according to this value in the **.dat* file, and this will be used during the simulation, and the rescaled half box length will be written in the **.cfg* file at saving. In case of using **non-periodic boundary conditions**, (the program has to be compiled using the _NO_PERIODIC compiler option) the number density given here refers to the total number of atoms divided by the volume of the spherical sample with radius R_0 (see section I.R and line 0.

3. Cut-off values for the hard sphere pair types in Angstrom, *ntypes*(ntypes+1)/2*. It can be given either in the same row in 1-1 1-2 1-3 2-2 2-3 3-3 order (for a 3 atom type system) or in an upper triangular matrix order split into rows like

1-1 1-2-1-3

2-2 2-3
3-3.
4. Maximum moves for each RMC type in Å.
5. Width for the histogram bins in Å, if compiled for local invariance calculation then the width for the local histogram bins can be given in Å, if not the same is used as for the normal histogram.
6. If there are particles closer to each other than the cut-off distance (see I.C), using the moveout option (1) means to move more frequently the 'tooclose' particles to increase their distance above the cut-off. (0) means not to use moveout option. There can be more, than one 'tooclose' particles among the n_{moved} moved particles, and it can be used even in the case of the molecular move!
7. The number of configuration to be collected after the time limit has been reached. A configuration will be saved at each frequency*savetime. They will be collected in the *_coll.cfg file.
8. Printing to the standard output in every number of steps specified here.
9. Time limit for the simulation in minute, . if positive or 0, step for saving in minute. If time limit is negative, it is interpreted as abs(number of steps to generate). Time limit is ignored, if compiled in _TEST_MODE, as in this case it runs to the number of generated steps specified by LAST_MOVE_DEF in *units.h*.
10. Number of *g(r), S(Q), X-ray* and *EXAFS* data, index of the leading series (see I.S above).
10-16e Only has to be included, if *g(r)* data series is present, and as many blocks of it after each other as the number of *g(r)* series is present.
11. The name of the *g(r)* data file, with maximum filename size regulated by FILE_NAME_SIZE located in *units.h*.
12. The first and last *r* data point to use (indexing start with 1. Optionally custom r spacing for this data set can be given here in A.
13. Constant to be subtracted from the experimental data after it was read, shifting the data along the *y*-axis.
14. Partial coefficients following each other in RMC order, as many as the number of RMC partials, their sum normalized to 1.
15. Sigma value. The deviation of the experimental and calculated data set is divided by its square; it is used for scaling the contribution of the different data series and constraint to each other. The smaller the value, the larger the contribution of the given data set will have to the total χ^2. After it optionally 1 can be given, if for this data set the Rw should be used instead of the normally used χ^2 to guide the simulation. The sigma also applied in this case, dividing Rw by sigma, and this is added to total χ^2_{tot}.
16. Whether to use renormalization of the data set by varying the amplitude (see 0. for details). Optional cubic switch: 1 has to be given, if constant, linear, quadratic and cubic renormalization is wanted, in this case include lines 16b-16e, if it is not given, then the next data series or constraint is read from next line.
16b-16e Can only be included, if 1 was given for cubic switch in line 12.
16b. Whether to use renormalization of the data set by changing the constant (see 0. for details).
16c. Whether to use renormalization of the data set by varying the linear coefficient (see 0. for details).
16d. Whether to use renormalization of the data set by changing the quadratic coefficient (see 0. for details).
16e. Whether to use renormalization of the data set by changing the cubic coefficient (see 0. for details).
17-25b. Only has to be included, if *S(Q)* data series is present, and as many blocks of it after each other as the number of *S(Q)* series present.
17. The name of the *S(Q)* data file, with maximum filename size regulated by FILE_NAME_SIZE located in *units.h*.
18. The first and last *Q* data point to use (indexing start with 1). Optionally custom r spacing for this data set can be given here in A.
19. Constant to be subtracted from the experimental data after it was read, shifting the data along the *y*-axis.
20. Partial coefficients following each other in RMC order, as many as the number of RMC partials, their sum normalised to 1.

21. Sigma value. The deviation of the experimental and calculated data set is divided by its square; it is used for scaling the contribution of the different data series and constraint to each other. The smaller the value, the larger the contribution of the given data set will have to the total χ^2. After it optionally 1 can be given, if for this data set the Rw should be used instead of the normally used χ^2 to guide the simulation. The sigma also applied in this case, dividing Rw by sigma, and this is added to total χ^2_{tot}.

22. Whether to use renormalization of the data set by varying the amplitude (see 0. for details). Optional: 1 has to be given, if cubic renormalization is wanted, in this case include line 25b, if it is not given, then the next data series or constraint is read from next line.

23. Whether to use renormalization of the data set by changing the constant (see 0. for details).

24. Whether to use renormalization of the data set by varying the linear coefficient (see 0. for details).

25. Whether to use renormalization of the data set by changing the quadratic coefficient (see 0. for details).

25b. It can only be included, if 1 was given for cubic switch in line 18. Whether to use renormalization of the data set by changing the cubic coefficient (see 0. for details).

26-33b Only has to be included, if *F(Q)* data series is present, and as many blocks of it after each other as the number of *F(Q)* series present.

26. The name of the *F(Q)* data file containing not only the *Q* and *F(Q)* data points, but the *Q*-dependent coefficients as well always normalized that their sum is 1 for each *Q* point, with maximum filename size regulated by FILE_NAME_SIZE located in *units.h*.

27. The first and last *Q* data point to use (indexing start with 1). Optionally custom r spacing for this data set can be given here in A.

28. Constant to be subtracted from the experimental data after it was read, shifting the data along the *y*-axis.

29. Sigma value. The deviation of the experimental and calculated data set is divided by its square; it is used for scaling the contribution of the different data series and constraint to each other. The smaller the value, the larger the contribution of the given data set will have to the total χ^2. After it optionally 1 can be given, if for this data set the Rw should be used instead of the normally used χ^2 to guide the simulation. The sigma also applied in this case, dividing Rw by sigma, and this is added to total χ^2_{tot}.

30. Whether to use renormalization of the data set by varying the amplitude (see 0. for details). Optional: 1 has to be given, if cubic renormalization is wanted, in this case include line 33b, if it is not given, then the next data series or constraint is read from next line.

31. Whether to use renormalization of the data set by changing the constant (see 0. for details).

32. Whether to use renormalization of the data set by varying the linear coefficient (see 0. for details).

33. Whether to use renormalization of the data set by changing the quadratic coefficient (see 0. for details).

33b. Can only be included, if 1 was given for cubic switch in line 27. Whether to use renormalization of the data set by changing the cubic coefficient (see 0. for details).

34-41b. Only has to be included, if *F(g)* data series is present, and as many blocks of it after each other as the number of *F(g)* series present.

34. The name of the *F(g)* data file containing not only the *g* and *F(g)* data points, but the *g*-dependent coefficients as well always normalised that their sum is 1 for each *g* point, with maximum filename size regulated by FILE_NAME_SIZE located in *units.h*.

35. The first and last *g* data point to use (indexing start with 1). Optionally custom r spacing for this data set can be given here in A.

36. Constant to be subtracted from the experimental data after it was read, shifting the data along the *y*-axis.

37. Sigma value. The deviation of the experimental and calculated data set is divided by its square; it is used for scaling the contribution of the different data series and constraint to each other. The smaller the value, the larger the contribution of the given data set will have to the total χ^2. After it optionally 1 can be given, if for this data set the Rw should be used instead of the normally used χ^2 to guide the simulation. The sigma also applied in this case, dividing Rw by sigma, and this is added to total χ^2_{tot}.

38. Whether to use renormalization of the data set by varying the amplitude (see 0. for details). Optional: 1 has to be given, if cubic renormalization is wanted, in this case include line 33b, if it is not given, then the next data series or constraint is read from next line.

39. Whether to use renormalization of the data set by changing the constant (see 0. for details).

40. Whether to use renormalization of the data set by varying the linear coefficient (see 0. for details).

41. Whether to use renormalization of the data set by changing the quadratic coefficient (see 0. for details).

41b. Can only be included, if 1 was given for cubic switch in line 27. Whether to use renormalization of the data set by changing the cubic coefficient (see 0. for details).

42-51 Only has to be included, if *E(k)* data series is present, and as many blocks of it after each other as the number of *E(k)* series present.

42. The range of histogram points to be used for the calculation starting with 1 (*r* space). At least one first used-last used pair has to be given, if different starting or end point should be used for the different partials, then it has to be given one pair after the other in the same line for all the partials, which means *nytpes* pair, as only the partials containing the edge particles exist.

43. The power parameter *n* in $E(k)=k^n\chi(k)$ (see I.F.1.)

44. The name of the *E(k)* data file, with maximum filename size regulated by FILE_NAME_SIZE located in *units.h*.

45. The number of *k* data points in the file.

46. The first and last *k* data point to use (indexing start with 1). Optionally custom r spacing for this data set can be given here in A.

47. The RMC type of the absorbing particle (starting with 1).

48. The name of the coefficient file containing the (*r*,*k*) dependent coefficients for the Fourier transformation. Data belonging to the same *r* values are in columns and belonging to the same *k* form rows. It has to be at least as many columns as the last used r value given in line 42 and as many rows as the last used *k* point given in line 46.

49. Sigma value. The deviation of the experimental and calculated data set is divided by its square; it is used for scaling the contribution of the different data series and constraint to each other. The smaller the value, the larger the contribution of the given data set will have to the total χ^2. After it optionally 1 can be given, if for this data set the Rw should be used instead of the normally used χ^2 to guide the simulation. The sigma also applied in this case, dividing Rw by sigma, and this is added to total χ^2_{tot}.

50. Whether to use renormalization of the data set by varying the amplitude (see 0. for details).

51. Whether to use renormalization of the data set by changing the constant (see 0. for details).

52. Number of cosine distribution of bond angle constraint, and if there is any the spacing in $cos(\theta)$ space, $dcos(\theta)$.

45-48. Only has to be included, if cosine distribution of bond angle constraint is present, and as many line as the number of constraint. Calculation method (0: step, 1: Gauss, 2: no angle, 3: experimental distribution from file), central type, neighbour1 type, neighbour2 type starting with 1, minimum and maximum distances $d_{min}1$, $d_{min}2$, $d_{max}1$, $d_{max}2$, for method 0-2: desired angle in degree, wcontrol parameter, for method 3: file name, for each method: sigma (see I.K.).

53. Example for step type cosine distribution constraint.

54. Example for Gauss type cosine distribution constraint.

55. Example for no angle type cosine distribution constraint.

56. Example cosine distribution constraint read from file.

57. Number of the coordination number constraints, number of neighbour type for each constraint, number of sub-constraint for each constraint (see I.L).

50-51. Only has to be included, if coordination number constraint is present, a line for each constraint. First comes the central type, neighbour type(s), r_{min}[first_neightype]…r_{min}[last_neightype], r_{max}[first_neightype] …r_{max}[last_neightype], desired coordination number[first_subconst]... desired coordination number[last_subconst], fraction[first_subconst]...fraction[last_subconst], sigma[first_subconst]...sigma[last_subconst]

58. Example for coordination number constraint with one neighbour type and two subconstraint.

59. Example for coordination number constraint with two neighbour type and one subconstraint.

60. Number of the average coordination constraints (see I.M).

61. Only has to be included, if there are average coordination constraints, a line for each constraint. Type of the central atom, type of neighbour, min distance, max distance, desired average coordination number and sigma

62. Number of common neighbour constraint, number of secondary types/constraint. ONLY available, if the code was compiled with _ADVANCED_GEOM_CONST compiler option!

63. Only has to be included if there are common neighbour constraints, a line for each constraint. Type of primary1, type of primary2, minimum and maximum distance between the primaries, types of the secondary1…N, minimum distance primary1-secondary1… minimum distance primary1-secondaryN, minimum distance primary2-secondary1… minimum distance primary2-secondaryN, maximum distance primary1-secondary1… maximum distance primary1-secondaryN, maximum distance primary2-secondary1… maximum distance primary2-secondaryN, target coordination number, desired fraction, sigma. ONLY available, if the code was compiled with _ADVANCED_GEOM_CONST compiler option!

64. Non-bonded potential switch: 0: no potential is used, 1 Lennard-Jones potential is used. Weight mode can be 0, in this case all the vdW partials will be summed and weighed with one sigma, the same goes for Coulomb partials, so only one sigma has to be given for vdW and one for the Coulomb interaction. If it is 1, then the vdW partials and Coulomb partials will be weighed separately, and each requires its own sigma parameter, so *npartials* sigma has to be given for both the vdW and for the Coulomb interaction. In case of 2 all the bonded and non-boned potential related interactions will be weighed with the same sigma, so only one sigma for the vdW interaction in line 69 has to be given, and line 70 (Coulomb weight) has to be present, but not read. No weight parameters are read from the **.top* or **.itp* file either. Weight mode 3 is the same as weight mode 0, it was introduced to be able to use negative values as well for all the options. In case of negative values for the weight mode (-1, -2 and -3), it is interpreted as its absolute value to regulate the weight mode, but the program will look for a percentage value after the last sigma in line 69 to set the lower limit of χ_P^2 (see details in I.Q).. The combination rule parameter is discussed earlier (I.Q.1.2(5)), can be 0-3. vdW fudge, Coulomb fudge (I.Q.1.2(3)), lead potential series index (0), and the optional LJ_power for the repulsion term, if it is not given, then 12 is assumed.

65-70 Only used in case of non-bonded LJ potential calculation.

65. vdW and Coulomb cut-off in reduced units, like x_{max}, the potential will be calculated only up to this reduced distance.

66. Number of different GROMACS type of atoms, (can differ from the number of RMC types (I.Q.1.2)).

67. Lennard-Jones first potential parameter will be read according to the combination rule (I.Q.1.2(5)) for all the GROMACS partials in case of combination rule=0, or for all the GROMACS types in case of combination rule>0. LJ sigma has to be given in Angstrom.

68. Lennard-Jones second potential parameter will be read according to the combination rule (I.Q.1.2(5)) for all the GROMACS partials in case of combination rule=0, or for all the GROMACS types in case of combination rule>0. LJ epsilon has to be given in kJ/mol.

69. wdW weight parameter, only one entry, if weight mode=0, 2, 3 in line 64, number of RMC partials entry, if weight mode=1. Can be positive or negative (this means scalable, see I.S for details).

70. Coulomb weight parameter, only one entry, if weight mode=0, 2, 3 in line 64, number of RMC partials entry, if weight mode=1. Can be positive, negative or only if weight mode=1, zero (see I.S for details).

64b-70b Only used, if 10 (tabulated potential) is given for the potential switch.

64b Non-bonded potential switch, 10 for tabulated potential. Optionally leading potential series index, if not given 1 is assumed.

65b Cut-off for the tabulated potential. If the largest distance for which tabulated potential is supplied is smaller than cut-off, for that partial the cut-off is set to r_{max}.

66b Temperature to calculate U/kT. It does not play any algorithmic role, only for display!

67b Number of tabulated potential files to use, then as many blocks of 68b - 70b as the number of tabulated potentials.

68b Name of the tabulated potential file.

69b Index of the partial (indexing starting with 1) for which this tabulated potential is used for.

70b Weight parameter for this tabulated potential.

71. FNC switch, if 0, no FNC is applied. fnc=1 means normal FNC; fnc=2: normal FNC, but if there are out-of range FNC pairs, then the constraint range will be set according to the largest distance of the given FNC constraint, fnc=3: in case of out-of range pairs, only moves, where the distance of them is

closer to the desired range are accepted (I.N). For option 1-3 *.*fnc* file has to be supplied (II.G). fnc=4 usage of flexible, MD-like molecules (bonded interactions) kept together by forces (I.Q). No *.*fnc* file is needed, but *.*top* describing the molecular topology has to be given (I.Q.1.2(6)). In case of fnc=4 after the FNC switch the define options for the processing of the topology file can be given in the format that the actual define option given in the topology has to be preceded without space by –D (see I.Q.1.2(6)). There can be any number of define options.

72. Number of bins to leave out from the calculation at the small distance end (before the first used histogram bin). Can be a fraction! 0.5 for example means that the lower edge of the first histogram bin starts at bin_size*0.5.

73. x_{max}: largest distance between the particles to include in the histogram calculation in reduced unit (maximum is √3).

74. Number of atoms moved in a RMC move step. If molecular move is used, then the number of moved atoms has to be the same, as the number of atoms in the molecule. Otherwise use 1.

75. Size of the history buffer, each containing the χ^2 for a given phase of the simulation. When the buffer is full, the data is written to disc. Large value means less frequent disc writing, so greater program speed. (0->no history record, 50-100 is recommended).

76. Regulates the interval, the χ^2 is saved to the history buffer. Integer number, denotes the savings to disc between each history buffering, 0 means history is recorded at each saving.

77. If 0, normal atomic RMC, if 1, molecular move is applied. For this later the program have to be compiled with an adequate *makemovecus.cpp* file, and the *.*cus* file has to be given with the molecular move related parameters (see I.D).

78. If the histogram file *.*hgm*, {and the *.*cnc* and *.*acn* files if there is/are (average) coordination number constraint(s)} and the *.*tca* file if the moveout option is on} are available, in case of (1) loading of them will be attempted, and if they are compatible with the given constraints and parameters, then initial histogram calculation will not be done. In case of option 0 initial histogram calculation will be performed, and if they were existing files, they will be overwritten. The values of the initial calculation will be saved in the **start.hgm*, **start.cnc* and **start.acn* files as well to be preserved. It has to be emphasized that in case of loading the histogram and coordination constraint files, the validity of the actual values cannot be checked, so care must be taken to use files corresponding to the *.*cfg* and/or *.*bcf* files! If local invariance is calculated, then the loading of the *.*lhgm* file is attempted as well in case of option 1. If potential is calculated, then the loading of the binary *.*pot* file is attempted as well.

79. The desired number of atoms in a grid cell (see I.E). It is preferable to set the desired maximum number of particles in a grid cell in the *.*dat* file to a relatively low value (5-10) depending of course on the system size to ensure at least 5 or preferably more grid cell in each direction.

80. Only has meaning for multi-component systems, where particles from different types can be swapped with each other to help the mixing of the simulation box. First swap fraction has to be given, a real number (between 0-1), which regulates the fraction of swaps related to the moves. In case of (1) only swaps, (0) no swap at all. After this in case of swaps as many integer as the number of mixed partials (their number is *ntypes*(ntypes-1)/2)* specify, which mixed partials can participate in the swap, (0) not involved, (1) involved. The order of the partials is the same, as usual, without the 'clean' partials (1-2, 1-3,…2-3,…). Cannot be used together with fnc>0, in case of molecular move!

81. Number of threads for parallel execution. Has to be present even in case the program was compiled for standard consecutive execution, only the value is ignored.

82. Only read, if the program is compiled with the _LOCAL_INV compiler option (I.P): number of intervals (N_{loc}) (only one can be given for bin-based calculation, for distance-based the intervals are consecutive); sigma for each interval, calculation mode (0: bin-based, 1: distance-based. default,); n+1 real values, loc_ratio_n which in case of the bin-based calculation mean the starting and limiting reduced distances, for which the local invariance is calculated, the histograms are stored only between them. In case of the distance-based approach loc_ratio_n and $n+1$, they represent the fraction of neighbour atoms (the same fraction value resulting in different atom number for each type according to the different number of atoms/type) to begin and end the local invariance calculation with. If mode is not given after sigma, than mode=1 is assumed, and the local histogram is calculated for the same interval, as the normal histogram.

83. Only read, if the program was compiled with _NO_PERIODIC compiler option (I.R). The radius of the spherical sample is given here in Angstrom. Optionally 1 can be given to recentre the sample in the simulation box, if 0 is given or nothing, then the sample will not be recentred

84. Only read, if the program was compiled with VIBR_AMP compiler option (I.S). The sigma values in Angstrom of the Gaussian distributions for each atom type describing the atomic displacement distribution.

85. Optionally can be given OLD_OUT=1 for write old format output to *.*out* file, or 0 or omitting this line not to create the file. From version 1.6 the compiler option _OLDFORMAT_OUT was abolished, the code is always compiled to be able to create the *.*out* file, whether to create it or not is controlled here.

86. Optionally can be given SUM_PPCF=1 for sum the ppcf and write it to the end of the *.*ppcf* file, or 0 or omitting this line not to. From version 1.6 the compiler option _SUM_PPCF was abolished, the code is always compiled to be able to sum the ppcf, whether to do it or not is controlled here.

C.2. Free format **.dat* file

The first line should contain #003 (or for the older version #002, see II.C) at the beginning of the line, indicating that this is a free format data file! The concept of this input is that the input data is divided into sections, and each section will have a dedicated tag [TAG] in the free format *.*dat* file. (A tag has to be enclosed in square brackets!) The order of the sections is arbitrary, and the lines following a section tag will belong to those section regardless empty lines between them, while a new section tag is found.

Tag [GENERAL] is mandatory for a simulation, without experimental section hard-sphere RMC can be run, which can be used to create a random configuration.

