Development of the time-dependent reverse Monte Carlo simulation, RMC*t*

IOP Publishing Journal of Physics: Condensed Matter

J. Phys.: Condens. Matter **19** (2007) 335223 (22pp) doi:10.1088/0953-8984/19/33/335223

Development of the time-dependent reverse Monte Carlo simulation, RMC*t*

O Gereben[1], **L Pusztai**[2] **and R L McGreevy**[3]

[1] Ardeus Ltd, H-2030 Érd, Fenyőfa ucta 32, Hungary
[2] Research Institute for Solid State Physics and Optics, Hungarian Academy of Sciences, H-1525 Budapest, PO Box 49, Hungary
[3] ISIS Facility, RAL, CCLRC, Chilton, Didcot, Oxfordshire OX11 0QX, UK

Received 9 May 2007, in final form 10 May 2007
Published 4 July 2007
Online at stacks.iop.org/JPhysCM/19/335223

Abstract

The aim of the present work is to develop a method of time-dependent reverse Monte Carlo modelling (RMCt), to model the atomic dynamics of materials based on data from inelastic neutron scattering experiments, such as the *dynamic pair correlation function*, $g(r, t)$ or the *dynamic structure factor*, $S(Q, \omega)$.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Reverse Monte Carlo (RMC) modelling is a general method of structural modelling based on experimental data, mainly from diffraction [1, 2]. RMC was originally developed for modelling the structures of liquids and glasses but has now also been applied to crystalline and magnetic structures. Many different sorts of data can be used and many different types of system can be modelled. The method produces static 'snap-shot' images of structures, but has also been extended to produce dynamic images, 'movies', by using Monte Carlo moves as an analogue of time steps [3]. While this is 'false dynamics', and has no real time scale, it can be extremely instructive in understanding how structure relates to dynamics.

This raised the idea of developing an extension of the RMC method, known as RMCt (t denoting time), to produce dynamic models based on dynamical data, for example the dynamic structure factor $S(Q, \omega)$ as measured by inelastic neutron scattering. The original concept was initially developed by McGreevy and Zetterström [4] and then further developed by Evrard [5] on the basis of the RMC++ software [6]. In this paper we present the RMCt algorithm together with full details of its implementation. We describe some of the lessons learned and present initial results of the application to 'ideal' data for a simple liquid.

As an example of the potential application of RMCt we can consider the case of ion conductors. There have been numerous RMC studies of the structures of ion-conducting

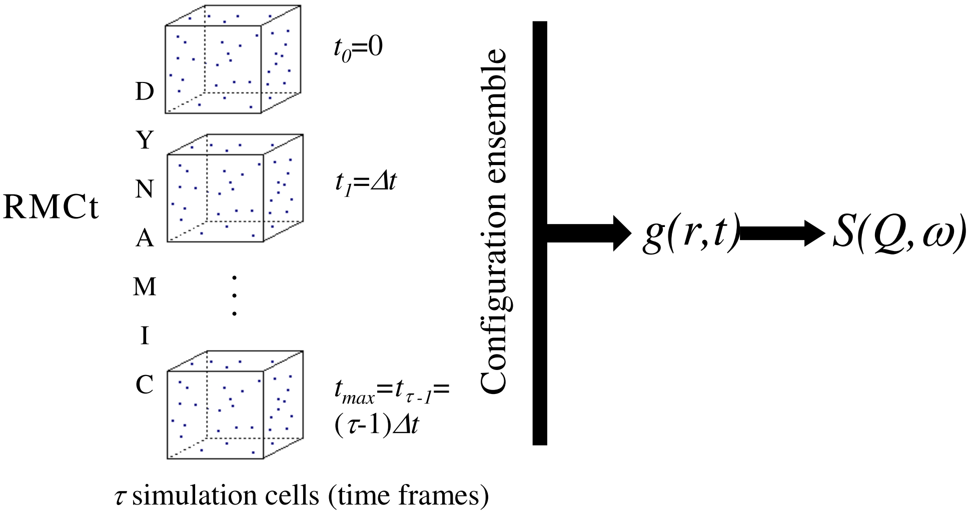

Figure 1. The basic idea of RMC*t* simulation. There are τ simulation cells representing consecutive time frames from the system's trajectory separated by Δt time difference. All the available particle distances inside the configurations and among them have to be calculated, from which the dynamic pair correlation function, $g(r,t)$ can be determined. $S(Q,\omega)$ is calculated from $g(r,t)$ by double Fourier transformation (see appendix A for details).

crystals and glasses, which have been used to help understand the mechanism of ion conduction [7]. However, the structural models themselves provide no direct information on this dynamic process. In an 'ideal' situation an RMC*t* dynamical model would be structurally consistent with an RMC structural model, but provide an additional direct view of the conduction process and timescales involved. Correlations between ionic motions should be readily visible in an RMC*t* model, but can only be inferred very indirectly, if at all, from an RMC model. Since RMC can provide a range of structural models consistent with the available data, one would expect the RMC*t* model, simultaneously fitted to structural and dynamical data, to narrow down this range and hence provide a 'better' structural model.