In a section the data is preceded by a keyword and the keyword should be followed by a = sign. The order of keywords in a section is arbitrary, but the order of data belonging to a keyword is fixed. Most of the keywords indicate one parameter, but there are a few combined keywords, where more than one parameter is expected. The naming convention regarding the keywords is that if hyphen is used then it is a single keyword indicating one parameter, in combined keywords the different parts are separated by underscore. (For example SIGMA-MASTER is a single keyword, while COORDNUM_FRACT_SIGMA-MASTER is a combined keyword with three parts, expecting three parameters.) If more than one parameter is expected after a keyword, then it can be split into several lines, the reading of data belonging to a keyword will end if a punctuation mark !, a new keyword or a new tag is found.
Information about the implemented tags can be seen by starting the program with
exename *–helptags*

about all the keywords:
exename *–helpkeys*

about all the keywords belonging to the specified tag:
exename *–helpkeys tag*

about the mandatory keywords:
exename *–helpkeysman*

about the optional keywords:
exename *–helpkeysop*

and about a specific keyword:
exename *–helpkey key*

Some sections (information about data sets and constraints) can have multiple instances; this is indicated by the multiplicity in the help. In these cases the number of instances gives the number of data sets or constraints. In case of the other sections which have multiplicity=1, theoretically the section can be split (meaning that the section tag occur multiple time), but the data will belong nevertheless to the same section. Same goes for keywords, there are keywords that can occur only once in a section. If they are repeated, then always the last one will be used! There are keywords, which can occur multiple times, and their number indicate the number of instances what they are referring to (as for example the number of sub-constraints for a coordination constraint). This is indicated for a keyword, in the help.

Most of the parameters given in the **.dat* file have a corresponding keyword, but some (mostly those, which are indicating number of sets, constraints…) are calculated from the entries found in the file. Some parameters have alternative versions; this is indicated in the help as well. Only some cases, where the concept somewhat changed compared to the fixed format input are discussed below.

In case of the fixed format input file the fnc switch with fnc=4 value indicates that flexible molecules should be used. In the free format the FNC-TYPE keyword will relate only to real fnc or the lack of it having only four different values: NONE corresponding to fnc=0 meaning that no fnc is used, NORMAL corresponding to fnc=1, ADJUST to fnc=2 and MOVE-IN to fnc=3. The usage of flexible molecules in free format should be indicated by the [BPOT], even if it is not followed by any more keywords, as all of its keywords are optional.

The concept of giving the sigma parameters and the leading (potential) series index has somewhat changed. In the RMC_POT program beside the normal sigma parameter for weighing the data sets and constraints there is the possibility to use scalable sigma and the index of the leading series (see 0), which should be provided, if scaling is applied. This is done in case of the fixed format input that the leading series index has to be specified (or the first is assumed as default), and the scalable sigma is indicated by giving a negative fraction. Here this concept was changed concerning the input, but nothing is changed regarding the scalable sigma concept. No leading series index, or negative value indicating scalable sigma is used, instead there are three alternatives giving a sigma value:
SIGMA specify the normal sigma, SIGMA-MASTER the sigma of the leading series, and SIGMA SCALABLE the scalable sigma, and there is no need for the negative sign therefore. (These are either the keywords, or part of a longer keyword, but the meaning is the same.) Therefore only one (…_)SIGMA_MASTER should be specified in the data file!

In case of the LJ potential, where the number of expected sigma parameters for the LJ and Coulomb interactions are depending on the NB weight mode: in case of 0, 2, -2, 3, -3 one entry is expected for both LJ and Coulomb (although in case of 2 or -2 the Coulomb weight parameter has to be present, but it is not used), while in case of 1 or -1 *npartials* number is expected. All of these can be normal or scalable, and there can be given a leading potential series index (if not given 1 is assumed) as well in the fixed format. In the free format the weight mode and the leading potential series index is not given explicitly. The keyword SIGMA-OVERALLPOT indicates the case NB weight mode=2, and in this case one sigma parameter is read. For the other weight modes the VDW-SIGMA and COUL-SIGMA keyword can be used. If only one sigma parameter is given after the keyword, then the NB weight mode is set to 0, (all the partials are weighed with the same sigma). If in case of multicomponent systems more than one, preferably *npartials* number of sigma parameters are found, then the NB weight mode is set to 1, and the partials are displayed and weighed separately. If less then *npartials* sigma parameters are given, then the last given value will be used instead of the missing ones. The COUL-SIGMA can be omitted; in this case the values of the VDW-SIGMA values are used for them. Although these omission are possible but not encouraged as they easily can lead to problems! Regarding the differentiating between the normal and scalable sigma and giving the leading potential series index: precede the sigma values with ‘m’ for master (for example m0.001) or ‘s’ for scalable sigma (for example s0.5), or ‘ms’ (like ms0.5) for a scalable leading series. There should be only one preceding ‘m’ among the sigma parameters related to the potential!

Regarding the minus sign of the NB weight mode, it was used to indicate in the fixed format file that a low limit fraction for the χ^2 should be read as well. This is achieved by a separate keyword CHI-LOWLIMIT-RATIO in the free format data file. The default values for the keywords are located in the *units.h* file of the code, and displayed with the information about the keywords.
The implemented tags are the following:

```
Tag name:          AENET  Description:  To use AENET
                          Multiplicity: 1
Tag name:        AVCOORD  Description:  Average coordination constraint
                          Multiplicity: Number of average coordination constraints
Tag name:           BPOT  Description:  To use bonded potential
```

```
                        Multiplicity: 1
Tag name:          BVS  Description:  Bond valence sum constraint
                        Multiplicity: Number of bond valence sum constraints
Tag name:         CONC  Description:  Common neighbour constraint
                        Multiplicity: Number of common neighbour constraints
Tag name:        COORD  Description:  Coordination constraint
                        Multiplicity: Number of coordination constraints
Tag name:          COS  Description:  Cosine distribution of bond angles constraint
                        Multiplicity: Number of cosine distribution of bond angles constraints
Tag name:     CUSTMOVE  Description:  To use custom move
                        Multiplicity: 1
Tag name:          END  Description:  End of file processing
                        Multiplicity: 1
Tag name:          EXP  Description:  Parameters for one experimental set
                        Multiplicity: Number of experiments
Tag name:      GENERAL  Description:  General parameters of the run
                        Multiplicity: 1
Tag name:       LOCINV  Description:  To use local invariance calculation
                        Multiplicity: 1
Tag name:        NBPOT  Description:  To use non-bonded potential
                        Multiplicity: 1
Tag name:        NOPER  Description:  To use no periodic boundary conditions
                        Multiplicity: 1
Tag name:          SNC  Description:  Second neighbour constraint
                        Multiplicity: Number of second neighbour constraints
Tag name:         SWAP  Description:  To use swapping of atoms
                        Multiplicity: 1
Tag name:      VIBRAMP  Description:  To use bonded potential
                        Multiplicity: 1
```

The implemented keywords are the following:
Free format data input
Implemented keywords:

```
1-4-SCALE-VDW_COUL
        Description: Scaling factor for the 1-4 vdW and Coulomb interactions
        Status: Optional
        Tag name(s) where it can be used: NBPOT
        Acceptable values: two real numbers
        Default value: 1.000000
ABSORBER-TYPE
        Description: Type of the absorbing particle for EXAFS
        Status: Mandatory for TYPE=EXAFS
        Tag name(s) where it can be used: EXP
        Acceptable values: integer number
AENET-STEP
        Description: Calculate ANN potential in each AENET-SETP simulation step
        Status: OPTIONAL
        Tag name(s) where it can be used: AENET
        Acceptable values: integer number
        Default value: 5
ATOMS-IN-GRIDCELL
        Description: Number of atoms in a grid cell
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 5
AXS-INDEX_FPRIME_FRACTION
        Description: Only for I(Q) fitting, indicating this is an anomalous X-ray
        scattering data set. Index is the atom type index (starting with 1) for the
        edge, FPRIME is the real part of the E-dependent scattering factor
        f'(E) for this element, f(Q)+f'(E) is used during coefficient calculation.
        FRACTION indicates the fraction of f'(E): varying f'(E) between f'(E)-
        f'(E)*FRACTION -> f'(E)+f'(E)*FRACTION in every AXS-SHIFTSTEP. step.
        Status: Optional for TYPE=XRD
        Tag name(s) where it can be used: EXP
```

```
        Acceptable values: one integer, two real numbers
AXS-SHIFTSTEP
        Description: Shift f' for AXS I(Q) fitting sets in each AXS-SHIFTSTEP step
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 1000
BACKSCATT-FILE
        Description: Name of the file containing the (k,r)-dependent coefficients
        Status: Mandatory for TYPE=EXAFS
        Tag name(s) where it can be used: EXP
BIN-SHIFT
        Description: Number of bins to leave out from the calculation at the
        beginning of the histogram
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: 0.000000
CENT-TYPE
        Description: Type of central atom
        Status: Mandatory
        Tag name(s) where it can be used: AVCOORD, COORD, COS
        Acceptable values: integer number
CENT-TYPE_CHARGE
        Description: Type of central atom, charge of central atom (only needed, if
        default value for R0 and b should be used
        Status: MANDATORY
        Tag name(s) where it can be used: BVS
        Acceptable values: two integer numbers, second optional
CHEMICAL-SYMBOLS
        Description: Chemical symbols of the atom types in the order they are
        represented in the configuration, needed if X-ray atomic scattering factors,
        Compton-scattering or neutron coefficients are calculated
        are calculated
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: ntypes string, case sensitive
CHI-LOWLIMIT-RATIO
        Description: The chi2_pot will not decrease under fraction*starting chi2_pot
        Status: Optional
        Tag name(s) where it can be used: NBPOT
        Acceptable values: real number, NONE
        Default value: NONE
CHIK-POWER
        Description: (k/k_MAX)^CHIK-POWER for weighing the EXAFS data
        Status: Optional
        Tag name(s) where it can be used: EXP
        Acceptable values: integer number
        Default value: 3
COLL-NUMBER
        Description: Number of configurations to collect at the end of the run
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 0
COLL-SAVE-TIME
        Description: A configuration will be saved at each COLL-SAVE-TIME,
        it should be set to an integer multiple of SAVE-TIME
        Alternatives: COLL-STEP-FACTOR
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: COLL-STEP-FACTOR or its default: 1
COLL-STEP-FACTOR
        Description: A configuration will be saved at each COLL-STEP-FACTOR*SAVE-TIME
        Alternatives: COLL-SAVE-TIME
```

```
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 1
COMP-OPTION
        Description: A compiler option to use during the reading of the topology
        file, there can be more, than one
        Status: Optional
        Tag name(s) where it can be used: BPOT
        Acceptable values: strings
        Default value: NONE
COMPTON
        Description: Only for I(Q) fitting, indicating whether Compton-scattering
        contribution should be used (0: no, 1: yes)
        Status: Optional for TYPE=XRD
        Tag name(s) where it can be used: EXP
        Acceptable values: 0, 1
        Default value: 1
COMPTON-SUPR-COEFF
        Description: Whether to vary alpha during I(Q) fit (0: no, 1: yes)
        Status: Optional for TYPE=XRD
        Tag name(s) where it can be used: EXP
        Acceptable values: 0, 1
        Default value: 0
CONST-SUBTRACT
        Description: Constant to subtract from the data points
        Status: Optional
        Tag name(s) where it can be used: EXP
        Acceptable values: real number
        Default value: 0.000000
COORDNUM_FRACT_SIGMA
        Description: Desired coordination number, fraction and sigma parameter
        Alternatives: COORDNUM_FRACT_SIGMA-MASTER, COORDNUM_FRACT_SIGMA-SCALABLE
        Status: One of them is mandatory
        Tag name(s) where it can be used: CONC, COORD, SNC
        Acceptable values: integer, real, real number
COORDNUM_FRACT_SIGMA-MASTER
        Description: Desired coordination number, fraction and sigma parameter for
        leading series
        Alternatives: COORDNUM_FRACT_SIGMA, COORDNUM_FRACT_SIGMA-SCALABLE
        Status: One of them is mandatory
        Tag name(s) where it can be used: CONC, COORD, SNC
        Acceptable values: integer, real, real number
COORDNUM_FRACT_SIGMA-SCALABLE
        Description: Desired coordination number, fraction and scalable sigma
        parameter
        Alternatives: COORDNUM_FRACT_SIGMA, COORDNUM_FRACT_SIGMA-MASTER
        Status: One of them is mandatory
        Tag name(s) where it can be used: CONC, COORD, SNC
        Acceptable values: integer, real, real number
COORDNUM_SIGMA
        Description: Desired average coordination number and sigma parameter
        Alternatives: COORDNUM_SIGMA-MASTER, COORDNUM_SIGMA-SCALABLE
        Status: One of them is mandatory
        Tag name(s) where it can be used: AVCOORD
        Acceptable values: integer, real, real number
COORDNUM_SIGMA-MASTER
        Description: Desired average coordination number and sigma parameter for
        leading series
        Alternatives: COORDNUMBSIGMA, COORDNUM_SIGMA-SCALABLE
        Status: One of them is mandatory
        Tag name(s) where it can be used: AVCOORD
        Acceptable values: integer, real, real number
COORDNUM_SIGMA-SCALABLE
        Description: Desired average coordination number and scalable sigma
        parameter
```

```
        Alternatives: COORDNUM_SIGMA, COORDNUM_SIGMA-MASTER
        Status: One of them is mandatory
        Tag name(s) where it can be used: AVCOORD
        Acceptable values: integer, real, real number
COUL-SIGMA
        Description: Weight parameters for the Coulomb interactions;
        if  NB weight mode=1: npartials number is expected, otherwise one value.
        If key not given, vdW-SIGMA weight will be applied!
        Alternatives: SIGMA-OVERALLPOT
        Status: Either this and VDW-SIGMA, or SIGMA-OVERALLPOT mandatory
        Tag name(s) where it can be used: NBPOT
        Acceptable values: One or npartials real numbers.
        Precede the number with m for leading series, with s for scalable
CREATE-OUT
        Description: Whether to write old format *.out file (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: 0, 1
        Default value: 0
CTYPE_FTYPE_FROM_TO
        Description: Type of central and first neighbour atoms, minimum and maximum
        distance in A from each other
        Status: Mandatory
        Tag name(s) where it can be used: SNC
        Acceptable values: two integer and two real numbers
CUSTOM-LINE
        Description: For new format custom move file, CUSTOM-LINE = should proceed
        the data for each line
        Status: Mandatory for new format custom files for each line
        Tag name(s) where it can be used: CUSTOM
        Acceptable values:
CUSTOM-SFACTORTABLE
        Description: Use the given file for custom surface factors
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: file name
        Default value: built-in is used
CUTOFF-AENET
        Description: Cutoff for the ANN potential in A
        Status: OPTIONAL
        Tag name(s) where it can be used: AENET
        Acceptable values: real number
        Default value: same as used in potential construction
CUTOFF-TAB
        Description: Cut-off for the tabulated potential interactions in A
        Status: Optional
        Tag name(s) where it can be used: NBPOT
        Acceptable values: real number
        Default value: 1.0 in reduced unit corresponding to boxedge A
CUT-OFF
        Description:  Distance  in  A  between  the  atom  pairs  for  each  partial,
        if MIN or MIN-ALL is given, then the minimal distance for each partial is
        determined automatically and set, see the difference in the manual
        Status: OPTIONAL
        Tag name(s) where it can be used: GENERAL
        Acceptable values: npartials real numbers, MIN or MIN-ALL
        Default value: MIN-ALL
DATAFILE
        Description: Name of the data file
        Status: Mandatory
        Tag name(s) where it can be used: EXP
        Acceptable values: arbitrary string
DEBUG
        Description: Write debug information (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
```

```
        Acceptable values: 0, 1
        Default value: 0
DELTAE0_NGRID
        Description: Maximum E0-shift in eV and number of grid points for the
        shift in one direction in case of EXAFS
        Status: Optional for TYPE=EXAFS
        Tag name(s) where it can be used: EXP
        Acceptable values: real, integer number
        Default value: 0.000000, 0 no shift
DCOSTH
        Description: Spacing in cos(theta) space, has to be the same for all the COS
        constraints
        except DISTRIB-TYPE = filename. If different, last value is used
        Status: Optional
        Tag name(s) where it can be used: COS
        Acceptable values: real number
        Default value: 0.050000
DISTRIB-DEGREES
        Description: Desired angle in degree
        Status: Mandatory if DISTRIB-TYPE is not read from file
        Tag name(s) where it can be used: COS
        Acceptable values: real number
DISTRIB-TYPE
        Description: Distribution type for cosine distribution constraint
        Status: Optional
        Tag name(s) where it can be used: COS
        Acceptable values: UNIFORM, GAUSSIAN, filename, ABSENT
        Default value: GAUSSIAN
DISTRIB-WIDTH
        Description: Wcontrol parameter
        Status: Mandatory if DISTRIB-TYPE is not read from file
        Tag name(s) where it can be used: COS
        Acceptable values: real number
E0GRID-FROM_TO_START
        Description: Using only the range of E0 grid points for EXAFS specified by
        FROM, TO, should be between -NGRID -> +NGRID, therefore DELTAE0_NGRID is
        needed as well.
        Optionally, START is the grid index of the starting grid point.
        Status: Optional for TYPE = EXAFS
        Tag name(s) where it can be used: EXP
        Acceptable values: three integer numbers
        Default value: 0, 0, 0: no shift, or symmetric range, if DELTAE0_NGRID is
        given
EPSILON-NONLIN
        Description: Iteration stops, if the fitted parameters change less than this
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: 0.001000
EXAFS-SHIFTSTEP
        Description: Shift E0 and therefore k for selected EXAFS data set in each
        EXAFS-SHIFTSTEP step
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 1000
FACTOR-NONLIN
        Description: Factor to change fudge for Levenberg-Marquardt non-linear
        regression
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: 10.000000
FNC-TYPE
        Description: Type of the FNC constraint
        Status: Optional
```

```
        Tag name(s) where it can be used: GENERAL
        Acceptable values: NONE, NORMAL, ADJUST, MOVE-IN
        Default value: NONE
FPRIME-INDEX_FPRIME
        Description: In case of X-ray data sets, index of the atom type to use f' for
        (starting with 1),
        then f' for this atom type in this experiment. If index 0 no f' is used.
        f' is the real part of the E-dependent scattering factor part for this
        element, f(Q)+f'(E) is used instead of f(Q) for coefficient calculation for
        this type
        Status: Optional for TYPE=XRD
        Tag name(s) where it can be used: EXP
        Acceptable values: one integer and one real number
FRACT
        Description: Swap fraction
        Status: Mandatory
        Tag name(s) where it can be used: SWAP
        Acceptable values: real number
G-RANGE
        Description: The value of the first and last g-data points to use for an
        electron diffraction data set
        Alternatives: POINT-RANGE
        Status: Optional
        Tag name(s) where it can be used: EXP
        Acceptable values: 2 real numbers
        Default value: whole data range
GAUSSIAN-SIGMA
        Description: Sigma parameter in A controlling the width of the Gaussian
        distribution,
        one values for each atom type
        Status: Mandatory
        Tag name(s) where it can be used: VIBRAMP
        Acceptable values: ntypes real numbers
HALFBOX
        Description: Half length of the simulation box in A
        Alternatives: NDENS
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: read from *.cfg
HST-SAVE-TIME
        Description: History is buffered with this interval, should be or set to be
        the integer multiple of SAVE-TIME
        Alternatives: HST-STEP-FACTOR
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Default value: HST-STEP-FACTOR or its default: 1
HST-STEP-FACTOR
        Description: Number of savings to disc between history buffering
        Alternatives: HST-SAVE-TIME
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 1
HST_BUFFSIZE
        Description: Size of the history buffer
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 0
I(Q)_A_B_ALPHA
        Description: Indicator to fit I(Q) instead of F(Q), if custom values are
        needed should be supplied here.
        If b and alpha custom value is needed, something for a value should be given
        as well!
        Status: Optional for TYPE=XRD
```

Tag name(s) where it can be used: EXP
Acceptable values: one integer and optionally three real numbers

I(Q)BACKG_MU_DMU-MAX
Description: Indicator to use background correction for I(Q) fitting, if custom values used, should be supplied here: mu and dmu_max.
Status: Optional for TYPE=XRD
Tag name(s) where it can be used: EXP
Acceptable values: one integer and optionally two real numbers
Default value: 0, 0.700000, 0.200000

I(Q)BACKG-STEP
Description: Change the background correction factor mu in every I(Q)BACKG-STEP, step
Status: Optional
Tag name(s) where it can be used: GENERAL
Acceptable values: integer number
Default value: 1000

IND_ANN-FILE_TYPE-NAME
Description: Atom type index, ANN potential file name, optionally atom type name, only required, if it is different than in CHEMICAL-SYMBOLS
Status: MANDATORY
Tag name(s) where it can be used: AENET
Acceptable values: integer, string, string

IND_SIGMA-MASTER_FILE
Description: The index of the partial for this tabulated potential, sigma, file name with tabulated potential
Alternatives: IND_SIGMA-SCALABLE_FILE, IND_SIGMA_FILE
Status: One of them is mandatory
Tag name(s) where it can be used: NBPOT
Acceptable values: integer, real, string

IND_SIGMA-SCALABLE_FILE
Description: The index of the partial for this tabulated potential, sigma, file name with tabulated potential
Alternatives: IND_SIGMA-MASTER_FILE, IND_SIGMA_FILE
Status: One of them is mandatory
Tag name(s) where it can be used: NBPOT
Acceptable values: integer, real, string

IND_SIGMA_FILE
Description: The index of the partial for this tabulated potential, sigma, file name with tabulated potential
Alternatives: IND_SIGMA-MASTER_FILE, IND_SIGMA-SCALABLE_FILE
Status: One of them is mandatory
Tag name(s) where it can be used: NBPOT
Acceptable values: integer, real, string

ISOTOPE-COUNT_SYMBOLS_RATIOS
Description: For a chemical element use as many isotopes for a neutron data set as given by isotope count, the isotopes should be specified in the format like C14, sum of the ratios has to be 1!
Can give as many lines as the number of components for a neutron data set.
Status: Optional for TYPE=ND
Tag name(s) where it can be used: EXP
Acceptable values: integer, string(s), real number(s)
Default value: natural isotope abundance

K-RANGE
Description: The value of the first and last k-data points to use for an EXAFS data set
Alternatives: POINT-RANGE
Status: Optional
Tag name(s) where it can be used: EXP
Acceptable values: 2 real numbers
Default value: whole data range

LAMBDA-NONLIN
Description: Fudge factor for Levenberg-Marquardt non-linear regression
Status: Optional
Tag name(s) where it can be used: GENERAL
Acceptable values: real number

```
        Default value: 0.001000
LAST-ACC
        Description: Number of accepted moves for the run,
        (in _TEST_MODE only LAST-GEN is an alternative)
        Alternatives: LAST-GEN, RUN-TIME
        Status: OPTIONAL
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 10000
LAST-GEN
        Description: Number of generated moves for the run,
        (in _TEST_MODE only LAST-ACC is an alternative)
        Alternatives: LAST-ACC, RUN-TIME
        Status: OPTIONAL
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 10000
LEAD-ANGLE
        Description: The index of the leading potential interaction in its own type
        Alternatives: LEAD-BOND, LEAD-HARMDIH, LEAD-PERDIH, LEAD-RBDIH
        Status: One of them is mandatory
        Tag name(s) where it can be used: BPOT
        Acceptable values: one integer number
        Default value: Setting it to 1 counting all the potential-related series
LEAD-BOND
        Description: The index of the leading potential interaction in its own type
        Alternatives: LEAD-ANGLE, LEAD-HARMDIH, LEAD-PERDIH, LEAD-RBDIH
        Status: One of them is mandatory
        Tag name(s) where it can be used: BPOT
        Acceptable values: one integer number
        Default value: Setting it to 1 counting all the potential-related series
LEAD-HARMDIH
        Description: The index of the leading potential interaction in its own type
        Alternatives: LEAD-ANGLE, LEAD-BOND, LEAD-PERDIH, LEAD-RBDIH
        Status: One of them is mandatory
        Tag name(s) where it can be used: BPOT
        Acceptable values: one integer number
        Default value: Setting it to 1 counting all the potential-related series
LEAD-PERDIH
        Description: The index of the leading potential interaction in its own type
        Alternatives: LEAD-ANGLE, LEAD-BOND, LEAD-HARMDIH, LEAD-RBDIH
        Status: One of them is mandatory
        Tag name(s) where it can be used: BPOT
        Acceptable values: one integer number
        Default value: Setting it to 1 counting all the potential-related series
LEAD-RBDIH
        Description: The index of the leading potential interaction in its own type
        Alternatives: LEAD-ANGLE, LEAD-BOND, LEAD-HARMDIH, LEAD-PERDIH
        Status: One of them is mandatory
        Tag name(s) where it can be used: BPOT
        Acceptable values: one integer number
        Default value: Setting it to 1 counting all the potential-related series
LJ-C6-EACH
        Description: LJ first potential parameters in A^6, used for combination
        rule=0
        Alternatives: LJ-C6-GMEAN, LJ-SIG-AMEAN, LJ-SIG-GMEAN
        Status: One of them is mandatory
        Tag name(s) where it can be used: NBPOT
        Acceptable values: nGROMACS partials real numbers
LJ-C6-GMEAN
        Description: LJ first potential parameters in A^6, used for combination
        rule=1
        Alternatives: LJ-SIG-AMEAN, LJ-C6-EACH, LJ-SIG-GMEAN
        Status: One of them is mandatory
        Tag name(s) where it can be used: NBPOT
        Acceptable values: nGROMACS types real numbers
```

LJ-CPOW-EACH
- Description: LJ second potential parameters in kJ/mol·A^{12}, used for combination rule=0
- Alternatives: LJ-CPOW-GMEAN, LJ-EPS-GMEAN
- Status: One of them is mandatory
- Tag name(s) where it can be used: NBPOT
- Acceptable values: nGROMACS partials real numbers

LJ-CPOW-GMEAN
- Description: LJ second potential related parameters in kJ/mol·A^{12}, used for combination rule=1, see the manual
- Alternatives: LJ-CPOW-EACH, LJ-EPS-GMEAN
- Status: One of them is mandatory
- Tag name(s) where it can be used: NBPOT
- Acceptable values: nGROMACS types real numbers

LJ-EPS-GMEAN
- Description: LJ epsilon parameters in A, used for combination rule=2 and 3
- Alternatives: LJ-CPOW-GMEAN, LJ-CPOW-EACH
- Status: One of them is mandatory
- Tag name(s) where it can be used: NBPOT
- Acceptable values: nGROMACS types real numbers

LJ-POW
- Description: Power of the LJ repulsion term
- Status: Optional
- Tag name(s) where it can be used: NBPOT
- Acceptable values: integer number
- Default value: 12

LJ-SIG-AMEAN
- Description: LJ sigma parameters in A, used for combination rule=2
- Alternatives: LJ-C6-GMEAN, LJ-C6-EACH, LJ-SIG-GMEAN
- Status: One of them is mandatory
- Tag name(s) where it can be used: NBPOT
- Acceptable values: nGROMACS types real numbers

LJ-SIG-GMEAN
- Description: LJ sigma parameters in A, used for combination rule=3
- Alternatives: LJ-C6-GMEAN, LJ-SIG-AMEAN, LJ-C6-EACH
- Status: One of them is mandatory
- Tag name(s) where it can be used: NBPOT
- Acceptable values: nGROMACS types real numbers

LOC-INT-FROM_SIGMA-SCALABLE
- Description: For BIN-BASED calculation give it once: starting reduced distance for the interval,
  for DISTANCE-BASED calculation as many times as the number of intervals: starting atom ratio for the interval; scalable sigma
- Alternatives: LOC-INT-FROM_SIGMA, LOC-INT-FROM_SIGMA-MASTER
- Status: One of them is mandatory
- Tag name(s) where it can be used: LOCINV
- Acceptable values: two real numbers

LOC-INT-TO
- Description: Include only once setting the end of the (last) interval. Largest reduced distance included in local invariance histogram calculation
- Status: Optional
- Tag name(s) where it can be used: LOCINV
- Acceptable values: real number
- Default value: Same as for normal histogram

LOC-MODE
- Description: Calculation mode for local invariance: bin-based (0), distance-based (1)
- Status: Optional
- Tag name(s) where it can be used: LOCINV
- Acceptable values: BIN-BASED, DISTANCE-BASED
- Default value: DISTANCE-BASED

LOC-R-SPACING
- Description: Local invariance histogram spacing in A
- Status: Optional
- Tag name(s) where it can be used: LOCINV
- Acceptable values: real number

```
        Default value: default normal histogram bin size
LOG-NONLIN-STEPS
        Description: Write non-linear regression steps, if there is any
        (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: 0, 1
        Default value: 0
MAX-MOVES
        Description: Maximum move for each atom type in A
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: ntypes real number
        Default value: 0.100000
MAX-NITER-NONLIN
        Description: Maximum number of iteration in case of I(Q) fitting with non-
        linear regression
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 1000
MOVEOUT
        Description: Whether to use moveout (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: boolean
        Default value: 1
NB-TYPE
        Description: Type of the non-bonded potential
        Status: Optional
        Tag name(s) where it can be used: NBPOT
        Acceptable values: LJ, TABULATED
        Default value: LJ
NDENS
        Description: Number density in 1/A^3
        Alternatives: HALFBOX
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: calculated from *.cfg
NEIGH-TYPE_FROM_TO
        Description: Type of neighbour atom, minimum and maximum distances in A
        Include twice for COS, once for AVCOORD, for COORD as many times as the
        number of neighbour types
        Status: Mandatory
        Tag name(s) where it can be used: AVCOORD, COORD, COS
        Acceptable values: an integer and two real numbers
NEIGH-TYPE_RMAX_R0_B_CHARGE
        Description: Type of neighbour atom, maximum distance in A, R0 (bond length)
        in A (optional to give value to use), b parameter in A (optional), allocated
        charge (optional). Give -1 for R0, and/or b if default should be used, and
        give the assumed charge of the neighbour for R0 default value determination.
        Include as many times, as the number of different neighbour types.
        Status: MANDATORY
        Tag name(s) where it can be used: BVS
        Acceptable values: an integer, two real, one optional real and one optional
        integer number
NGR-TYPES
        Description: Number of GROMACS types
        Status: Optional
        Tag name(s) where it can be used: NBPOT
        Acceptable values: integer number
        Default value: number of RMC types
NMOVED-ATOMS
        Description: Number of atoms to move in a single move
        Status: Mandatory
```

- Tag name(s) where it can be used: CUSTMOVE
- Acceptable values: integer number

PAIRS
- Description: Atom types to swap
- Status: Mandatory
- Tag name(s) where it can be used: SWAP
- Acceptable values: ALL, Type-ANY, Type1-Type2

PARTIAL-COEFFS
- Description: Coefficients for the partials
- Status: Mandatory for TYPE=GR and optional for ND
- Tag name(s) where it can be used: EXP
- Acceptable values: real number

POINT-RANGE
- Description: First and last data points to use
- Alternatives: G-RANGE, K-RANGE, Q-RANGE, R-RANGE
- Status: Optional
- Tag name(s) where it can be used: EXP
- Acceptable values: 2 integer numbers
- Default value: whole data range

POLY-BACK-FLAGS
- Description: Whether to use polinomial background correction, a value for each (offset, linear, quadratic, cubic) (0: no, 1: yes)
- Status: Optional
- Tag name(s) where it can be used: EXP
- Acceptable values: 0, 1
- Default value: 0

PPCF-AVERAGE
- Description: Whether to write summed ppcfs to the *.ppcf file as well at each saving (0: no, 1: yes)
- Status: Optional
- Tag name(s) where it can be used: GENERAL
- Acceptable values: 0, 1
- Default value: 0

PRINT-STEP
- Description: Number of generated moves between printing to general output
- Status: Optional
- Tag name(s) where it can be used: GENERAL
- Acceptable values: integer number
- Default value: 10000

PTYPE1_PTYPE2_FROM_TO
- Description: Type of primary1 and primary2 atoms, minimum and maximum distance in A from each other. Include as many times, as the number of second neighbours.
- Status: Mandatory
- Tag name(s) where it can be used: CONC
- Acceptable values: two integer and two real numbers

Q-RANGE
- Description: The value of the first and last Q-data points to use for a neutron or x-ray data set
- Alternatives: POINT-RANGE
- Status: Optional
- Tag name(s) where it can be used: EXP
- Acceptable values: 2 real numbers
- Default value: whole data range

R-POINT-RANGE
- Description: First and last r-data points to use from the EXAFS (k,r) coefficient file
- Alternatives: R-RANGE
- Status: Optional
- Tag name(s) where it can be used: EXP
- Acceptable values: 2 real numbers
- Default value: whole data range

R-RANGE
- Description: The value of the first and last r-data points to use for a g(r) data set
- Alternatives: POINT-RANGE, R-POINT-RANGE

```
        Status: Optional
        Tag name(s) where it can be used: EXP
        Acceptable values: 2 real numbers
        Default value: whole data range
R-SPACING
        Description: Size of the histogram bins in A
        Status: Optional
        Tag name(s) where it can be used: EXP, GENERAL
        Acceptable values: real number
        Default value: 0.100000
R-SWITCH-POWER
        Description: Modify the default value for the _R_SWITCH_GR_MIN_1
        compiler option, if it is on, then
        experimental g(r) data sets will be interpreted as (r^r_switch_power)*
        [g(r)-1], and this will be calculated and fitted
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 2
READ-COEFFS
        Description: Indicator whether to read the X-ray coefficients from the
        experimental data file in case of F(Q)-fitting or f(Q)-s in case of I(Q)
        fitting for the given X-ray data set (0: no, 1: yes)
        For free format *.dat version #002 reading will be the default to maintain
        downward compatibility
        Status: Optional for TYPE=XRD
        Tag name(s) where it can be used: EXP
        Acceptable values: 0, 1
        Default value: 0 from free format *.dat version #003
RECENTRE-SPHERE
        Description: Whether to recentre the spherical sample in the box
        (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: NOPER
        Acceptable values: 0, 1
        Default value: 0
RELOAD
        Description: Whether to load the histogram from disc (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: 0, 1
        Default value: 0
RENORM
        Description: Whether to use renormalization coefficient (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: EXP
        Acceptable values: 0, 1
        Default value: 0
RUN-TIME
        Description: Run time in minutes, ignored in _TEST_MODE
        Alternatives: LAST-ACC, LAST-GEN
        Status: OPTIONAL
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: 300.000000
SAMPLE-RADIUS
        Description: Radius of the spherical sample in A
        Status: Mandatory
        Tag name(s) where it can be used: NOPER
        Acceptable values: real number
SAVE-TIME
        Description: Saving to the disk in minutes
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: 2
```

```
SIGMA
        Description: Sigma parameter
        Alternatives: SIGMA-MASTER, SIGMA-SCALABLE
        Status: One of it is mandatory
        Tag name(s) where it can be used: COS, EXP
        Acceptable values: real number
SIGMA-MASTER
        Description: Sigma parameter for the leading series
        Alternatives: SIGMA, SIGMA-SCALABLE
        Status: One of it is mandatory
        Tag name(s) where it can be used: COS, EXP
        Acceptable values: real number
SIGMA-OVERALLPOT
        Description: Weight parameters for all the potential-related interactions
        (NB weight mode:2)
        Alternatives: VDW-SIGMA and COUL-SIGMA
        Status: Either this, or VDW-SIGMA and COUL-SIGMA mandatory
        Tag name(s) where it can be used: NBPOT
        Acceptable values: real number
SIGMA-SCALABLE
        Description: Scalable sigma parameter
        Alternatives: SIGMA, SIGMA-MASTER
        Status: One of it is mandatory
        Tag name(s) where it can be used: COS, EXP
        Acceptable values: real number
STYPE_FROM1_TO1_FROM2_TO2
        Description: Type of secondary atom, minimum and maximum distance in A to
        P1, to P2,
        (minimum and maximum distance to P2, if not given same used as for P1)
        Include as many times, as the number of secondary types.
        Status: Mandatory
        Tag name(s) where it can be used: CONC
        Acceptable values: an integer and two mandatory and two optional real
        numbers
STYPE_FROM_TO
        Description: Type of second neighbour atom, minimum and maximum distance in
        A to first neighbour,
        Include as many times, as the number of different second neighbour types.
        Status: Mandatory
        Tag name(s) where it can be used: SNC
        Acceptable values: an integer and two real numbers
TEMPERATURE
        Description: Temperature in K, no algorithmic meaning, for tabulated
        potential
        Status: Optional
        Tag name(s) where it can be used: NBPOT
        Acceptable values: real value
        Default value: 298.000000
TERMINATE-NONLIN
        Description: Terminate after this many successive moves ended with failed
        non-linear regression
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 10
THREADS
        Description: Number of threads to use
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: integer number
        Default value: 1
TITLE
        Description: Title of the run
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: arbitrary string
```

```
TOOCLOSE-FRACTION
        Description: What fraction of the moves should be from too close atoms in
        case of MOVEOUT option (between 0-1)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: 0.500000
TYPE
        Description: Type of the experimental set
        Status: Mandatory
        Tag name(s) where it can be used: EXP
        Acceptable values: EXAFS, GR, EDIFF, ND, XRD
USE-RFACTOR
        Description: Whether to use R-factor instead of normal chi2 for this set
        (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: EXP
        Acceptable values: 0, 1
        Default value: 0
VALENCE_SIGMA
        Description: Desired valence, sigma parameter.
        Alternatives: VALENCE_SIGMA-MASTER, VALENCE_SIGMA-SCALABLE
        Status: One of them is mandatory
        Tag name(s) where it can be used: BVS
        Acceptable values: two real numbers
VALENCE_SIGMA-MASTER
        Description: Desired valence, sigma parameter for leading series.
        Alternatives: VALENCE_SIGMA, VALENCE_SIGMA-SCALABLE
        Status: One of them is mandatory
        Tag name(s) where it can be used: BVS
        Acceptable values: two real numbers
VALENCE_SIGMA-SCALABLE
        Description: Desired valence, scalable sigma parameter.
        Alternatives: VALENCE_SIGMA, VALENCE_SIGMA-MASTER
        Status: One of them is mandatory
        Tag name(s) where it can be used: BVS
        Acceptable values: two real numbers
VDW-SIGMA
        Description: Weight parameters for the vdW interactions;
        if NB weight mode=1: npartials number is expected, otherwise one value
        Alternatives: SIGMA-OVERALLPOT
        Status: Either this and COUL-SIGMA, or SIGMA-OVERALLPOT mandatory
        Tag name(s) where it can be used: NBPOT
        Acceptable values: One or npartials real numbers.
        Precede the number with m for leading series, with s for scalable
WRITE-ATOMIC-ENERGY
        Description: Whether to write energy for all the atoms in the *.en file
        Status: OPTIONAL
        Tag name(s) where it can be used: AENET
        Acceptable values: 0, 1
        Default value: 0
WRITE-CNC-DETAIL
        Description: Whether to write the details about the neighbours into the
        *.cncd file
        Status: OPTIONAL
        Tag name(s) where it can be used: COORD
        Acceptable values: 0, 1
        Default value: 0
WRITE-EXAFS-COEFFS
        Description: Whether to write the EXAFS coefficients to the *.expt file,
        only active if E0 shift is applied (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: 0, 1
        Default value: 0
WRITE-LOG
```

```
        Description: Whether to create a log file, only for developers
        (0: no, 1: yes)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: 0, 1
        Default value: 1
XMAX-FACTOR
        Description: Largest distance between the particles to include in the
        histogram calculation in reduced unit, maximum sqrt(3)
        Status: Optional
        Tag name(s) where it can be used: GENERAL
        Acceptable values: real number
        Default value: 1.000000
```