The present study is intended to be a description and demonstration of a new method, using the well-known example of (model) liquid argon as a test case; it has to be stressed that it is not our aim to provide new information concerning the *system* itself. Molecular dynamics simulations for liquid argon are already very consistent with experiment and have been analysed in detail many times in the literature.

2. RMC*t* modelling

An RMC model consists of a configuration of N atoms (where N is typically several thousand) defined by three coordinates (x, y, z). The configuration may be of any shape compatible with the use of periodic boundary conditions; its volume then determines the density, which should be consistent with that of the experimental system being modelled. Atoms are moved randomly and a set of 'data' is calculated on the basis of the atomic coordinates in the configuration; moves are accepted or rejected in order to improve the agreement between the calculated and experimental (structural) data. Constraints can be applied as required, e.g. minimum atomic sizes or chemical bonding.

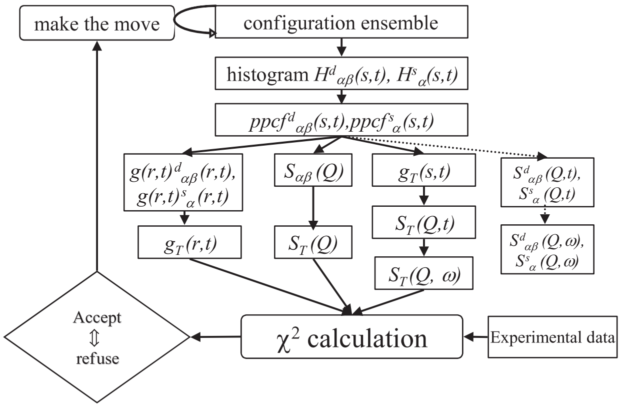
make the move
configuration ensemble
histogram $H^d{}_{\alpha\beta}(s,t)$, $H^s{}_{\alpha}(s,t)$
$ppcf^d{}_{\alpha\beta}(s,t)$,$ppcf^s{}_{\alpha}(s,t)$
$g(r,t)^d{}_{\alpha\beta}(r,t)$, $g(r,t)^s{}_{\alpha}(r,t)$
$S_{\alpha\beta}(Q)$
$g_T(s,t)$
$S^d{}_{\alpha\beta}(Q,t)$, $S^s{}_{\alpha}(Q,t)$
$g_T(r,t)$
$S_T(Q)$
$S_T(Q,t)$
$S_T(Q, \omega)$
$S^d{}_{\alpha\beta}(Q,\omega)$, $S^s{}_{\alpha}(Q,\omega)$
Accept ⇕ refuse
χ^2 calculation
Experimental data

Figure 2. The calculation sequence of RMC*t*. (The coordination constraints and velocity distribution constraint calculations are not shown.)

In RMC*t* the model consists of a sequence of configurations (see figure 1), representing the positions of the N atoms as a function of time. It is actually identical to the output of a molecular dynamics (MD) simulation. The velocities of the atoms can be calculated from the changes in coordinates in successive configurations and the defined time step between them, and the temperature is related to the kinetic energy distribution. The dynamic pair correlation function $g(r, t)$ (see appendix A) can be calculated from the sequence and Fourier transformed to give the dynamic structure factor, which can then be compared to experimental data.