An example free format file will be given here, this contains all the available sections, but obviously does not corresponds to a real simulation.

```
#003
[ GENERAL ]
 TITLE = example_only_for_format_demonstration  ! --Title of the run (default: blank)
 NDENS = 3.160000093151758e-02           ! --Number density A^{-3} (A=angstrom)
units
                 ! alternative: HALFBOX = 1.680614300000000e+01 ! Half box length in
A units;
                 ! (default: determine it from the .cfg file)
 CUT-OFF =
  4.200000 2.100000 2.400000
  3.000000 3.300000
  3.200000       ! --Distance of closest approach for each partials in A (in
11,12,...,22,23.. order);
                 ! Give as many values as the number of partials, or MIN for
automatic cutoff determination
 MAX-MOVES = 0.000000 0.000000 0.000000   ! --Maximal moves for each atomtypes in A
units; (default: 0.100000 for each);
                 ! when you provide less values than required, the last value passed
to the remaining ones
 R-SPACING = 0.100000   ! --R-spacing in A units (default: 0.100000)
 MOVEOUT = 0            ! --Whether to use moveout option (default: 1)
 TOOCLOSE-FRACTION = 0.500000       ! --The fraction of moves to choose from the too
close atomsin case of moveout option (default: 0.500000)
 PRINT-STEP = 1000      ! --Generated number of steps for printing (default: 10000)
 RUN-TIME = 2.000000    ! --Time limit of the simulation in minutes (default:
300.000000)
                 ! alternatives: LAST-GEN = 10000, or LAST-ACC = 10000, (default
RUN-TIME = 300.000000)
 SAVE-TIME = 1.000000         ! --Time step for saving in minutes (default: 2.000000)
 HST_BUFFSIZE = 1       ! --Size of the history buffer in lines (default: 0, no
history)
 HST-STEP-FACTOR = 1    ! --Number of savings between each history buffering for the
.hst file (default: 1 -- means history saved at every 1. saving).
                 ! alternative: HST-SAVE-TIME = 2.000000 ! --History save time to
*.hst file in minutes
                 ! this value is adjusted internally to be an integer multiple of
SAVE-TIME.
 COLL-NUMBER = 0  ! --Number of configuration to collect after RUN-TIME has been
elapsed (default: 0)
 COLL-STEP-FACTOR = 1    ! --Collection step, configuration is collected at every
COLL-STEP_FACTOR. saving (default: 1)
                 ! alternative: COLL-SAVE-TIME = 1.000000 ! --Time step for saving
collected configuration in minutes (default: equals to SAVE-TIME),
                 ! this value is adjusted internally to be an integer multiple of
SAVE-TIME.
 FNC-TYPE = NONE  ! --FNC switch: NONE(default) or NORMAL or ADJUST or MOVE_IN
```

```
 CHEMICAL-SYMBOLS = As Se I        ! --Chemical symbols in the order of the atom
types in the configuration (default: not given)
 BIN-SHIFT = 1.500000        ! --Initial bin shift (default: 0.000000)
 XMAX-FACTOR = 1.414000 ! --Xmax= FACTOR*halbox used in the run (default: 1.000000)
 CUSTOM-SFACTORTABLE = dummy_sfactor.txt  ! --Custom surface factor table is used
(default: built-in is used)
 RELOAD = 1      ! --Whether to load the histogram from file if possible (default:
0)
 ATOMS-IN-GRIDCELL = 5  ! --Maximum number of atoms in a gridcell (default: 5)
 EXAFS-SHIFTSTEP = 1000 ! --Shift E0 for selected EXAFS data sets in each EXAFS-
SHIFTSTEP. simulation step (default: 1000)
 LAMBDA-NONLIN = 0.001000    ! --Lambda fudge factor for Levenberg-Marquardt non-
linear regression (default: 0.001000)
 FACTOR-NONLIN = 10.000000    ! --Factor to change fudge for Levenberg-Marquardt non-
linear regression (default: 10.000000)
 TERMINATE-NONLIN = 10  ! --Terminate after this many successive moves ended with
failed non-linear regression (default: 10)
 MAX-NITER-NONLIN = 1000     ! --Maximum number of iteration in case of I(Q) fitting
with non-linear regression (default: 1000)
 EPSILON-NONLIN = 0.001000    ! --Iteration stops, if the fitted parameters change
less than this (default: 0.001000)
 I(Q)BACKG-STEP = 0          ! --Change mu background correction factor for I(Q)
fitting in each I(Q)BACKG-STEP. simulation step, (default: 1000)
 AXS-SHIFTSTEP = 5           ! --Change the f' for AXS I(Q) fitting in each AXS-
SHIFTSTEP. simulation step, (default: 1000)
 DEBUG = 0       ! --Write debug information (default: 0)
 WRITE-LOG = 1         ! --Whether to create a log file (default: 0)
 LOG-NONLIN-STEPS = 0   ! --Write non-linear regression steps, if there is any
(default: 0)
 WRITE-EXAFS-COEFFS = 1       ! --Whether to write the EXAFS coefficients into *.expt
(only active if dE0 shift is applied (default: 0)
 R-SWITCH-POWER = 1           ! --Modify the default value for the _R_SWITCH_GR_MIN_1
compiler option, if it is on, then
                 ! experimental g(r) data sets will be interpreted as
(r^r_switch_power)*[g(r)-1], and this will be calculated and fitted, default: 2)
 THREADS = 2           ! --Total number of threads to use (default: 1)
 CREATE-OUT = 0        ! --Create old .out output format containing ppcf's, psq's
and agreement to experimental data in single file (default: 0)
 PPCF-AVERAGE = 0 ! --Append the average of ppcf's of each saved configuration to the
.ppcf file (default: 0)

[ EXP ]
 TYPE = GR       ! --Type of the experiment, **MANDATORY ITEM** (it should be EXAFS,
GR, EDIFF, ND, XRD
 DATAFILE = c4.gr ! --File name containing the experimental data, **MANDATORY ITEM**
 R-RANGE = 0.050000 9.950000  ! --Range used for fit (default: 0.050000..9.950000,
the whole range of the data)
                 ! alternative: POINT-RANGE = 1 100 ! The applied point range to fit
(default: 1..100, the whole range of the data)
 R-SPACING = 0.100000   ! --R-spacing in A units (default: 0.100000)
 CONST-SUBTRACT = 0.000000    ! --Constant to subtract (default: 0)
 PARTIAL-COEFFS =
  0.0218000000000000 0.2520000000000000 0.1262000000000000
  0.1000000000000000 0.3000000000000000
  0.2000000000000000         ! --Weight of each partials (in 11,12,...,22,23..
order);
                 ! **MANDATORY ITEM**: you should provide as many values as each
partial of your system have
 SIGMA-MASTER = 0.02000000    ! --Standard deviation and leading series for scalable
sigma (this or any alternative is) **MANDATORY ITEM**
                 ! alternatives: (..._)SIGMA (standard deviation) or
                 ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
 USE-RFACTOR = 0.00000000     ! --Whether to use R-factor (1) instead of normal chi2
for this set (default: 0)
```

```
 RENORM = 1       ! --Whether to vary amplitudes/renormalize (default: 0)
 POLY-BACK-FLAGS = 0 0 0 0    ! --Whether to vary polynomial terms: 0th,1st,.. order
(default: 0 for each)

[ EXP ]
 TYPE = ND        ! --Type of the experiment, **MANDATORY ITEM** (it should be EXAFS,
GR, EDIFF, ND, XRD
 DATAFILE = c4.sq ! --File name containing the experimental data, **MANDATORY ITEM**
 Q-RANGE = 0.55000000 9.15000000    ! --Range used for fit (default:
0.55000000..9.15000000, the whole range of the data)
                  ! alternative: POINT-RANGE = 1 100 ! The applied point range to fit
(default: 1..100, the whole range of the data)
 R-SPACING = 0.10000000 ! --R-spacing in A units (default: 0.10000000)
 CONST-SUBTRACT = 0.00000000  ! --Constant to subtract (default: 0)
 PARTIAL-COEFFS =
  0.0200000000000000 0.1000000000000000 0.3000000000000000
  0.0800000000000000 0.3500000000000000
  0.1500000000000000          ! --Weight of each partials (in 11,12,...,22,23..
order);
                  ! **MANDATORY ITEM**: you should provide as many values as each
partial of your system have
 SIGMA = 0.01000000     ! --Standard deviation (this or any alternative is)
**MANDATORY ITEM**
                  ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma
                  ! of the leading series' starting chi2
 USE-RFACTOR = 0.00000000     ! --Whether to use R-factor (1) instead of normal chi2
for this set (default: 0)
 RENORM = 1       ! --Whether to vary amplitudes/renormalize (default: 0)
 POLY-BACK-FLAGS = 1 1 1 0    ! --Whether to vary polynomial terms: 0th,1st,.. order
(default: 0 for each)

[ EXP ]
 TYPE = XRD             ! --Type of the experiment, **MANDATORY ITEM** (it should be
EXAFS, GR, EDIFF, ND, XRD
 DATAFILE = test_alpha0.03_a1.2_b0.1_nof.iq     ! --File name containing the
experimental data, **MANDATORY ITEM**
 Q-RANGE = 1.75000000 22.75000000   ! --Range used for fit (default:
0.60000000..22.75000000, the whole range of the data)
                  ! alternative: POINT-RANGE = 24 444 ! The applied point range to
fit (default: 1..444, the whole range of the data)
 R-SPACING = 0.10000000 ! --R-spacing in A units (default: 0.10000000)
 CONST-SUBTRACT = 0.00000000  ! --Constant to subtract (default: 0)
 SIGMA = 10.00000000    ! --Standard deviation (this or any alternative is)
**MANDATORY ITEM**
                  ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma
                  ! of the leading series' starting chi2
 USE-RFACTOR = 0.00000000     ! --Whether to use R-factor (1) instead of normal chi2
for this set (default: 0)
 READ-COEFFS = 0  ! --Whether to read the X-ray coefficients based on the Waasmaier-
Kirfel table (default: 0)
 I(Q)_A_B_ALPHA = 1 1 0 0     ! --Whether to fit I(Q) instead of structure factor
(default: 0)
                  ! If custom values for a, b or alpha are needed, should be supplied
here. If only b and alpha are needed, give something for a as well!
 COMPTON = 0      ! --Whether to use Compton contribution during I(Q) calculation
(default: 1)
 I(Q)BACKG_MU_DMU-MAX = 0     ! --Whether to use background correction in case of
I(Q) data fit (default: 0)
                  ! If custom values mu and dmu_max are needed, should be supplied
here. If only dmu_max is needed, mu has to be given as well!
```

```
                  ! Default values are for mu: 0.70000000 dmu_max: 0.20000000
 AXS-INDEX_FPRIME_FRACTION = 1 -0.00001000 0.00200000 ! --Index of the atom type to
use to use AXS for, f' and fraction
                  ! to vary f' between f'-f'*fraction->f'+f'*fraction (default: 0,
not to use it)
 FPRIME-INDEX_FPRIME = 0      ! --Index of the atom type to use f' for and f'
(default: 0, not to use it)
 RENORM = 1       ! --Whether to vary amplitudes/renormalize (default: 0)
 POLY-BACK-FLAGS = 1 0 0 0    ! --Whether to vary polynomial terms: 0th,1st,.. order
(default: 0 for each)
 COMPTON-SUPR-COEFF = 0 ! --Whether to fit alpha parameter in I(Q) (default: 0)

[ EXP ]
 TYPE = EXAFS           ! --Type of the experiment, **MANDATORY ITEM** (it should be
EXAFS, GR, EDIFF, ND, XRD
 DATAFILE = asse_15i_as_bft_k0.dat  ! --File name containing the experimental data,
**MANDATORY ITEM**
 K-RANGE = 1.15000000 13.85000000   !  --Original  range  used  for  fit  (default:
0.60000000..22.75000000, the whole range of the data)
                  ! alternative: POINT-RANGE = 12 266 ! The original point range to
fit (default: 1..444, the whole range of the data)
                  ! actual range used due to E0 shit interpolation data point loss:
22 265
 R-SPACING = 0.10000000 ! --R-spacing in A units (default: 0.10000000)
 SIGMA = 0.00000800     !  --Standard  deviation  (this  or  any  alternative  is)
**MANDATORY ITEM**
                  ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                  !  (..._)SIGMA-SCALABLE  (sets  starting  chi2  of  this  set  or
constraint to the proportion given by this scalable sigma
                  ! of the leading series' starting chi2
 USE-RFACTOR = 0.00000000     ! --Whether to use R-factor (1) instead of normal chi2
for this set (default: 0)
 RENORM = 1       ! --Whether to vary amplitudes/renormalize (default: 0)
 POLY-BACK-FLAGS = 1 0 0 0    ! --Whether to vary polynomial terms: 0th,1st,.. order
(default: 0 for each)
 ABSORBER-TYPE = 1      ! --Type of absorbing atom, **MANDATORY ITEM**
 CHIK-POWER = 3         ! --Power of chi(k), (default: 3)
 BACKSCATT-FILE = as_edge_bs.dat    ! --File name containing the E(k,r) coefficients,
**MANDATORY ITEM**
 R-RANGE =
 2.20000000 3.20000000
 2.20000000 3.20000000
 2.20000000 3.20000000        ! --R-range in A (centre of bins) for the coeffs to use
in the fit for each partials (default: the whole r range of data -- for the 1st
partial: 0.20000000..3.50000000);
                  ! when you provide less values than required, the last value passed
to the remaining ones
                  ! alternative (example for the first partial only): R-POINT-RANGE =
21 31 ! The applied point range to fit (default: 1..34, the whole range of the data)
 DELTAE0_NGRID = 5.00000000 10            ! --Maximum E0 shift in eV, number of grid
points in one direction, (default: 0.00000000, 0, no shift is performed).

[ EXP ]
 TYPE = EDIFF           ! --Type of the experiment, **MANDATORY ITEM** (it should be
EXAFS, GR, EDIFF, ND, XRD
 DATAFILE = BMtMe_quasiediff.fed    ! --File name containing the experimental data,
**MANDATORY ITEM**
 G-RANGE = 0.175000 16.000000 ! --Range used for fit (default: 0.175000..16.000000,
the whole range of the data)
                  ! alternative: POINT-RANGE = 1 634 ! The applied point range to fit
(default: 1..634, the whole range of the data)
 R-SPACING = 0.100000   ! --R-spacing in A units (default: 0.100000)
 CONST-SUBTRACT = 0.000000    ! --Constant to subtract (default: 0)
 SIGMA-MASTER = 0.000100      ! --Standard deviation and leading series for scalable
sigma (this or any alternative is) **MANDATORY ITEM**
```

```
                  ! alternatives: (..._)SIGMA (standard deviation) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
 USE-RFACTOR = 0.000000 ! --Whether to use R-factor (1) instead of normal chi2 for
this set (default: 0)
 RENORM = 0       ! --Whether to vary amplitudes/renormalize (default: 0)
 POLY-BACK-FLAGS = 0 0 0 0    ! --Whether to vary polynomial terms: 0th,1st,.. order
(default: 0 for each)
```

[COS]

```
 DISTRIB-TYPE = GAUSSIAN           ! --Target distribution: GAUSSIAN(default) or
UNIFORM or ABSENT or filename(to be read from)
 DISTRIB-DEGREES = 120             ! --Mean angle in degrees, **MANDATORY ITEM**
 DISTRIB-WIDTH = 0.17              ! --Width parameter, **MANDATORY ITEM**,
interpreted as
                                   ! half-width in angles for UNIFORM and sigma in cosine
for GAUSSIAN and ABSENT
 DCOSTH = 0.2                      ! --Spacing in cos(theta) space, (default: 0.05).
It should be the same for all COS constraints (except in the case of file).
 CENT-TYPE = 3                     ! --Central atom type, **MANDATORY ITEM**
 NEIGH-TYPE_FROM_TO = 1 2.3 3      ! --Neighbour type, rmin and rmax (in A units),
**MANDATORY ITEM**
 NEIGH-TYPE_FROM_TO = 2 3 3.8      ! --Neighbour type, rmin and rmax (in A units),
**MANDATORY ITEM**
 SIGMA = 0.0001         ! --Standard deviation (this or any alternative is)
**MANDATORY ITEM**
                  ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
```

[COORD]

```
 CENT-TYPE = 2                     ! --Central atom type, **MANDATORY ITEM**
 NEIGH-TYPE_FROM_TO = 1 2.1 2.7          ! --Neighbour type, rmin and rmax (in A
units), **MANDATORY ITEM**
 COORDNUM_FRACT_SIGMA = 2 0.4 0.00015    ! --Desired average coordination number,
fraction, standard deviation (this or any alternative is) **MANDATORY ITEM**
                  ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
 COORDNUM_FRACT_SIGMA = 3 0.6 0.00025    ! --Desired average coordination number,
fraction, standard deviation (this or any alternative is) **MANDATORY ITEM**
                  ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
```

[AVCOORD]

```
 CENT-TYPE = 2                     ! --Central atom type, **MANDATORY ITEM**
 NEIGH-TYPE_FROM_TO = 1 2.1 2.7          ! --Neighbour type, rmin and rmax (in A
units), **MANDATORY ITEM**
 COORDNUM_SIGMA = 2 0.036244125          ! --Desired average coordination number,
standard deviation (this or any alternative is) **MANDATORY ITEM**
                  ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
```

[CONC]

```
 PTYPE1_PTYPE2_FROM_TO = 4 3 1.8 2  ! --Primary type 1, primary type 2, minimum and
maximum distance between them, **MANDATORY ITEM**
 STYPE_FROM1_TO1_FROM2_TO2 = 1 1.55 1.6 0.9 1.1 !  --Secondary type, minimum and
maximum distance to primary1, minimum and maximum distance to primary2 (in A units),
**MANDATORY ITEM**
 COORDNUM_FRACT_SIGMA = 1 0.8 1     !  --Desired coordintion number, fraction,
standard deviation (this or any alternative is) **MANDATORY ITEM**
                 ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                 !  (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
```

[SNC]

```
 CTYPE_FTYPE_FROM_TO = 3 1 0.9 1.1  ! --Central type, first neighbour type, minimum
and maximum distance between them, **MANDATORY ITEM**
 STYPE_FROM_TO = 3 0.9 1.1          ! --Second neighbour type, minimum and maximum
distance to first neighbour (in A units), **MANDATORY ITEM**
 STYPE_FROM_TO = 4 0.9 1.1          ! --Second neighbour type, minimum and maximum
distance to first neighbour (in A units), **MANDATORY ITEM**
 COORDNUM_FRACT_SIGMA = 1 1 1 !  --Desired coordintion number, fraction, standard
deviation (this or any alternative is) **MANDATORY ITEM**
                 ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                 !  (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
```

[SWAP]

```
 FRACT = 0.1                        ! --Fraction of swap moves, **MANDATORY ITEM**
 PAIRS = 1-2                        !  --Atom pairs to swap, **MANDATORY ITEM**,
syntax: list of keyword values (consecutively or on multiple lines without keywords):
                              ! ALL: all type of atoms to swap; Type-ANY: all pairs
containing atom type {Type} are swapped; Type1-Type2: only pairs with atom types
Type1 and Type can be swapped
```

[LOCINV]

```
 LOC-MODE = DISTANCE-BASED          ! --Local invariance calculation mode, can be
BIN-BASED or DISTANCE-BASED (default)
 LOC-R-SPACING = 0.05         ! --Local invariance histogram bin size in A (default:
first normal histogram bin size)
 LOC-INT-FROM_SIGMA-MASTER =  0 6.3275902       !  --Local invariance interval
beginning, BIN-BASED method: minimum reduced distance for local invariance
calculation,
                 ! (DISTANCE-BASED method: the fraction of neighbour atoms for the
local invariance calculation at the beginning of the current interval);
                 ! standard deviation and leading series for scalable sigma (this or
any alternative is) **MANDATORY ITEM**
                 ! alternatives: (..._)SIGMA (standard deviation) or
                 !  (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
 LOC-INT-FROM_SIGMA =  0.03 0.2          !  --Local invariance interval beginning,
BIN-BASED method: minimum reduced distance for local invariance calculation,
                 ! (DISTANCE-BASED method: the fraction of neighbour atoms for the
local invariance calculation at the beginning of the current interval);
                 ! standard deviation (this or any alternative is) **MANDATORY
ITEM**
                 ! alternatives: (..._)SIGMA-MASTER (standard deviation and leading
series for scalable sigma) or
                 !  (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
 LOC-INT-FROM_SIGMA =  0.05 0.25         !  --Local invariance interval beginning,
BIN-BASED method: minimum reduced distance for local invariance calculation,
```

! (DISTANCE-BASED method: the fraction of neighbour atoms for the local invariance calculation at the beginning of the current interval);

! standard deviation (this or any alternative is) **MANDATORY ITEM**

! alternatives: (..._)SIGMA-MASTER (standard deviation and leading series for scalable sigma) or

! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or constraint to the proportion given by this scalable sigma of the leading series' starting chi2

LOC-INT-TO = 0.06 ! --Local invariance interval end, BIN-BASED method: minimum reduced distance for local invariance calculation,

! (DISTANCE-BASED method: the fraction of neighbour atoms for the local invariance calculation at the end of the final interval)

[NBPOT]

NB-TYPE = LJ ! -- Number of GROMACS types (default: number of RMC types),

NGR-TYPES = 3 ! -- Number of GROMACS types (default: number of RMC types),

CUTOFF-VDW_COUL = 8 9 ! -- VdW and Coulomb cut-offs in A, (default: 1, 1), in reduced units, corresponding here to 16.806143 A

! if only the vdW cut-off is provided, the same value will be used for the Coulomb term, as well.

! if only the factor for vdW provided, the same value will be applied for Coulomb term, as well.

LJ-SIG-AMEAN = 3.5 3.3 3 ! -- (combination rule = 2): LJ sigma parameters for each GROMACS types [in A], values for the partials calculated by arithmetic mean = , **MANDATORY ITEM**

! alternatives: LJ-C6-EACH (combination rule = 0): LJ first potential parameters [in kJ*A^6/mol] for each GROMACS partials

! or LJ-C6-GMEAN (combination rule = 1): C6 coefficient [in kJ*A^6/mol] for each GROMACS types, values for the partials calculated by geometric mean

! or LJ-SIG-GMEAN (combination rule = 3): LJ sigma parameters for each GROMACS types [in A], values for the partials calculated by geometric mean

LJ-EPS-GMEAN = 0.3 0.2 0.25 ! -- (combination rule = 2): LJ epsilon parameters for each GROMACS types [in kJ/mol], values for the partials calculated by geometric mean = , **MANDATORY ITEM**

! alternatives: LJ-CPOW-EACH (combination rule = 0): LJ second potential parameters [in kJ/mol·A^{12}] for each GROMACS partials

! or LJ-CPOW-GMEAN (combination rule = 1): CN coefficient [in kJ/mol·A^{12}] for each GROMACS types, values for the partials calculated by geometric mean

! or LJ-EPS-GMEAN (combination rule = 3): LJ epsilon parameters for each GROMACS types [in kJ/mol], values for the partials calculated by geometric mean

SIGMA-OVERALLPOT = m0.1 ! --Weighing parameter for all the non-bonded potential terms, no prefix: standard deviation; 'm': leading potential series index; 's': scalable sigma, **MANDATORY ITEM**

! alternatives: standard deviation(s) for VdW (VDW-SIGMA: mandatory) and Coulomb (COUL-SIGMA: optional, if not given, then VDW-SIGMA values(s) are used

CHI-LOWLIMIT-RATIO = NONE ! -- The chi^2 belonging to the potential term does not decrease below the limit given by this ratio of the initial

! chi^2 of the potential term (default: NONE) -- this effects only, if non-bonded interactions are present

LJ-POW = 12 ! --Power of the repulsion term in 6-{POW} LJ (default: 12)

1-4-SCALE-VDW_COUL = 0.5 0 ! --Factor(s) for scaling the 1-4 VdW and Coulombic parameters (default: 1, 1),

! if only the factor for vdW provided, the same value will be applied for Coulomb term, as well.

[NBPOT]

```
 NB-TYPE = TABULATED                        ! -- Number of GROMACS types (default:
number of RMC types),
 TEMPERATURE = 273                          ! --Temperature in K, no algorithmic
meaning, for tabulated potential (default: 298),
 CUTOFF-TAB = 10                      ! --Cut-off for the tabulated potential(s) in A,
(default: 1), in reduced units, corresponding here to 16.806143 A
 IND_SIGMA_FILE = 2 0.1 As-Se.tabpot      ! --Index of the partial for this tabulated
potential, standard deviation (this or any alternative is) **MANDATORY ITEM** file
name for the tabulated potential
                             ! alternatives: (..._)SIGMA-MASTER (standard deviation
and leading series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
 IND_SIGMA-MASTER_FILE = 4 ms0.5 Se-Se.tabpot   ! --Index of the partial for this
tabulated potential, standard deviation and leading series for scalable sigma (this
or any alternative is) **MANDATORY ITEM** file name for the tabulated potential
                             ! alternatives: (..._)SIGMA (standard deviation) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2
 IND_SIGMA_FILE = 5 1 Se-I.tabpot   ! --Index of the partial for this tabulated
potential, standard deviation (this or any alternative is) **MANDATORY ITEM** file
name for the tabulated potential
                             ! alternatives: (..._)SIGMA-MASTER (standard deviation
and leading series for scalable sigma) or
                  ! (..._)SIGMA-SCALABLE (sets starting chi2 of this set or
constraint to the proportion given by this scalable sigma of the leading series'
starting chi2

[ CUSTMOVE ]
 NMOVED-ATOMS = 5        ! --Number of atoms moved in a single move, **MANDATORY
ITEM**
                         ! The move parameters should be listed after each CUSTOM-LINE
keywords line by line
 CUSTOM-LINE = 0.5             !normal translation amplitude (Angstrom)
 CUSTOM-LINE = 5         !rotation amplitude (degrees)
 CUSTOM-LINE = 0.01            !max individual move amplitude for carbon (angstroms)
 CUSTOM-LINE = 0.01            !max individual move amplitude for chlorine (angstroms)
 CUSTOM-LINE = 3.00            !safety distance (in Angstrom)

[ BPOT ]
 COMP-OPTION = _FF_OPLS              ! --Compiler option for bonded potential for all
of the topology file(s)
 COMP-OPTION = ORI_BOND              ! --Compiler option for bonded potential for all
of the topology file(s)
 COMP-OPTION = ORI_ANGLE             ! --Compiler option for bonded potential for all
of the topology file(s)
 COMP-OPTION = ORI_DIH         ! --Compiler option for bonded potential for all of the
topology file(s)

[ VIBRAMP ]
 GAUSSIAN-SIGMA = 0.05 0.07 0.06          ! --Sigma parameters in A controlling the
width of the Gaussian distribution for each atom type, **MANDATORY ITEM**

[ END ]
```

*D. The structure of the *.cfg file*

The current format is version 3 of the text-type coordinate file, which changed a bit from version 2.0, but the old format can be read as well. The change only effected the 5th line of the file, it will contain all kind of the existing generated, tried and accepted moves handled by the program regardless whether they are involved in the actual simulation or not, after them their names. The structure of the **.bcf* file has changed similarly.

First a header can be found with general information. The coordinates of the atoms are arranged according to types, first are all the coordinates of the first type, then the second and so on. It has to be noted that regardless the three separate box vectors, which represent the half length of the simulation box in Ångstrom, only cubic simulation box can be handled, and only the first box vector is read.

If there are virtual sites, then in TEST_MODE a version 4 format **.cfg* file is created, which includes the virtual particles as well. In this case the 8th line will contain the number of virtual sites after *ntotal*, and in line 9 the number of virtual types after the number of normal types. In this case there are blocks with the number of virtual sites for each type similarly to the one describing the normal types, and the coordinates of the virtual sites follow the coordinates of the atoms. The information regarding the virtual types is only for display, as all the relevant information concerning the virtual sites for the run comes from the topology.

```
(Version 3 format configuration file) !file created by SimpleCfg::save !
BMtMe_(with FNC)_only_for_checking


            5000             4584              4584                0                0
               0                0                 0                0                0
               0                  0                  moves generated            tried
        accepted    potential-acc          swap-gen          swap-acc      E0shift-gen
     E0shift-acc  I(Q)mucorr-gen  I(Q)mucorr-acc     f'shift-gen     f'shift-acc
            0 configurations saved

       1300 molecules of all types
          3 types of molecules
          1 is the largest number of atoms in a molecule
          0 Euler angles are provided

          F (box is cubic)
            Defining vectors are:
             30.450200   0.000000   0.000000
              0.000000  30.450200   0.000000
              0.000000   0.000000  30.450200

        300 molecules of type  1
          1 atomic sites
              0.000000   0.000000   0.000000

        200 molecules of type  2
          1 atomic sites
              0.000000   0.000000   0.000000

        800 molecules of type  3
          1 atomic sites
              0.000000   0.000000   0.000000

 -0.401797913580607  -0.456779264844952  0.595887013035009
 -0.880737716836848  0.494151149455218  -0.656959975967055
 -0.554530511055525  -0.353245955004950  0.885121211703789
 0.406319420672594  0.389833805383036  -0.855405069648957
 0.919944814288837  0.762568608721879  -0.185056930847424
 0.573462973900463  0.034026607847307  0.202465206711665
 -0.746369290392583  0.178957135325712  -0.948424915243704
 -0.862191047832178  -0.514506207176586  0.355296772334447
 0.408792067729221  -0.001200463338156  0.229725315721684
 0.432388851173119  0.878801946366788  0.200428054316574
 -0.116400037693675  -0.860609330853663  -0.947427893912050
•
•
•
 -0.550081821216744  -0.851298541754620  -0.924841685076493
 -0.709210176450055  -0.991277379975797  0.994665938293646
```

```
 -0.819921420524851  -0.922772485395741 -0.989073934225886
 -0.790493404801405  -0.970241507519512  0.990513018228407
```

E. The structure of the experimental data files

The multiple data entry in a line can be separated by as many spaces and or tabulators, as you like. The data does not have to be equidistant, the *r, Q, k* values are read from the files. Only in case of the EXAFS coefficient file for historical reasons the numerical values are **separated by comma**!

E.1. *g(r)* data

First the number of data points has to be given. The title of the data series can be given following this in the first line, or in the second line, anyway it is not read. The data will start at the 3rd line, the *r* and the *g(r)* data has to be given, for one data point /line.

```
305         ! g(r) data

0.05  0
0.15  0
    •
    •
    •
30.35 1.00031
30.45 1.00016
```

E.2. Neutron scattering, *S(Q)* data

First the number of data points has to be given. The title of the data series can be given following this in the first line, or in the second line, anyway it is not read. The data will start at the 3rd line, the *Q* and *S(Q)* data has to be given, for one data point /line.

```
290         !S(Q) data

0.001000000 1.726670E-08
0.001050000 1.717757E-08
    •
    •
    •
0.485000000 1.539494E-11
0.495000000 2.366464E-11
```

E.3. X-ray scattering, *F(Q)* and *I(Q)* data,

In the first line the number of data points has to be given. The title of the data series can be given following this in the first line, or in the second line, anyway it is not read.

The data will start at the 3rd line, the compulsory two columns are the *Q* and *F(Q)* or in case of *I(Q)* fitting the *Q* and *I(Q)* data.

Using the old format **.dat* file only *F*(Q) can be fitted, and the *ntypes*(ntypes+1)* columns of the normalized coefficients should follow the *F(Q)* values.

From RMC version 2.0 if the free format **.dat* file is used there is a possibility to calculate the atomic scattering factors and from them the coefficients by the program. By default the 2.0-2.1 versions calculated the coefficients, causing problems with downward compatibility. From RMC version 2.2 if the **.dat* file version #002 is used, then the default (if no READ-COEFFS is given) is to read the coefficients for downward compatibility, but in case of the newly introduced version #003 **.dat* file the coefficients are calculated by default (see II.C). In both cases the READ-COEFFS keyword can regulate the behaviour, 0: calculating, 1: reading. The key word can be given in the [EXP] section of the data set in question.

In case of reading for *F(Q)* fitting the *ntypes*(ntypes+1)* columns normalized coefficients in RMC order should follow the *F(Q)* as before.

If *I(Q)* is fitted and the coefficients should be read, then *ntypes* columns should contain the not normalized atomic scattering factors, *f(Q)* in RMC order.

If scalable background correction is used for the *I(Q)* fitting, then another column, *IB(Q)* containing the original background should be supplied as well. This should be the last column in the line, either following the *I(Q)* if the coefficients are calculated or the last column of *f(Q)* if they are read.

This is an example for the normal *F(Q)* fitting with the coefficients for a three component system:

```
444
! xrd data
0.6   -1.13286    0.14809     0.30537     0.16809     0.15743     0.17332     0.0477
0.65  -1.10374    0.14807     0.30536     0.1681      0.15744     0.17333     0.04771
•
•
•
22.7  -0.00617    0.13798     0.28402     0.18294     0.14615     0.18828     0.06063
22.75 -0.00645    0.13803     0.28415     0.18284     0.14624     0.18819     0.06055
```

This is an example for *I(Q)* fitting with background correction and the coefficients are calculated:

```
444
format: Q, I(Q), IB(Q)
0.6   294.0007921167610       0.0010
0.65  200.0437465786670       0.0011
    •
    •
    •
22.7   43.9352457507714       0.0452
22.75  43.5924530628035       0.0453
```

E.4. Electron diffraction, *F(g)* data

In the first line the number of data points has to be given. The title of the data series can be given following this in the first line, or in the second line, anyway it is not read.

The data will start at the 3rd line, the *g* the *F(g)* followed by the *ntypes*(ntypes+1)* columns of the normalized coefficients.

The following example is for a 3-component system:

```
100
!electron diffraction data
1.75E+00    -3.88E-01   -6.39E-01   -4.54E-01   -1.38E-01   -7.91E-02   3.38E-01
1.80E+00    -2.19E-01   -7.56E-01   -4.31E-01   1.69E-01    2.70E-01    2.52E-01
    •
    •
    •
6.95E+00    -3.59E-02   -1.08E-01   4.97E-02    3.52E-02    3.79E-02    -2.52E-02
7.00E+00    -1.90E-02   -9.63E-02   3.22E-02    1.66E-02    2.44E-02    -4.51E-03
```

E.5. EXAFS data

Two files are needed for each *EXAFS* data series, the experimental $\chi(k)$ data file, and the coefficient file.

The structure of the $\chi(k)$ data file is very similar to the *g(r)* or *S(Q)* data file, first the number of data points has to be given. The title of the data series can be given following this in the first line, or in the second line, anyway it is not read. The data will start at the 3rd line, the *k* and $\chi(k)$ data has to be given, for one data point /line. The *E(k)* data, which is actually used as the experimental data during the fitting is calculated based on the used data range given in the **.dat* file, and this is written in to the **.expt* file, (if free format **.dat* is used, only if WRITE-EXAFS-COEFFS=1 keyword is given).

```
266
AsSeI, As edge
```

```
0.6         0
0.65        0
 •
 •
 •
13.85       -6.4922E-5
13.9        -1.0510E-4
```

In a coefficient file the coefficient for the Fourier transformation from the histogram to the *E(k)* data is given, (see I.F) for details. The file has to contain as many blocks (given here as an example) following each other without empty lines separating them as the number of RMC types. The first header line contains the number of RMC types (actually the number of partials the edge particle is involved in), then **separated by a comma** the number of *k* data points for a block. The block should be repeated for each edge type-*itype* (*i*=0, *ntypes*) begins here. The next line should contain the number of the *r* columns and then the name of the partial. If number of *r* columns is given, then the program can check, whether there is enough data in the line which should be used according to line 34 in the **.dat* file. If it is not given, then the program will still continue, but no checking can be performed. The *r* columns correspond to the histogram bins of the data set determined by the general bin shift and the data set's own *r* spacing, has to contain definitely the first part of the histogram bins up to the last used bin/*r* column to be used for the fitting, but no more is necessary. Then has to follow as many line as the number of *k* points, containing the coefficients **separated by comma**. Be aware, that no *k* value has to be specified in the beginning of each line, the first value is the first *c(r,k)*!

It has to be noted, that not all the *r* columns has to be used during the fitting. Either the index range of the used r columns has be specified (for old format **.dat* file only this can be given), or the first and last r value to be used. In this later case the used columns are determined based on the *r* spacing value of the data set and the general bin shift. Sometimes instead of the first *c(r,k)* the *k* value is given at the beginning of each row for easier navigation in the file, but this does not matter, as normally this column is not used during the fitting, and anyhow there is no count in the corresponding histogram bin…

```
3,444
34 As-As
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.07098, 0.17069, 0.16248,
0.16508, 0.1661, 0.17524, 0.17698, 0.19435, 0.22409, 0.15616, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.07542, 0.14545, 0.15335,
0.15229, 0.14975, 0.15474, 0.15183, 0.16232, 0.18129, 0.2682, 0, 0, 0, 0
 •
 •
 •
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.08905, 0.13954, 0.14374,
0.13931, 0.13355, 0.13474, 0.12779, 0.13216, 0.14169, 0.20472, 0, 0, 0, 0
34 As-Se
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.06068, 0.15069, 0.16248,
0.16508, 0.1661, 0.17524, 0.17698, 0.19435, 0.22409, 0.33616, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.07542, 0.14545, 0.15335,
0.15229, 0.14975, 0.15474, 0.15183, 0.16232, 0.18129, 0.2682, 0, 0, 0, 0
 •
 •
 •
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.08905, 0.13954, 0.14374,
0.13931, 0.13355, 0.13474, 0.12779, 0.13216, 0.14169, 0.20472, 0, 0, 0, 0
34 As-I
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.06068, 0.15069, 0.16248,
0.16508, 0.1661, 0.17524, 0.17698, 0.19435, 0.22409, 0.89616, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.07542, 0.14545, 0.15335,
0.15229, 0.14975, 0.15474, 0.15183, 0.18232, 0.18129, 0.9282, 0, 0, 0, 0
 •
 •
 •
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.08905, 0.13954, 0.14374,
0.13931, 0.13355, 0.13474, 0.12779, 0.13216, 0.14169, 0.20472, 0, 0, 0, 0
```

F. Experimental cosine distribution of bond angles data

In case of calculation method 3 for the cosine distribution of bond angles instead of the angle and *wcontrol* parameter the name of the experimental distribution file has to be given in the **.dat* file. The structure of the cosine distribution file is very similar to the *g(r)* or *S(Q)* data file, the first line contains the number of data points, the second can contain some text or be empty, and from the third line should be given a cos(θ) and a distribution value. The cos(theta) values have to be equidistant, and denote the middle of the interval, and the total –1- → +1 interval has to be divisible with the *dcos(θ)* value (in the example 0.05), and the distribution has to be given for the whole interval. The distribution does not have to be normalized, this is done if necessary by RMC_POT.
An example is:

```
40

-0.975      0.00003511
-0.925      0.0001345
-0.875      0.0004726
-0.825      0.00152295
•
•
•
0.875 0.00000001
0.925 0
0.975 0
```

G. The structure of the *.fnc file

The **.fnc* file is needed, if fnc option 1-3 is used. It has to be compatible with the **.cfg* file. The number of different FNC constraints has to be given in line 4. In line 5 the minimum distances in line 6 the maximum distances for the FNC constraint types has to be given. In line 8 the number of atoms has to follow. From line 10 begins the 3 lines blocks for each atom. The first of it contains the index of the atom, then the number of FNC neighbours. The next line contains the indices of the neighbours, and the following line the indices of the constraint types. Even if there is no constraint for an atom, a line with the atom index and 0 for the number of neighbours has to be given, but the second and the third line will be missing.

```
Fixed neighbours constraints (.FNC) file for XYZ3

 No. of possible rmin-rmax pairs:
  4
  2.06999993  1.01999998  2.54999995  1.66999996
  2.19000006  1.13999999  2.78999996  1.90999997

 10000

 1 4
 2001 4001 4002 4003
 1 2 2 2
 2 4
 2002 4004 4005 4006
 1 2 2 2
 3 4
 2003 4007 4008 4009
 1 2 2 2
 4 0
•
•
•
 9999 4
 2000 4000 9998 10000
 2 3 4 4
 10000 4
```

```
 2000 4000 9998 9999
 2 3 4 4
```

H. The structure of the topology file

As a simple example first the topology file of the spce water containing 2000 molecules will be given:

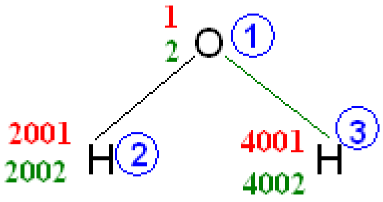
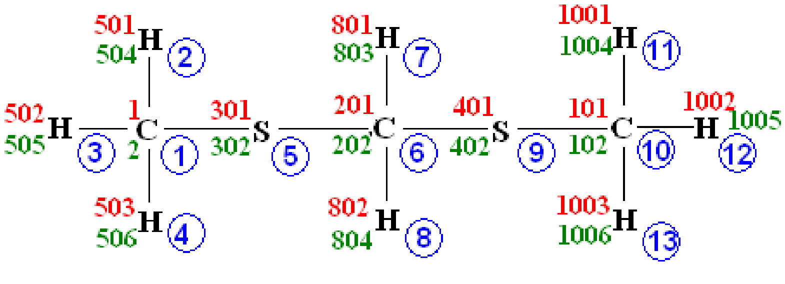


Figure 11: Numbering of the atoms for water: blue circled numbers are the GROMACS atom indices; red numbers are the RMC indices in the first molecule, green in the second molecule.

```
#include “ffoplsaa.itp”

[ moleculetype ]
; molname          nrexcl
SOL                2

[ atoms ]
;   nr   type       resnr residue  atom   cgnr     charge       mass RMC_index

     1      opls_116   1       SOL      OW       1     -0.8476              ; 1    .2
     2      opls_117   1       SOL     HW1       1      0.4238              ; 2001 2002
     3      opls_117   1       SOL     HW2       1      0.4238              ; 4001 4002

[ bonds ]
; I     j      funct length      force.c. RMC_sigma
1      2      1      0.1         345000      ; 4.5e-3
1      3      1      0.1         345000


[ angles ]
; I     j      k      funct angle       force.c.    RMC_sigma
2      1      3      1      109.47      383         6e-4

[ system ]
; Name
spce_water

[ molecules ]
       SOL                  2000
```

The second example will show the topology and two include topology files with the molecule type definitions of a bis(methylthio)methane and dimethylsulfide system consisting 100 BMtMe and 10 DMeS molecules.