Figure 2 shows the calculation sequence of RMC*t*. In a randomly chosen configuration a randomly chosen atom (or atoms) is (are) moved. In the case of a multiple atom move, a 'custom' molecular move is also possible. All interparticle distances are calculated, including distances between atoms in different configurations. The result is stored in a histogram $H(s, t)$ that has two independent parameters; s has the dimension of distance whereas t denotes time. Partial dynamic pair correlation function(s) $g_{\alpha\beta}(r, t)$ are calculated from the histogram. The weighted linear combination of the partial $g_{\alpha\beta}(r, t)$ functions gives the total $g_{\mathrm{T}}(r, t)$. If the 'experimental' data are the total $g_{\mathrm{T}}(r, t)$ then the calculated and experimental $g_{\mathrm{T}}(r, t)$ are compared by calculating χ^2 and its logarithmic version, χ^2_{lg} (see equations (20) and (24) in appendix B.3). If the dynamic structure factor, $S_{\mathrm{T}}(Q, \omega)$, is used as experimental data then it is obtained by Fourier transform from the total $g_{\mathrm{T}}(r, t)$. (The partial $S_{\alpha\beta}(Q, \omega)$ are calculated at the beginning and at the end of the simulation, for information purposes only.) The χ^2 will then be calculated as a measure of difference according to equation (20) (appendix B.3). Constraints can be placed on the coordination number of an atom type, the average coordination number of an atom type and the coordination number of individual atoms (the so-called 'fixed neighbours' constraint) [6] similarly to RMC, but new types of constraints are also possible. If an auxiliary $S(Q)$ constraint is present, it is calculated in the same way as in RMC. Constraints imposed on the velocity distribution will be discussed later (see section 3.3). If χ^2 decreases the move is accepted; if it increases then the move is accepted with a probability that is inversely proportional to the increase of χ^2.

The calculation can be extended to multi-component systems, similarly to the static case.

Even though the steps of RMC*t* modelling are basically the same as in the static case, RMC*t* obviously requires much more CPU time. For example, the distance calculation requirements for each atomic move are a hundred times greater and the initial distance calculation requirements are 10^4 times greater. For practical reasons this means that RMC*t* models are currently limited in terms of the number of atoms and time steps. We have so far used up to 3000 atoms and 2000 time steps; with a time step of 5 fs, this gives a total time of 10 ps, which is reasonable for studying the dynamics of simple liquids. If $S(Q, \omega)$ is to be fitted then the maximum distance and time should be large enough to avoid significant truncation errors in the transform from $g(r, t)$.

Further technical aspects of the algorithm can be found in appendix B. As a result of the above calculation scheme, model data could be very well reproduced, as shown in figures 7 and 8 (section 4).

3. Details of the algorithm

3.1. Fitting criteria

In RMC studies of liquids and glasses the functions fitted are normally $g(r)$ or $S(Q)$. These are typically weakly oscillating functions with the difference between the minimum, maximum and asymptotic values being relatively small. During the definition of χ^2 the σ parameter (see appendix B) is typically taken as a constant, which is a suitable choice in such cases. In RMCPOW [8], where powder diffraction data are fitted, there are multiple Bragg peaks with large differences between the maximum and minimum values, so a different definition of χ^2 is used. In RMC*t* the question of how to define the fitting criteria is more problematic. The self-correlation function $g^{\mathrm{s}}(r, t)$ has a single very sharp peak at ($r = 0$, $t = 0$), while the function at other (r, t) has a relatively low value. $g^{\mathrm{d}}(r, t)$ is essentially similar to $g(r)$. The differences between the model and 'experimental' $g(r, t)$ can vary far more during the course of modelling than in RMC. The strategy we have used for solving this is to simultaneously fit $g(r, t)$ and $lg[g(r, t) + 1]$ using different values of the weighting factor, σ. As the simulation progressed the weighting of the logarithmic function was given less and less importance. This is only relevant for the initial tests, fitting $g(r, t)$, because in the course of fitting $S(Q, \omega)$ the conventional χ^2 can be applied.

3.2. Maximum displacement of individual atoms between consecutive configurations

$g^{\mathrm{s}}(r, t)$ gives information about the displacement of individual atoms from their original positions during time interval t. For $t = 0$, obviously $r = 0$ for all atoms. (This delta-function point is not included in the $g^{\mathrm{s}}(r, t)$ calculated by RMC*t*, but its contribution is included during the double Fourier transformation used for calculating $S(Q, \omega)$.) As indicated previously, a time step of order 5 fs gives a reasonable balance between the total time of the simulation and the number of time steps. However, this limitation does cause other problems.

To illustrate this we have carried out an MD simulation [9] of a 86.3 K Lennard-Jones (LJ) system (mimicking liquid Ar) with $N = 452$, $\tau = 50$ and $\Delta t = 5$ fs. The value of dr, the r spacing in the $g(r, t)$ histogram, was chosen as 0.1 Å, which might be a typical practical value used in RMC*t*. The maximum displacement of individual atoms between consecutive configurations was found to be 0.03 Å, so for d$r = 0.1$ Å it takes typically three time steps to change the $g^{\mathrm{s}}(r, t)$ histogram. This means that $g^{\mathrm{s}}(r, t)$ at this dr resolution does not contain sufficiently detailed information to properly determine the short-time dynamics at the given temperature. It is not practical to use significantly smaller dr (of the order of 10^{-4} would be required), because of the enormous number of histogram bins, most of which would be zero at larger r. The use of a non-uniform histogram bin size would also be complicated.

Test RMC*t* simulations using the 86.3 K Ar $g(r, t)$ as 'experimental' data confirmed the expected unrealistically large atomic displacements at small t values. To overcome this problem we have introduced an additional parameter $D_{\max}$ for each particle type to limit the maximum displacement of an atom from its image in a consecutive configuration.

3.3. Requirement for a velocity distribution constraint

Although the 'experimental' data used during RMCt modelling contain information about the dynamic properties of the system, it is probably necessary to apply additional constraints to facilitate convergence, and to produce as reliable a description of the real system as possible.

It is possible to determine the average velocity of the ith atom from its position in two consecutive configurations (time frames) ($q + 1$ and q, where the time value of the qth time frame can be described by $t = q\Delta t$, $0 \leqslant q < \tau - 1$) according to the following formula:

$$\overline{v}_{i,q} = \frac{\sqrt{(x_{i,q+1} - x_{i,q})^2 + (y_{i,q+1} - y_{i,q})^2 + (z_{i,q+1} - z_{i,q})^2}}{\Delta t} \qquad \begin{array}{c} 1 \leqslant i \leqslant N \\ 0 \leqslant q < \tau - 1. \end{array} \tag{1}$$

From the τ configurations we can determine $\tau - 1$ average velocities for an atom. From the average velocities it is possible to calculate the average velocity distribution, $f\overline{(v)}$ (averaging over all possible consecutive configuration pairs and every atom). It has to be emphasized that the calculated velocities are *average* and not *instantaneous*. Although it is the latter that is available from theory, since Δt is relatively small (5 fs in our case) we will use the former as an approximation.

For a multi-component system there are as many velocity distributions as particle types; in the current version of the program all are calculated using the same velocity histogram spacing, Δv, and maximum velocity, so the maximum velocity has to be chosen to be able to contain the widest distribution.

The velocity distribution of the ideal gas is described by the Maxwell–Boltzmann distribution [10]:

$$f(v) = 4\pi \left[\frac{m}{2\pi kT}\right]^{3/2} v^2 \mathrm{e}^{-\frac{mv^2}{2kT}} \tag{2}$$

which should be adequate for the system studied, as was confirmed by the MD simulation. Other types of systems might need a different definition of the velocity distribution.

Now we wish to investigate if practically applicable dr and Δt parameters allow for a meaningful representation of the distribution of velocities. In our 'experimental' LJ Ar test system (86.3 K, $\Delta t = 5$ fs, d$r = 0.1$ Å, created by MD simulation) the largest velocity was 600 m s^{-1}, which corresponds to a displacement of 0.03 Å. That is, the 0.1 Å resolution distorts the data allowing the occurrence of larger velocity values (corresponding to distances between 0.03–0.1 Å), as well.

A test RMCt simulation (run 1, see table 1 for details) where $D_{\max} = 0.04$ Å produced a very good fit where the difference in $g^{\mathrm{s}}(r, t)$ has virtually disappeared. The final distribution of the average velocities (see figure 3), is, however, very different from the theoretical Maxwell–Boltzmann distribution, although the 86.3 and 120 K MD systems used as 'experimental' data and starting configurations had a relatively good match with the corresponding theoretical distributions.

The average velocity distribution for run 1 (figure 3) increases monotonically up to 800 m s^{-1}, corresponding to $D_{\max} = 0.04$ Å for $\Delta t = 5$ fs, and then falls to zero. For comparison the average velocity distribution for run 2 (see details in table 1) is also given in figure 3. The parameters for this simulation were the same as for run 1, except that $D_{\max} = 4$ Å was used, which is a large enough value not to limit the atomic displacement. Non-zero $g(r, t)$ values appeared only in the first time frame for the first r bin, $0 < r < 0.1$ Å, as in the

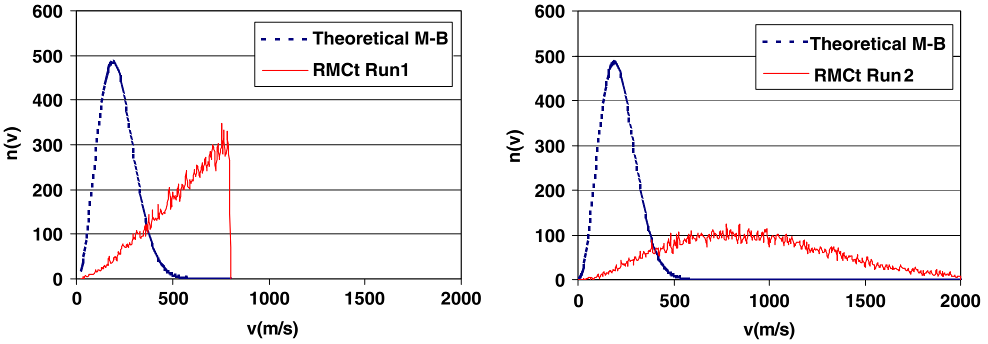

Figure 3. Final distribution of the average velocities for run 1 ($D_{\max} = 0.04$ Å) and run 2 ($D_{\max} = 4$ Å) which show the spreading of velocities through the available part of the first histogram bin of the (RMC*t*) calculated $g(r,t)$.