BMtMe

DMeS

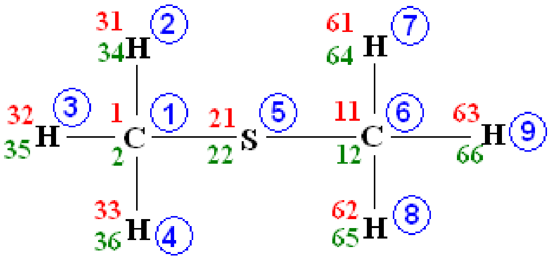


Figure 12: The GROMACS atom numbers (blue circled) and the RMC indices for the first (red) and second (green) molecules of the system.

bis(methyltio)methane.itp

```
[ moleculetype ]
; molname         nrexcl            index_offset for RMC
BMtMe       3

[ atoms ]
;   nr       type          resnr    residue     atom  cgnr    charge   mass   first_i
sec_i
#ifdef _FF_OPLS
     1      opls_209   1      BMtMe    CA       1  -0.013               ; 1     2
#ifdef _DEUTERIUM
     2      opls_140   1      BMtMe   DA1       1   0.06   2.014102     ; 501   504
     3      opls_140   1      BMtMe   DA2       1   0.06   2.014102     ; 502   505
     4      opls_140   1      BMtMe   DA3 1     1   0.06   2.014102     ; 503   506
#else
     2      opls_140   1      BMtMe   HA1       1   0.06                ; 501   504
     3      opls_140   1      BMtMe   HA2       1   0.06                ; 502   505
     4      opls_140   1      BMtMe   HA3       1   0.06                ; 503   506
#endif
     5      opls_202   1      BMtMe    S1       2  -0.335               ; 301   302
     6      opls_210   1      BMtMe    CB       3   0.216               ; 201   202
#ifdef _DEUTERIUM
     7      opls_140   1      BMtMe   DB1       3   0.06   2.014102     ; 801   803
     8      opls_140   1      BMtMe   DB2       3   0.06   2.014102     ; 802   804
#else
     7      opls_140   1      BMtMe   HB1       3   0.06                ; 801   803
     8      opls_140   1      BMtMe   HB2       3   0.06                ; 802   804
#endif
     9      opls_202   1      BMtMe    S2       4  -0.335               ; 401   402
   10 opls_209   1      BMtMe    CC       5  -0.013               ; 101   102
#ifdef _DEUTERIUM
    11      opls_140   1      BMtMe   DC1       5   0.06     2.014102   ; 1001  1004
    12      opls_140   1      BMtMe   DC2       5   0.06     2.014102   ; 1002  1005
    13      opls_140   1      BMtMe   DC3       5   0.06     2.014102   ; 1003  1006
#else
    11      opls_140   1      BMtMe   HC1       5   0.06                ; 1001  1004
    12      opls_140   1      BMtMe   HC2       5   0.06                ; 1002  1005
    13      opls_140   1      BMtMe   HC3       5   0.06                ; 1003  1006
#endif
#endif

[ pairs ]
;I    j     funct
2     6     1
3     6     1
4     6     1
1     7     1
1     8     1
1     9     1
5     10    1
7     10    1
8     10    1
6     11    1
6     12    1
6     13    1

[ bonds ]
#ifdef _ORI_BOND
;original OPLS bonds
; I   j     funct r     k                 RMC_sigma
1     2     1     ;0.109            284512.0    1e-3
```

```
1      3      1      ;0.109           284512.0
1      4      1      ;0.109           284512.0
1      5      1      ;0.181           185769.6    2e-3
5      6      1      ;0.181           185769.6
6      7      1      ;0.109           284512.0
6      8      1      ;0.109           284512.0
6      9      1      ;0.181           185769.6
9      10     1      ;0.181           185769.6
10     11     1      ;0.109           284512.0
10     12     1      ;0.109           284512.0
10     13     1      ;0.109           284512.0
#else
;from Page (2000) J. Phys. Chem. A Vol. 104 p 6672
1      2      1      0.1108           284512.0    ; 3e-3
1      3      1      0.1108           284512.0
1      4      1      0.1108           284512.0
1      5      1      0.1805           185769.6    ; 4e-3
5      6      1      0.1806           185769.6
6      7      1      0.1108           284512.0
6      8      1      0.1108           284512.0
6      9      1      0.1806           185769.6
9      10     1      0.1805           185769.6
10     11     1      0.1108           284512.0
10     12     1      0.1108           284512.0
10     13     1      0.1108           284512.0
#endif


[ angles ]
#ifdef _ORI_ANGLE
;original OPLS angles
; I    j      k      funct theta        k                  RMC_sigma
2      1      3      1      ;107.8            276.144     1.5e-4
2      1      4      1      ;107.8            276.144
3      1      4      1      ;107.8            276.144
7      6      8      1      ;107.8            276.144
11     10     12     1      ;107.8            276.144
11     10     13     1      ;107.8            276.144
12     10     13     1      ;107.8            276.144
2      1      5      1      ;109.5            292.880     1.6e-4
3      1      5      1      ;109.5            292.880
4      1      5      1      ;109.5            292.880
7      6      5      1      ;109.5            292.880
8      6      5      1      ;109.5            292.880
7      6      9      1      ;109.5            292.880
8      6      9      1      ;109.5            292.880
11     10      9     1      ;109.5            292.880
12     10      9     1      ;109.5            292.880
13     10      9     1      ;109.5            292.880
1      5      6      1      ;98.9             518.816     2.1e-4
6      9      10     1      ;98.9             518.816
#else
;angles coming from Page (2000) J. Phys. Chem. A Vol. 104 p 6672
2      1      3      1      108.9             276.144     ; 1.7e-3
2      1      4      1      108.9             276.144
3      1      4      1      108.9             276.144
7      6      8      1      108.9             276.144
11     10     12     1      108.9             276.144
11     10     13     1      108.9             276.144
12     10     13     1      108.9             276.144
2      1      5      1      107.5             292.88      ; 1.8e-3
3      1      5      1      107.5             292.88
4      1      5      1      107.5             292.88
7      6      5      1      107.5             292.88
8      6      5      1      107.5             292.88
7      6      9      1      107.5             292.88
```

```
8     6     9     1     107.5             292.88
11    10    9     1     107.5             292.88
12    10    9     1     107.5             292.88
13    10    9     1     107.5             292.88
1     5     6     1     102.8             518.816     ; 1.9e-3
6     9     10    1     102.8             518.816
#endif
5     6     9     1     115.9             418.400     ; 2e-3

[ dihedrals ]
;ai   aj    ak    al    funct   C0      C1       C2      C3       C4    C5 RMC_sigma
2     1     5     6     3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000 3e-3
3     1     5     6     3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
4     1     5     6     3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
7     6     5     1     3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
8     6     5     1     3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
7     6     9     10    3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
8     6     9     10    3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
6     9     10    11    3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
6     9     10    12    3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
6     9     10    13    3 ; 1.35352  4.06057   0.00000  -5.41410  0.000  0.000
; no opls param for Ct-S-CT-S,the CT-CT-S-CT will be used instead
#ifdef _SMALL_DIH_BAR
; divided by 4 to facilitate free rotation not to let it froze
1     5     6     9     3 0.23535  0.5784375 0.6024975 -1.416285  0.000  0.00 ; 5e-4
5     6     9     10    3 0.23535  0.5784375 0.6024975 -1.416285  0.000  0.00 ; 2.0e-
4
#else
1     5     6     9     3 0.94140  2.31375   2.40999   -5.66514   0.000  0.00 ; 3.2e-
4
5     6     9     10    3 0.94140  2.31375   2.40999   -5.66514   0.000  0.00 ; 3.3e-
4
#endif
```

Include topology file for DMeS, *`dimethylsulfide.itp`*. The index offset 1300 shows that there are 1300 other atoms (100*13 atoms of the BMtMe), so therefore the first index in the RMC configuration will begin with 1301. If the number of molecules and therefore atoms for the component(s) preceding the DMeS in the system description, then only the index offset has to be changed and not all the first and second RMC indices.

```
[ moleculetype ]
; molname         nrexcl            index_offset
DMeS        3     ;           1300

[ atoms ]
;   nr       type   resnr  residue  atom     cgnr    charge  mass.first_ind
second_ind
#ifdef _FF_OPLS
     1      opls_209   1      DMeS      CA       1    -0.013             ;     1     2

#ifdef _DEUTERIUM
     2      opls_140   1      DMeS     DA1       1     0.06    2.014102  ;     31
      34
     3      opls_140   1      DMeS     DA2       1     0.06    2.014102  ;     32
      35
     4      opls_140   1      DMeS     DA3       1     0.06    2.014102  ;     33
      36
#else
     2      opls_140   1      DMeS     HA1       1     0.06              ;     31
      34
     3      opls_140   1      DMeS     HA2       1     0.06              ;     32
      35
     4      opls_140   1      DMeS     HA3       1     0.06              ;     33
      36
```

```
#endif
     5      opls_202   1      DMeS       S       2    -0.334             ;     21
      22
     6      opls_209   1      DMeS      CB       3    -0.013             ;     11
      12

#ifdef _DEUTERIUM
     7      opls_140   1      DMeS     DB1       3     0.06     2.014102 ;     61
      64
     8      opls_140   1      DMeS     DB2       3     0.06     2.014102 ;     62
      65
     9      opls_140   1      DMeS     DB3       3     0.06     2.014102 ;     63
      66
#else
     7      opls_140   1      DMeS     HB1       3     0.06              ;     61
      64
     8      opls_140   1      DMeS     HB2       3     0.06              ;     62
      65
     9      opls_140   1      DMeS     HB3       3     0.06              ;     63
      66
#endif
#endif

[ pairs ]
;I    j     funct
2     6     1
3     6     1
4     6     1
1     7     1
1     8     1
1     9     1

[ bonds ]
#ifdef _ORI_BOND
;original OPLS bonds
; I   j     funct length            force.c           RMC_sigma
1     2     1     ;0.109            284512.0    1.4e-3
1     3     1     ;0.109            284512.0
1     4     1     ;0.109            284512.0
6     7     1     ;0.109            284512.0
6     8     1     ;0.109            284512.0
6     9     1     ;0.109            284512.0
1     5     1     ;0.181            185769.6    1.2e-3
5     6     1     ;0.181            185769.6
#else
;reset according to T. Iilija, S. Tsuchiya, M. Kimura: Bull. of Chem . Soc of Japan
Vol. 50, No.10 (1977) p2564
; I   j     funct length            force.c.          RMC_sigma
1     2     1     0.1116            284512.0    ;1.5e-4
1     3     1     0.1116            284512.0
1     4     1     0.1116            284512.0
6     7     1     0.1116            284512.0
6     8     1     0.1116            284512.0
6     9     1     0.1116            284512.0
1     5     1     0.1807            185769.6    ;2.3e-4
5     6     1     0.1807            185769.6
#endif


[ angles ]
#ifdef _ORI_ANGLE
;original OPLS angles
; I   j     k     funct angle       force c.          RMC_sigma
2     1     3     1     ;107.8            276.144     5.1e-4
2     1     4     1     ;107.8            276.144
3     1     4     1     ;107.8            276.144
```

```
7     6     8     1     ;107.8            276.144
7     6     9     1     ;107.8            276.144
8     6     9     1     ;107.8            276.144
2     1     5     1     ;109.5            292.880     2.3e-4
3     1     5     1     ;109.5            292.880
4     1     5     1     ;109.5            292.880
7     6     5     1     ;109.5            292.880
8     6     5     1     ;109.5            292.880
9     6     5     1     ;109.5            292.880
1     5     6     1     ;98.9       518.816     1.8e-4
#else
;reset according to T. Iilija, S. Tsuchiya, M. Kimura: Bull. of Chem . Soc of Japan
Vol. 50, No.10 (1977) p2564
; I   j     k     funct angle       force.c.          RMC_sigma
2     1     3     1     109.3       276.144     ; 1.1e-3
2     1     4     1     109.3       276.144
3     1     4     1     109.3       276.144
7     6     8     1     109.3       276.144
7     6     9     1     109.3       276.144
8     6     9     1     109.3       276.144
2     1     5     1     109.3       292.880      ; 7.1e-3
3     1     5     1     109.3       292.880
4     1     5     1     109.3       292.880
7     6     5     1     109.3       292.880
8     6     5     1     109.3       292.880
9     6     5     1     109.3       292.880
1     5     6     1     99.05       518.816     ; 9.1e-3
#endif

[ dihedrals ]
;ai   aj    ak    al    funct   C0     C1       C2          C3    C4      C5
RMC_sigma
2     1     5     6     3 ; 1.35352  4.06057   0.0000  -5.41410   0.0000  0.000 2.6e-
3
3     1     5     6     3 ; 1.35352  4.06057   0.0000  -5.41410   0.0000  0.000
4     1     5     6     3 ; 1.35352  4.06057   0.0000  -5.41410   0.0000  0.000
1     5     6     7     3 ; 1.35352  4.06057   0.0000  -5.41410   0.0000  0.000
1     5     6     8     3 ; 1.35352  4.06057   0.0000  -5.41410   0.0000  0.000
1     5     6     9     3 ; 1.35352  4.06057   0.0000  -5.41410   0.0000  0.000
```

The topology file: *BMtMe_DMeS.top:*

```
#include “ffoplsaa.itp”

#include “bis(methylthio)methane.itp”
#include “dimethylsulfide.itp”

[ system ]
; Name
BMtMe-DMeS

[ molecules ]
   BMtMe             100
   DMeS               10
```

The include topology file contains conditional parts, so it depends on whether the arguments like _DEUTERIUM or _ORI_BOND are passed to the processing of the topology file, which part of the topology is active. In case of the DMeS bond declaration, in the _ORI_BOND section the r and k values are after the semicolon, which means that GROMACS will ignore them and use the values given in the force field files (which actually are the same as given here) only RMC uses them, but in the other part based on the work of Iilija both GROMACS and RMC are using the values given here (no semicolon).

At the beginning of the topology file the #include “ffoplsaa.itp” specifies the force field, it is necessary for GROMACS, and it is checked by the RMC program, what type of force field is used, as some initialization has to be done accordingly (like in case of OPLSAA the 1-4 interactions are calculated by scaling, while other force fields use explicit parameters.

I. Structure of the tabulated potential file

The tabulated potential file should contain the *r* (Å) – *U(r)* [kJ] values with equidistant *r* spacing. First line: the number of points, Second line is empty or comment. From the third line the *r* (Å) – *U(r)* [kJ] values, one per row.

```
990

0.01000000         0.00940000
0.02000000         0.00939000
0.03000000         0.00938000
0.04000000         0.00937000
```

III. Usage of the RMC program

A. Compilation of the program

As the program was developed with the possibility to be run on different platforms, due to the differences of the operating systems and the available compilers some code changes are necessary before compilation. The program was tested both on PC having Windows operation system using Microsoft Visual C++ compiler, and on GNU/LINUX platform.

A.1. Pre-processor directive for code building

Pre-processor directives regulate the conditional building of the code. The description of the options regulating the different building of the code can be found in the header file *altern.h* for Windows based compiler, like Microsoft Visual Studio having the simple form of #define PARAMETER. The options can be turned on in LINUX environment by passing the appropriate command line argument to make. In WINDOWS environment, if we want to choose the given option, then the #define PARAMETER belonging to it has to be in the code, if not then it has to be commented out from the code preceding it with //. Here the available option will be given.

Choosing the platform:
#define _MICROSOFT_VC
If switched on, compiling using MS Visual C++ with WINDOWS is assumed. If option _GNU_LINUX is not passed to the compiler explicitly, then _MICROSOFT_WINDOWS is assumed. DO NOT SWITCH THIS OFF, as in case of the Linux Makefile the _GNU_LINUX option is automatically switched on.

Normal or test mode:
#define _TEST_MODE
Whether the code will be built for normal running (this is what usually needed), or for running in test mode. This has to be disabled, if the program is used in normal running mode.
This was introduced during the testing of the program to ensure that the random number generator starts with the same value each time, and the program runs to a fixed amount of steps to make the results produced by the different versions comparable. It is also useful for performance testing. In fixed format the run will stop at ngenerated or naccepted steps, can be set by run time (-/+ value). In free format RUN-TIME is not processed, if nothing given, then it runs to a number of generated steps specified by LAST_MOVE_DEF in *units.h*. Default value can be also changed with key LAST-GEN or LAST-ACC in case of free format parameter file.

χ^2 component output:
#define _WRITE_CHI2_DETAIL
If it is on, the χ^2 components will be written in to the *.chi file for every rejected move.

How to handle neutron, X-ray and electron diffraction fitting:

#define _MUL_SCAT_VECTOR

In case of neutron, X-ray or electron diffraction, *Q·S(Q)*, *Q·F(Q)* or *g·F(g)* is assumed at reading the experimental data, and this is calculated and fitted

How to handle g(r) fitting:

#define _R_SWITCH_GR_MIN_1

If this is option is on, then r^N*·[g(r)-1]* is assumed at reading the experimental data for *g(r)* sets, and this will be calculated and fitted. N is specified by R_SWITCH_POWER_DEF which is located in units.h. Default value can be also changed with key R-SWITCH-POWER in case of free format parameter file.

Neighbour list saving:

Both only have effect, if the neighbourlist is calculated (cosine distribution of bond angle, common and second neighbour constraints, if details of the coordination constraints should be saved, and for Aenet).

#define _NEI

If it is on, saving the neighbour list into the *.nei file.

#define _NEIE

If it is on, the squared neighbour distances and vector is also saved to the *.nei file.

Average atomic move:

#define _AV_MOVE

If it is on, the distance so far the atoms moved from their starting positions in each direction is separately accumulated for every atom, and the average is displayed on screen at every run status screen display.

Local invariance calculation:

#define _LOCAL_INV

If it is on, local invariance is calculated.

Local invariance histogram saving:

#define _LOCAL_INV_NS

If it is on, the local invariance histogram is not written to disk to save time.

Use non-periodic boundary conditions:

#define _NO_PERIODIC

If it is on, the periodic boundary conditions will not be used, and the a spherical sample with R0<half box length /2 will be simulated.

Handle electron diffraction:

#define _READ_EDIFF

If it is on, the program will read the number of ediff data sets among Xray and EXAFS in case of the fixed format, otherwise not, so the old format can be used for the fixed format **.dat* files, Incase of free format it can handle EIFF regardless of this switch.

Advanced geometric constraints:

#define _ADVANCED_GEOM_CONST

Use advanced geometric constraints as Common Neighbour Constraint, Second Neighbour Constraint and Bond Valence Sum Constraint.

Integer 64:

#define _USE_INT64

If it is on, then hopefully 8 bytes (64-bit) long integers will be used for typedef longint variables depending on the compiler, if not then probably 4 bytes. See chapter III.B for details.

ATLAS usage

#define _ATLAS

Regulating the LINUX platform based ATLAS library usage for matrix operations. The libraries have to be installed separately (see the ATLAS web site), only use this option if they are installed! The installed ATLAS libraries are using the BLAS routines, optimised for the given platform, and can increase the speed of the vector-vector, matrix-vector and matrix-matrix operations.

Old style header files:

#define _OLD_HEADER

It should be on only if old style header files (like name.h) are used (this depends on the compiler), which is most probably not the case nowdays, so normally should be commented out.

Compiling for MAC:

#define _CODE_WARRIOR_MAC

If it is on, compile the code for Code Warrior on Macintosh, but the RMC++ code was never tested on this platform, so there is no guarantee that it can be compiled without any change! DEFAULT is the PC or UNIX version, this option has to be commented out normally

A.2. Constant values

There are some constant values in the *units.h* file, which can be altered if need arises. These are:

```
#define PI 3.14159265359
#define SQRPI 1.7724539 //square root of Pi
#define INVPI 0.3183099 //inverse of Pi
#define SQRT3 1.7320508075688772 //sqrt(3) (maximum value in sftable)
#define k_BOLTZMANN 1.38064852e-23 //J/K
#define K_SHIFT_CONST 0.262468423789 //2*m_e/h_bar/h_bar [1/A/A/eV] for EXAFS shift calculation
//the tolerable difference coming from the different number representation in the
//binary and decimal number system (if the numbers are represented by 15 digits
//after the decimal point)
#define TOLERANCE 1.0e-15
//tolerable difference at some comparisions
#define TOLERANCE2 1.0e-13
//this is used to ensure the accuracy of the bin->dr conversion
#define GRID_TOL 1.0e-14
//this is a safety increase for array dimensions in NeighbouList object
#define SAFE_ADD 20;
//used during the load of CoordNumbConst and AvCoordConst
#define LOAD_TOL 1.0e-8
//defining the confidence interval used for the calculation of "negative" cosine distribution of
bond angles constraints
#define CONF_INT 3
//size of a line for the line buffer for some file processing
#define LINE_SIZE 200
//The CACHE related things are architecture dependent, the given values and caching concept is for
the Intel64 architecture.
#define NUMBER_OF_CACHE_LINES_TO_FETCH 1//Number of cache lines to cache in the same time
(prefetch)
#define CACHE_LINE_SIZE 64 //Byte .Size of the L1 cache, needed in some cases to optimise cache
usage. False sharing between
                                         //threads has to be prevented. This can happen, when
although different threads are writing different
                                         //memory addresses, but the addresses are so close to
each other, that they would be cached together
                                         //into the same cache line (are inside the same
CACHE_ALIGNMENT block). Because of this, if one part
                                         //of a cache line is modified, the whole cache line is
written back to memory, so different threads may
```

```
                                         //want to write the same part of the memory holding back
each other causing if this happens too often
                                         //to slow the performance down.
#define CACHE_PADDING       NUMBER_OF_CACHE_LINES_TO_FETCH*CACHE_LINE_SIZE //the thread segments of
some arrays have to be separated
                                         //at least with CACHE_PADDING-size(data_type) amount of
bytes have to be kept between the threads segment data

// The size of the file names
#define FILE_NAME_SIZE 50
// The size of names
#define NAME_SIZE 100
//Max number of items to read from topology
#define INPUT_SIZE 50
//number of cosine distr method
#define N_COS_METHOD 4

//These are needed for the topology in case of MD-like molecules
//number of recognized GROMACS directives
#define N_GR_DIR 12
//number of recognized preprocessor directives
#define N_COMP_DIR 5
//number of force field types
#define N_FORCEF 7
//default value for the number of molecule types
#define N_MOLTYPE 5
//Maximum number of segments in the preprocessor arrays (number of topology and include top.
files)
#define N_TOPFILE 5
//Maximum number of active (embedded) ifdef statements in a file in the same time
#define N_ACT_IFDEF 5
//number of different potential types (harmonic bond and angle and Ryckaert-Bellemans dihedral)
#define N_POT_TYPE 3
//number of different dihedral functions
#define N_DIH_FUNCT
//for the Coulomb interaction (1/4/pi/epsilon0) (kJ*A/mol/e2)
#define f_Coulomb 1389.35485

//if atoms (or virtual sites) are too close, the potential can be extremely high, wiping out the
meaningful digits
//during updating the potential.This is checked during the updating of the exclusions, where this
is most likely to happen.
#define POT_WARNING 1.0e15

//Default values
#define LAST_MOVE_DEF 10000//the run will end at ngenerated or naccepted = end_move , this is the
default
#define WRITE_LOG_DEF 0 //creating a logfile
#define DEBUG_DEF 0 //write debug info
#define LOG_NONLIN_STEPS_DEF 0 //write the iteration steps to logfile
#define WRITE_EXAFS_COEFFS_DEF 0 //write the EXAFS coeffs into *.expt, only used if E0 shift
applied, as it can be a lot of matrix elements
#define R_SWITCH_POWER_DEF 2//if _R_SWITCH_GR_MIN_1 is on, this value is used
#define SFACTOR_SIZE_DEF 501 //number of elements in the surface factor arrays

#ifdef _LOCAL_INV
       //step for resetting the thread boundaries for the neighbour atoms, if distance based
calculation is used the
       #define LOC_LOAD_BALANCE_STEP 1000
#endif

//Default values for free format data files
#define AUTO_CUTOFF_DEF 1
#define RSPACING_DEF  0.1
#define MAX_MOVES_DEF 0.1
#define MOVEOUT_DEF 1
#define CFG_COLL_DEF 0
```

```
#define CFG_COLL_FREQ_DEF 1
#define PRINTSTEP_DEF 10000
#define TIMELIM_DEF 300.0
#define TIMESAVE_DEF 2.0
#define TOOCLOSE_FRACTION_DEF 0.5  //the fraction the moved atoms are chosen from among the 'tooclose' atoms rather than from all the atoms
#define FNC_DEF 0
#define BINSHIFT_DEF 0.0
#define XMAX_DEF 1.0
#define LOAD_HIST_DEF 0
#define HIST_BUFF_DEF 0
#define HIST_STEP_DEF 1
#define CUSTMOVE_DEF 0
#define MAX_GRIDATOM_DEF 5
#define NTHREADS_DEF 1
#define OLD_OUT_DEF 0
#define SUM_PPCF_DEF 0
#define NMOVED_DEF 1
#define LEAD_SERIES_IND_DEF 1
#define SWAP_FRACTION_DEF 0.0
#define E0_SHIFTSTEP_DEF 1000
#define AXS_SHIFTSTEP_DEF 1000
#define IQ_BACKG_CORR_STEP_DEF 1000

#define NO_VALUE_DEF -999.0 //no value was supplied at optional data reading
#define RENORM_DEF 0 //used for all the renormalization parameters, no renormalization by default
#define SUBTRACT_DEF 0.0
#define USE_RFACTOR_DEF 0
#define USE_CUBIC_DEF 0
#define EXAFS_CHI2_POWER_DEF 3
#define MAX_E0_SHIFT_DEF 0.0 //this means no shift, even if the DELTAE0_NGRID line is given
#define EXAFS_NGRID_DEF 0  //this means no shift, even if the DELTAE0_NGRID line is given
#define IQFIT_DEF 0//whether to fit I(Q) in case of x-ray data sets
#define IQ_ALPHA_DEF 0.0 //for the I(Q) fitting, initial guess
#define READ_COEFFS_DEF 0//whether to read the X-ray coefficients, from free format version #003, for #002 it is 1 (read coeffs) to be able to use the old #002 *.dat without modification
#define NONLIN_MAX_NIT_DEF 1000 //maximum number of iterations for the non-linear regression in case of I(Q) fit
#define NONLIN_EPSILON_DEF 0.001//stop iterating, if delta_par smaller than this
#define NONLIN_LAMBDA_DEF 0.001 //lambda for nonlin regression
#define NONLIN_FACTOR_DEF 10.0 // multiply or divide lambda by this during iteration
#define NONLIN_TERMINATE_DEF 10//terminate the program after this many failed regression, counted after the last successful regression
#define IQBACKGCORR_DEF 0//no background correction for I(Q) by default
#define IQ_MU_DEF 0.7//initial correction factor for I(Q) background correction
#define IQ_DMU_MAX_DEF 0.2//initial correction factor for I(Q) background correction
#define COMPTON_DEF 1//whether to use Compton scattering contribution
#define BVS_B_DEF 0.37//default value for the BVS b parameter

#define WRITE_CNC_DETAIL_DEF 0//whether to write detailed cnc information

#define POT_TYPE_DEF 1
#define POT_CHI_LOW_LIM_FRACTION_DEF 2.0 //this means that it is not set by default
#define POTENTIAL_DEF 0
#define LEAD_SERIES_IND2_DEF 1
#define NB_WEIGHT_MODE_DEF 2
#define VDW_COMB_RULE_DEF 0
#define CUTOFF_DEF 1.0//in reduced units
#define FUDGE14_DEF 1.0
#define LJ_REP_N_DEF 12

#define TEMPERATURE_DEF 298.0

#define DCOSTHETA_DEF 0.05
#define DISTRIB_TYPE_DEF "GAUSSIAN"

#ifdef _LOCAL_INV
```

```
        #define LOC_CHI2_MODE_DEF 1
#endif
#ifdef _NO_PERIODIC
#define RECENTRE_FLAG_DEF 0
#endif

//AENET
#define WRITE_ATOMIC_ENERGY_DEF 0 // do not write energy for all the atoms at each save
#define RELAX_DEF 0 //whether to relax the atoms inside cutoff
#define AENET_STEP_DEF 5//calculate ANN in each AENET_SET_DEF steps by default
```

A.3. Compilation on Linux platform, the usage of the Makefile for RMC_POT

The supplied Linux Makefile can have the following command line options, which will regulate the building of the code. The status of the switches in *altern.h*, whether they are commented out or not is of no consequence, as the Makefile will always pass the Linux platform specific _GNU_LINUX switch to the compiler, and the options in the *altern.h* will be bypassed. Instead, an option can be switch on by passing command line arguments to the make. The name of the executable will contain indicators of the used option switches to avoid confusion. The file names will always begin with '*rmcp*' and end with '*.exe*'.

Table 8: The command line options for make, and the indicators in the executable name are the following:

Command line argument	file name indicator	option switch
AENET=0	_aen	for _AENET
AGC=0	_agc	for _ADVANCED_GEOM_CONST
EDIFF=0	_ediff	for _READ_EDIFF reading electron diffraction data as well
LOC=0	_loc	for _LOCAL_INV
LOC=ns	_locns	for _LOCAL_INV_NS
MULQ=0	_mq	for _MUL_SCAT_VECTOR
NOPER=0	_nop	for _NO_PERIODIC
RGR=X	_rgrX	for _R_SWITCH_GR_MIN_1, X is the power of r, only given, if X>1
VIBRAMP=0	_vamp	for _VIBR_AMP
NEI=0	_nei	for _NEI
NEI=e	_neie	for _NEI and _NEIE
OH=0	_oh	for _OLD_HEADER
AVM=0	_avm	for _AV_MOVE
AT=0	_atlas	for _ATLAS
I64=0	_i64	for _USE_INT64
TEST=0	_t	for _TEST_MODE
TESTDUR=0	_tdur	for _TEST_MODE writes thread duration
CHI=0	_chi	for _WRITE_CHI2_DETAIL is ON (the χ^2 components will be written in to the **.chi* file for every rejected move)
ARCH=X	X	where X is added to the end of the file name before extension to differentiate between different architecture or versions, if necessary
INT=0	_i	compile with Intel icc
MAC=0	_mac	for _CODE_WARRIOR_MAC

For example compiling the RMC code for locale invariance calculation with vibrational amplitude calculation as well on a 64-bit architecture use

```
make =LOC=0 VIBRAMP=0 ARCH=64
```

This will result in executable named rmc_pot_loc_vamp64.exe

Always delete the **.o* object files before starting compilation with a new set of options, as make cannot detect in the object files which options were used during their compilation, and it will use the old compilation's object files instead of recompiling them if the source was not modified!

A.4. Compilation under Windows with Microsoft Visual C++

From version 1.6 there is only one project, and the code uses native c++11 threads and capable both one or multi-threaded execution.

Choose among the other preferred option switches in the *altern.h* file by turning them off (commenting out) or turning on. Then build the application. The executable will be named RMC_POT.exe, regardless the chosen option switches.

B. Using 4 or 8 bytes integers for typedef longint

Some variables were defined with the custom type **longint**. This makes it possible that at compilation time can be decided, whether 4 or 8 bytes integers should be used. The advantages of the 4 bytes integers is that the memory requirements is smaller, but in case of larger systems there can be overflow in the normal or total histogram for example, and the usage of the 8 bytes integers would be inevitable. To complicate the matter, the variable name of the 8 bytes long integers is not part of standard C++, and can depend on the chosen platform/compiler. Therefore the _USE_INT64 compiler option switches between the two possibilities.

The name of the 64-bit integer is not part of the standard C++, and it differs for Windows and the used Linux platform and compiler. The **typedef** part is in *altern.h*. As a default, **int** type is used for 4 bytes integer. For the 8-bytes integers as default in case of the Windows version **__int64** is used, and in case of the Linux version **long int** is used.

It has to be noted that the **long int** type in case of some platform/compiler combination can mean only 4 bytes integer, on Linux then try to alter the appropriate part of *altern.h* to **long long int** instead. The actual size used by the program is displayed at the beginning of each run. If your platform/complier is different, it can happen that the sizes given here are different for your compilation. It is important to know that the actual size of the **longint** variables are written at the beginning of the binary **.bcf* file, as in that case it is important to know it to be able to read the file correctly, and the program checks it, and only reads the file, if the code with which the file was generated used the same **longint** size, as the present compilation! For downward compatibility, if the size of the **longint** variable is missing from the beginning of the **.bcf* file, it tries to interpret it as an old format **.bcf* file beginning with the number of atoms and reading it accordingly!

C. About multi-threading

Computationally heavy parts are parallelized using native C++11 threads. The workload is equally (as possible) divided among the threads. Parallelization is used during the histogram and its change calculation, the *ppcf*, the initial and modified partial and *total g(r), S(Q), F(Q),F(g)* and *E(k)* calculation and the copying of the changed histogram, *ppcf* and partial parts after acceptance/rejection was decided. The local invariance update and the non-bonded potential calculation are also parallelized. The calculation of the neighbour list connected cosine distribution of bond angles is handled exclusively by the main thread.

For the comparison of the standard and the multi-threaded performance efficiency is used:

$E(p) = \frac{S(1)}{S(p)p} 100\%$, where p is the number of processors, and $S(1)$ is the elapsed time for the single-threaded standard and $S(p)$ for the p-threaded application.

Some additional information about the multi-threading used in RMC_POT and speed test results can be found in the document RMC_POT_speed_test.pdf or can be downloaded from the Documentation page of the website.

D. Starting the program, help

To get information about starting the program and the free format data file (introduced from version 1.6.2) type

- `exename –help` (general information)
- `exename –helptags` (about the implemented tags)
- `exename –helpkeys` (about the implemented keywords)
- `exename –helpkeys tag` (about the keywords belonging to this tag)
- `exename -helpkey key` (about the specified keyword)
- `exename –helpkeysman` (about the implemented mandatory keywords)
- `exename –helpkeysop` (about the implemented optional keywords)

For running the program the executable and the configuration and other necessary files have to be in the same directory. The program can be started by the executable file name followed by the following variation of command line arguments:

- `exename`
- `exename filename`
- `exename filename cont`
- `exename filename y`
- `exename filename cont y`

As the program from version 2.0 can calculate the X-ray scattering factors and from these the coefficients there are some help about the implemented chemical elements and ions:

- exename `–helpXcoeffs` (listing the available symbols)
- exename `–helpXcoeffsat` (listing the available atoms)
- exename `–helpXcoeffsion` (listing the available ions)
- exename `–helpXcoeff symbol` (listing the parameters for symbol)
- exename `–helpXcoeffsS symbol` (listing the parameters for entries beginning with symbol)
- exename `–helpXcoeffsZ Z` (listing the parameters for entries with atomic number ‘Z’)

There is help available from version 2.0 about the Compton scattering calculation tables as well:

- exename `–helpCompton` (listing all the symbols Compton is implemented for)
- exename `–helpComptonS symbol` (listing the parameters for entries beginning with symbol)
- exename `–helpComptonZ Z` (listing the parameters for entries with atomic number ‘Z’)

There is help available from version 2.3 about the implemented coherent neutron scattering lengths

- exename `–helpNcatlenth` (listing all the symbols the b_coh is implemented for
- exename `–helpNcatlenthS symbol`(listing the b_coh for the entries beginning with symbol)
- exename `–helpNcatlenthZ Z` (listing the b_coh for the entries with atomic number 'Z')

From version 2023.1 there is help about the parameters of the BVS constraint:

- exename `–helpBVparam` (listing all the atom type pairs for which parameters are implemented for
- exename `–helpBVparamS symbol`(listing the parameters for all the pairs containing symbol)
- exename `–helpBVparamZ Z` (listing the parameters for all the pairs where one has atomic number 'Z')

If the program is started without any command line option, then it will ask for the ***filename***, give it without any extension. For example if you have *my_rmc.cfg* and *my_rmc.dat*, then give *my_rmc* for the filename.

‘y’ indicates to close the window at the end of simulation, which is useful, if the output is redirected to a file, and the program is running in the background.

cont option is for continuation of the run (see I.Z).

IV. Auxiliary programs

These programs helps to create starting configuration and/or **.fnc* file. Theoretically RMC started from any initial configuration should reach the same results, only the amount of simulation time is different. Care has to be taken that at first not to small sigma should be applied (enough ‘bad’, χ^2 increasing configuration should be accepted), otherwise the simulation can get stuck in a local minimum.

For atomic systems the initial configuration can be some sort of a crystal configuration (for example created by *Crystal*), in this case it is practical to shake the crystal configuration, which can be achieved running a hard sphere RMC simulation without any experimental data, but already using the right hard sphere cut-offs and number density. This will result in a random configuration. Coordination number or average coordination number constraint can be applied, if necessary even in this stage.

To create multi-component random system from a one-component system, then simply the configuration has to be split up into appropriately sized blocks, and as usually the atoms in a block would be close to each other sometimes even causing ‘phase separation’ in the box due to the creation method a hard sphere RMC with swaps for all the available partial should be run, while the system is adequately mixed.

If the initial configuration should have a certain coordination number around a central atom, then FNC_config can be used for creating the **.cfg* and **.fnc* file as well. FNC_config can also be used for creating only an **.fnc* file based on a **.dat* file for a **.cfg* file.

For molecules the easiest way to create a prototype of the molecule with the roughly right coordinates in GROMACS **.gro* file format (see the GROMACS manual and the web site for details). Then use the GROMACS auxiliary program *genconf* to multiply the molecule in the box. Make sure that the correct box size is set at the and corresponding to the number density.

The Auxiliary programs except the original GROMACS programs can be found on the Auxiliary programs page of the RMC web site.

A. Creating fcc starting configuration with Crystal

Crystal is a simple Fortran program written by Furio Ercolessi, SISSA in 1997. It can create an fcc lattice based on the size of the unit cell (does not really matter, as it will be rescaled), the number of cells is each direction. Keep in mind that RMC can only handle cubic box! It can perturb the perfect lattice, if random displacement is set to >0. The coordinates generated by this program are in a box from 0→box size, make sure to transform them to reduced coordinates from –1→+1 box (for example it can easily be done in Excel), and apply the correct RMC version 3 format header, which can be copied from the example given in the validation package. In the RMC header the half box size has to be given in Angstrom! If this (and the number of atoms) does not result in the same number density given in the **.dat* file, then the value in the **.dat* file will be used, and the simulation box will be rescaled, and this fact is written at the screen output at the beginning of the simulation.

Crystal can only generate a one component system with certain number of atoms ($4*N_{cell}^3$) as it is an fcc crystal cell, but this can be easily transformed to a multi-component system with any number of atoms, only use the appropriate header, and generate a configuration large enough to hold the required total number of atoms (the extra atomic coordinates at the end of the file will not be used).

B. Using FNC_config

FNC_config can be used two ways. Using option 1 we can create a configuration, when a given number of atoms can be found around the central atoms between the given distance range. For this an RMC format configuration file containing the coordinates of the central atoms has to be present with name *filename.incfg*.

If option 2 is used, then only an **.fnc* file is created for a configuration. The configuration file itself is not needed, only based on the structure of this parameter **.dat* file gives the rendering of the atoms to the molecules. The concept is very similar to how the RMC indices of the first and second instances of a molecule have to be given in the topology. The structure of the **.dat* file should be the following for 10 molecules of dimethylsulfide, the numbering of the atoms id the same as in Figure 12.

Constraint type 1 is for the C-S bond, 2 for C-H bonds, 3 for C-C non-bonding distance, 4 for H-H non-bonding distance and 5 for non-bonding S-H distance.

First 5 lines form a header part, than two lines for each constrained pair, in the first line the index of the constraint type and for the first occurrence the minimum and maximum distance in Å, in the second line the RMC indices of the fnc atom pair in the first and in the second molecule.

```
90                !number of atoms in the configuration
1                 !number of molecule types
10                !number of molecules/type following each other in the same line
9                 !number of atoms/molecule type following each other in the same
line
21                !number of fnc constrained pairs/molecule following each other in
the same line
1 1.638 1.96      !constraint index
1 21 2 22   !indices of the constrained pair in the first and second molecules
1                 !constraint index
11 21 12 22 !indices of the constrained pair in the first and second molecules
2 0.944 1.24      !constraint index
1 31 2 34         !indices of the constrained pair in the first and second molecules
2                 !constraint index
1 32 2 35         !indices of the constrained pair in the first and second molecules
2                 !constraint index
1 33 2 36         !indices of the constrained pair in the first and second molecules
2                 !constraint index
11 61 12 64 !indices of the constrained pair in the first and second molecules
2                 !constraint index
11 62 12 65 !indices of the constrained pair in the first and second molecules
2                 !constraint index
11 63 12 66 ! indices of the constrained pair in the first and second molecules
3 2.36 3.11 !constraint index
1 11 2 12         ! indices of the constrained pair in the first and second molecules
4 1.44 2.0  !constraint index
31 32 34 35 ! indices of the constrained pair in the first and second molecules
4                 !constraint index
31 33 34 36 !indices of the constrained pair in the first and second molecules
4                 !constraint index
32 33 35 36 !indices of the constrained pair in the first and second molecules
4                 !constraint index
61 62 64 65 !indices of the constrained pair in the first and second molecules
4                 !constraint index
61 63 64 66 !indices of the constrained pair in the first and second molecules
4                 !constraint index
62 63 65 66 !indices of the constrained pair in the first and second molecules
5 2.04 2.74 !constraint index
21 31 22 34 !indices of the constrained pair in the first and second molecules
5                 !constraint index
21 32 22 35 !indices of the constrained pair in the first and second molecules
5                 !constraint index
21 33 22 36 !indices of the constrained pair in the first and second molecules
5                 !constraint index
21 61 22 64 !indices of the constrained pair in the first and second molecules
5                 !constraint index
21 62 22 65 !indices of the constrained pair in the first and second molecules
5                 !constraint index
21 63 22 66 !indices of the constrained pair in the first and second molecules
```

*C. Converting configuration files from GROMACS *.gro: convert_gromacs*

Converting configuration between RMC and GROMACS in both ways can be done by the convert_gromacs software. It uses a parameter file and the initial configuration file and produces the other type coordination file. It can convert consecutive GROMACS configurations from the same file to separate RMC files, or consecutive RMC (**coll.cfg*) to GROMACS as well. In case of RMC_GROMACS conversion a starting topology file can be created as well, if the fnc option is 1. The constrained distances related to bonds can be converted to bonds, the 1-3 non-bonded distances to angles, and other distances more than two bonds away can be converted to type 2 constraints (see the GROMACS manual about bonds, angles and constraints).

Only the bond and angle types have to be given for a molecule, not all the bonds and angles. So if there are more fnc pairs in a molecule with the same fnc constraint, then only one has to be given, even in the case of the angles, specify one central atom only for an angle constraint. The program will determine all the others.

```
1 3                     ! conversion type (0) RMC->MD, (1) MD->RMC number of conf. to
convert
DMeS_flex_md.cfg  3     ! RMC configuration file, number of RMC types only for MD-
>RMC conversion
DMeS_flex_md_3conf.gro  ! GROMACS *.gro file
DMeS.top                ! GROMACS *.top file
DMeS all-atom           ! Title for the MD
ffoplsaa.itp 1 1        ! force field parameter file; bond type, angle type: 'D' for
default

0     0.0103135         ! whether to rescale box (0: no, 1: yes), new number density,
if necessary
1                             ! number of different residue types
1                             ! character string (max 3 char) for each residue
defining X in the bonded and nonbonded constraint name definition cbX_index and
ncbX_index12:30 11/15/2008
1331 9  DMeS      3     ! as many line as residue types containing first the number
of this residue, number of atoms and the name (max 5 char) for the residue,  number
of bonds for exclusions (default=3)
1     2     CA    opls_209    12.011      1     -0.013      1     ! The index of this
atom in the conf. in the first then in the second molecule, atom name, atom type,
mass, charge group, charge,RMC-type(only MD->RMC)
31    34    HA1   opls_140    1.008       1     0.06        3
32    35    HA2   opls_140    1.008       1     0.06        3
33    36    HA3   opls_140    1.008       1     0.06        3
21    22    S     opls_202    32.060      2     -0.334      2
11    12    CB    opls_209    12.011      3     -0.013      1
61    64    HB1   opls_140    1.008       3     0.06        3
62    65    HB2   opls_140    1.008       3     0.06        3
63    66    HB3   opls_140    1.008       3     0.06        3
0 c2cl4.fnc       ! whether to use FNC, name of the FNC file (0:no, 1:use FNC lower
limit, 2: use FNC upper limit, 3:use FNC average limit)
2                 ! number of bond types to create for each residue
1 2         ! constraint indices in the FNC file for each desired bond const. for
each residue
185769.6 284512.0 ! force constants for each desired bond constraints separated by
space
3                 ! number of angle types to create for each residue
3 4 5 ! constraint indices in the FNC file for each desired angle const. for each
residue
518.816 276.144 292.88! force constants for each desired angle constraints separated
by space
5 1 1! index of the central atom in the molecule (according to topology) for each
desired angle const. for each residue
99.05 109.3 109.3       ! desired angle in degree for each desired angle const. for
each residue09:45 11/15/2008
0                                         ! number of non-bonded constraint types to
create for each residue
10 11 10 11                               ! constraint indices in the FNC file for
each desired non-bonded const. for each residue
```

For more detailed description see the document RMC-GROMACS_converter.pdf.

D. Converting RMC configurations to CrystalMaker text format *.cmtx

Both the text type and the binary configurations file of RMC_POT can be converted to the text type **.cmtx* file of the visualization software CrystalMaker. The converter is self-explanatory, and needs a parameter file to contain the conversion details.

It can use the **.cfg*, **.bcf* or **_coll.cfg* file for RMC coordinate file input, the last one being the text type configuration file produced with RMC with collecting the configurations options containing multiple configurations consecutively.

It is possible to convert only a part of the configurations, if not all the atoms required for the visualization, this is governed by the mode option, there are 4 possible ways to select the atoms.

option 1: The indices of the specified number of atoms for each type will be spread among the atoms of this type with as equal step as possible. As usually atoms with similar indices are close to each other, especially if no swaps were used during the simulation, this way the atoms can be found everywhere in the simulation box.

option 2: The first given number of atoms will be used from a type. (Most probably atoms close to each other are chosen this way!)

option 3: Atoms of given indices specified at the end the parameter file will be selected. (Index can range from 1 -> ntotal, and has to follow the order of atom types.)

option 4: Only particles between two parallel planes will be saved into the file. The planes have to be parallel to one of the sides of the simulation box! There have to be an integer after option 4 to specify which axis is perpendicular to the cutting plane (x:0, y:1, z:2), followed by two real numbers between -1 and +1) to specify the reduced coordinates, between which the atoms will be outputted. Example for this line of the parameter file choosing the atoms, with reduced y-coordinates are between -0.7 -> 0.5: 4 1 -0.7 0.5.

If all the atoms of a configurations has to be selected, then the quickest way to do it to use option 2 and give the number of all the atoms of each type to use.

An example parameter file is given here:

```
ccl4.cfg ! name of the input configuration file
5                ! number of configurations to convert (only meaningful for *coll.cfg)
2                ! number of used types
3 1 –0.7 0.5     ! mode option (for option 4 index of the perpendicular axis(x:0, y:1,z:2),min, max reduced coord)
C  2             ! Atom’s chemical symbol, number of atoms to use for each type
Cl 8             ! Atom’s chemical symbol, number of atoms to use for each type
1                ! Atom indices to select, only in case of option 3
2                ! Atom indices to select, only in case of option 3
2049     ! Atom indices to select, only in case of option 3
4097     ! Atom indices to select, only in case of option 3
6145     ! Atom indices to select, only in case of option 3
8193     ! Atom indices to select, only in case of option 3
2050     ! Atom indices to select, only in case of option 3
4098     ! Atom indices to select, only in case of option 3
6146     ! Atom indices to select, only in case of option 3
8194     ! Atom indices to select, only in case of option 3
```

The green part has to be included only for option 4, the blue for option 3, the number of atoms denoted by pink only for options 1-3.

If no atoms of an atom type ha to be used in case of option 1-3, than give zero for the number of atoms.

The name of the output file will be generated by the input file by replacing the original file extension by **.cmtx*. The program will decide based on the original file extension about the type of the input file, so it is important that the original RMC file extensions should be used, (**.cfg* 1 text type configuration, **.bcf* 1 binary configuration, **_coll.cfg* multiple text configurations. In case of multiple configurations atoms of the first *nconfig* configuration will be selected for output, and put into separate files with name **_coll000X.cmtx*.

The program can be started by **executablename** then it asks for the parameter file name, or by **executablename parfilename**.

Appendix 1

The derivation of the correct volume to normalize the histogram for spherical sample, without the usage of periodic boundary conditions is given below.

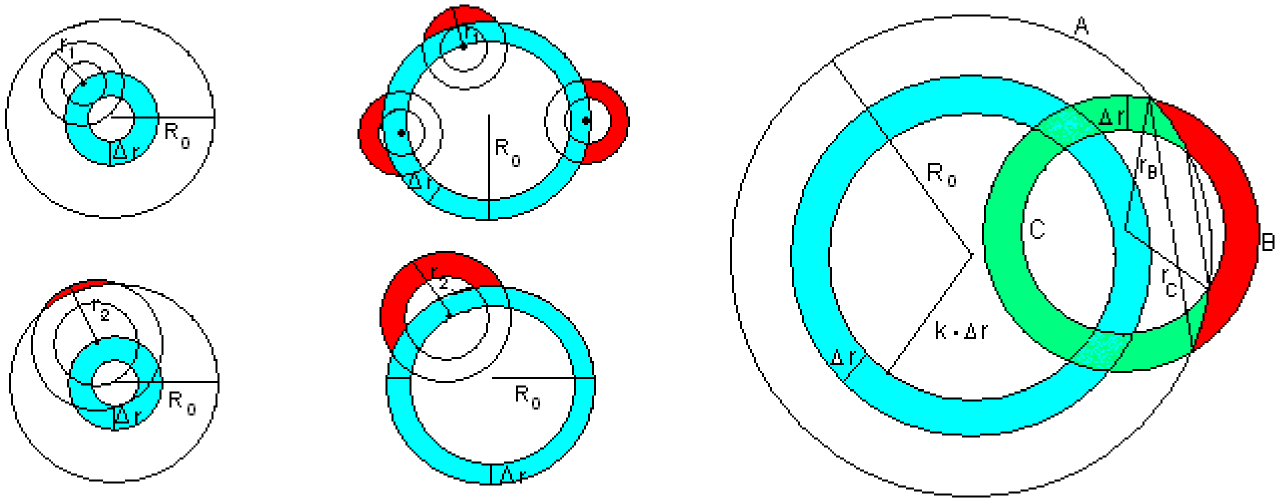


Figure 13: a) The demonstration of the protruding volumes shown in red, which are outside the spherical sample represented by sphere *A* with R_0, and should be subtracted from the appropriate volume element. b) The protruding part of the volume element (red) which should be subtracted, from the whole spherical shell of sphere *B* and *C*, and the part to use for the normalisation of the histogram (green).

First we will calculate the protruding spherical shell part of the volume element shown in red in Figure 13b. The radius of the smaller and larger spheres *C* and *B* has the relationship rB,j =rC,j +□r=(j+1)·□r, where □r is the size of the histogram bin. Let's first only consider sphere B and the yellow part of it protruding form sphere A (Figure 14a), and calculate its volume, Vout_B$\left(r_{B,j}-\Delta r/2,k\right)$. It has to be noted that the protruding volume depends obviously on the radius of the spheres B and C, but they have a fixed relationship; and depends on the position of their origin, which can be placed at the middle of each histogram bin (*k* being the index of the bin staring with 0) that is why we have two independent variable for V_{out_B}.

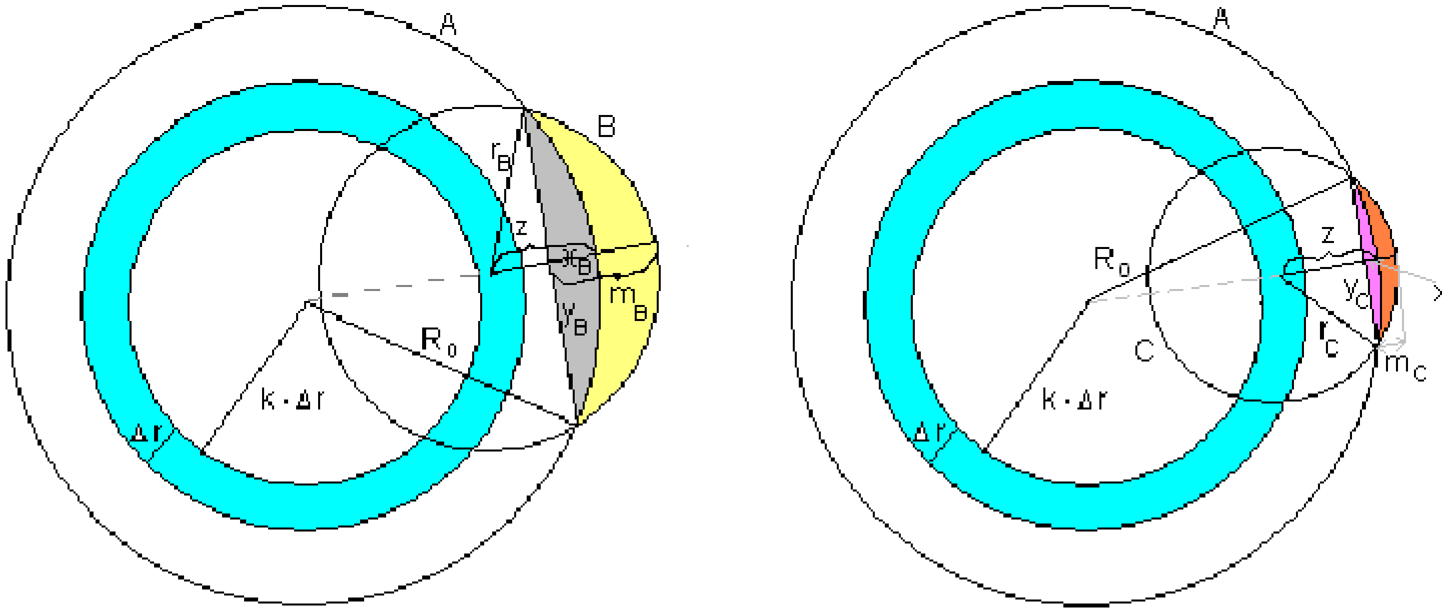


Figure 14: a) The notation for the calculation of the protruding volume of sphere *B* form sphere *A*. The spherical caps of sphere *A* and *B* cut off by the same plane have a circular base with y_B radius. The volume of the spherical cup *A* (cut off from sphere *A*) is $V_{A\_yB}\left(r_{B,j}-\Delta r/2,k\right)$ and it is shown in grey, the volume of the spherical cup *B* cut off from sphere *B* is $V_{B\_yB}\left(r_{B,j}-\Delta r/2,k\right)$ and it is the cumulated volume shown in grey and yellow. b) Similarly shown the protruding volume of sphere *C* form sphere *A*: the spherical caps of sphere *A* and *C* cut off by the same plane have a circular base with y_C radius. The volume of the spherical cup *A* (cut off from sphere *A*) is V_{A_yC} $\left(r_{B,j}-\Delta r/2,k\right)$ and it is shown in pink, the volume of the spherical cup *C* cut off from sphere *C* is V_{C_yC} $\left(r_{B,j}-\Delta r/2,k\right)$ and it is the cumulated volume shown in pink and orange.

The volume of the spherical cap cut off from sphere *A* is denoted by $V_{A\_yB}\left(r_{B,j}-\Delta r/2,k\right)$ (Figure 14), and can be calculated as

$$V_{A\_yB}\left(r_{B,j}-\Delta r/2,k\right)=\frac{\pi}{3}x_B^2\left(3R_0-x_B\right) \tag{E 35}$$

where $x_B=m_B+z-r_{B,j}$, and the volume of spherical cap cut from sphere B is $\boldsymbol{V_{B\_yB}}\left(r_{B,j}-\Delta r/2,k\right)$ can be calculated similarly as

$$V_{B\_yB}\left(r_{B,j}-\Delta r/2,k\right)=\frac{\pi}{3}m_B^2\left(3r_B-m_B\right) \tag{E 36}$$

Based on Figure 14a and b $z=R_0-(k+0.5)\Delta r$. To determine x_B, two equations can be written based on the right angle triangles in Figure 14a, from which x_B can be determined according to E 38, and then m_B can be calculated as well.

$$\begin{aligned}&I. \quad R_0^2=(R_0-x_B)^2+y_B^2\\&II. \quad r_{B,j}^2=(z-x_B)^2+y_B^2\\&x_B=\frac{z^2-r_{B,j}^2}{2z\ -2R_0}\end{aligned} \tag{E 37}$$

So the yellow volume can be calculated as $V_{out\_B}\left(r_{B,j}-\Delta r/2,k\right)=\boldsymbol{V_{B\_yB}}\left(r_{B,j}-\Delta r/2,k\right)$- $\boldsymbol{V_{A\_yB}}$ $\left(r_{B,j}-\Delta r/2,k\right)$.

Similarly we can write that $x_C=m_C+z-r_{C,j}$ and x_C can be determined from the following equations based on the right angle triangles in Figure 14b, and after that m_C: can be calculated as before.

$$\begin{aligned}&I. \quad R_0^2=(R_0-x_C)^2+y_C^2\\&II. \quad r_{C,j}^2=(z-x_C)^2+y_C^2\\&x_C=\frac{z^2-r_{C,j}^2}{2z-2R_0}\end{aligned} \tag{E 38}$$

The spherical caps cut off with the circular base with radius y_C from sphere *A* and sphere *C* can be calculated as

$$\begin{aligned}&V_{A\_yC}\left(r_{B,j}-\Delta r/2,k\right)=\frac{\pi}{3}x_C^2\left(3R_0-x_C\right)\\&V_{C\_yC}\left(r_{B,j}-\Delta r/2,k\right)=\frac{\pi}{3}m_C^2\left(3r_C-m_C\right)\end{aligned} \tag{E 39}$$

The protruding pat of sphere C from sphere *A*, shown in orange in Figure 14b is defined by V_{out_C} $\left(r_{B,j}-\Delta r/2,k\right)=$ $V_{C\_yC}\left(r_{B,j}-\Delta r/2,k\right)$-$V_{A\_yC.}$ $\left(r_{B,j}-\Delta r/2,k\right)$. Based on this the red protruding spherical shell is $V_{out}\left(r_{B,j}-\Delta r/2,k\right)=$ $V_{out\_B}\left(r_{B,j}-\Delta r/2,k\right)$- $V_{out\_C}\left(r_{B,j}-\Delta r/2,k\right)$. This is true, when both sphere *B* and *C* are protruding from sphere *A*, which happens, if $r_{B,j}-z>\Delta r$. And in this case the volume element remaining inside sphere *A* is

$$V_{in}\left(r_{B,j}-\Delta r/2,k\right)=\Delta V_j-V_{out}\left(r_{B,j}-\Delta r/2,k\right)=$$
$$\frac{4}{3}\pi\left(r_{B,j}^3-r_{C,j}^3\right)-V_{out}\left(r_{B,j}-\Delta r/2,k\right)$$ E 40

If only sphere *B* is protruding, which happens if $0<r_{B,j}-z<\Delta r$ then $\boldsymbol{V_{out}}\left(r_{B,j}-\Delta r/2,k\right)=V_{out\_B}\left(r_{B,j}-\Delta r/2,k\right)$, and $V_{in}\left(r_{B,j}-\Delta r/2,k\right)$ can be calculated according to E 40. If none of the sphere are protruding, then no volume element correction necessary.

Now we have to decide, in what extent we have to correct each volume element, as this depends on the origin of sphere *B* and *C*, which is determined by the position of the particle in question. We will assume near homogenous distribution for the particles inside the spherical sample *A*, so we can calculate the average number of particles in each concentric spherical shell histogram bin from the number density, as

$$\rho=\frac{N}{V}=\frac{N_k}{\Delta V_k}\qquad \Delta V_k=\frac{4}{3}\pi\left(r_{k+1}^3-r_k^3\right)\qquad k=0\ldots N_{bins}$$ E 41

where N_k is the average number of particles in the k^{th} histogram bin, N_{bins} is the number of histogram bins. The corrected volume element Δv_j can be calculated as the average of the volume element $V_{in}\left(r_{B,j}-\Delta r/2,k\right)$ for each particle:

$$\Delta v\left(r_j\right)=\frac{N_{in,j}\Delta V_j+\sum_{k>Nbins-j-1.5}^{Nbins}N_k\Delta V_{in}\left(r_{B,j}-\Delta r/2,k\right)}{N}$$ E 42

where ΔV_j is the uncorrected volume element (shown by dark green in Figure 15), $N_{in,k}$ is the number of particles inside the blue sphere, *N* is the total number of particles, N_k is the number of atoms in the k^{th} spherical shell, and $V_{in}\left(r_{B,j}-\Delta r/2,k\right)$ (light green) is the part of the volume element *j* inside sphere *A* for particles positioned in the k^{th} shell shown in dark yellow.

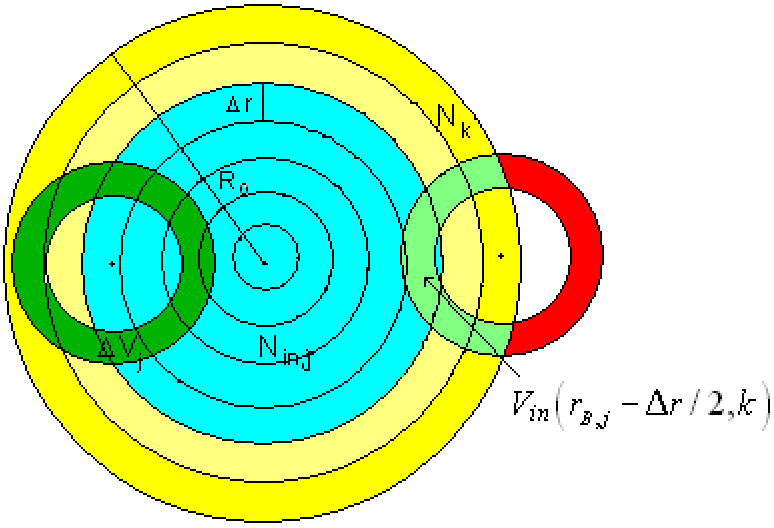


Figure 15: The calculation of the average volume element for the normalization of the j^{th} histogram bin. For those $N_{in,j}$ particles inside the blue spherical part of the sample, no correction is necessary. For the particles in each spherical yellow shell only the light green part of the original volume element can be used during the averaging.

---

[1] McGreevy, R.L., Pusztai, L.: Molec. Simul. **1**, (1988) 359.

[2] Pusztai, L.: J. Non-Cryst. Sol. **227-230**, (1998) 88.

[3] McGreevy, R.L.: J. Phys.: Cond. Matter **13**, (2001), R877.

[4] Evrard, G., Pusztai, L.: J. Phys.: Cond. Matter. **17**, (2005) S1.

[5] Gereben, O., Jóvári, P., Temleitner, L., Pusztai, L.: J. Optoelectron. Adv. Mater. **9**, (2007) 3021

[6] Wassmaier, D., Kirfel, A. Acta Cryst. **A51** (1995).,416

[7] Balyuzi, H.H.M. Acta Cryst. (1975). A31, 600

[8] Cliffe, M.J., Dove, M.T., Drabold, D.A., Goodwin, A.L. *Phys Rev Lett.* **104**, (2010) 125501.

[9] Barrett, J.H. *Phys. Rev.B* **3** (1971), 1527