Table 1. Parameters of RMC*t* models, including data for the ensembles producing quasi-experimental data and the starting configurations.

		Quasi-experimental LJ MD (86.3 K, $\Delta t = 5$ fs)				LJ MD for starting ensemble (120 K, $\Delta t = 5$ fs)				
	Fitting	τ	W_{tframes}	N_{atoms}	dr (Å)	τ	W_{tframes}	N_{atoms}	dr (Å)	max_displ (Å)
Run 1	$g(r,t)$	50	40	452	0.10	50	40	452	0.10	0.04
Run 2	$g(r,t)$	50	40	452	0.10	50	40	452	0.10	4.00
Run 3	$g(r,t)$	50	40	452	0.03	50	40	452	0.03	0.40
Run 4	$g(r,t)$	50	40	452	0.01	50	40	452	0.01	0.40
Run 5	$S(Q,\omega)S(Q)$	100	80	2992	0.10	100	80	2992	0.10	0.04
Run 6	$g(r,t)$	50	40	452	0.10	50	40	452	0.10	0.04

'experimental data'. The distribution of the average velocities spreads along the whole available velocity range up to 2000 m s^{-1}.

To see how a higher resolution of both the histogram and experimental data (without additional constraints) affects the modelling, two more test runs were started (see table 1 for details). In run 3 dr = 0.03 Å and in run 4 dr = 0.01 Å were used both for the histogram calculation and the 'experimental' data, with $D_{\max} = 0.4$ Å. This did not put any effective constraint on the models as it corresponds to a maximum velocity of 8000 m s^{-1}. The RMC*t* models were stopped at relatively high χ^2, but the expected effect due to the finer histogram binning is easily visible (see figure 4).

These findings confirm that in the cases where the experimental data do not contain sufficiently detailed information about the short-range–short-time dynamics (which is the normal case) it is well advised to apply constraints on the velocity distribution. This suggestion is equally valid for modelling $g(r,t)$ or $S(Q,\omega)$. The velocity distribution constraint (calculated from a fine resolution distance distribution at $t = \Delta t$) may be viewed as applying very fine histogram binning only in the region where it is useful and required. In the case of fitting $S(Q,\omega)$ the maximum $Q_{\max}$ and $\omega_{\max}$ effectively determine the dr and Δt that should be used. Actually, this would be a severe problem in the case of most data sets where $Q_{\max}$ and $\omega_{\max}$ are related and constrained by the incident neutron energy. This illustrates even more the necessity to use additional constraints such as the velocity distribution or wide Q-range $S(Q)$ in RMC*t* modelling.

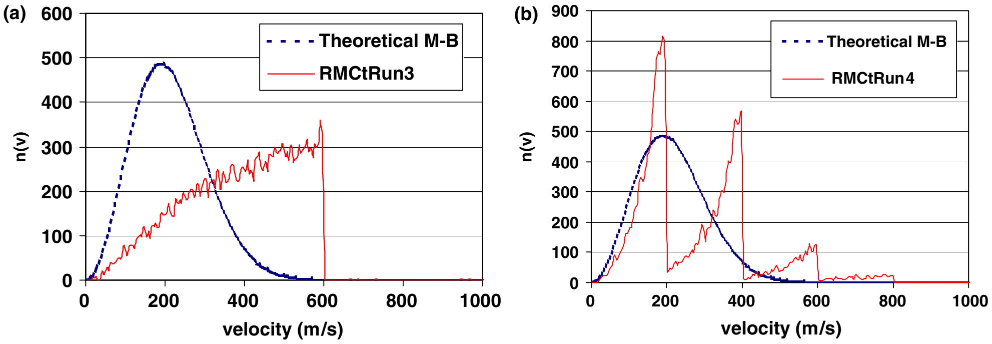

Figure 4. The final distribution of average velocities for run 3 (dr = 0.03 Å) and run 4 (dr = 0.01 Å). It is obvious that without an effective $D_{\max}$ constraint the smaller histogram bin size constrains the velocity distribution better.

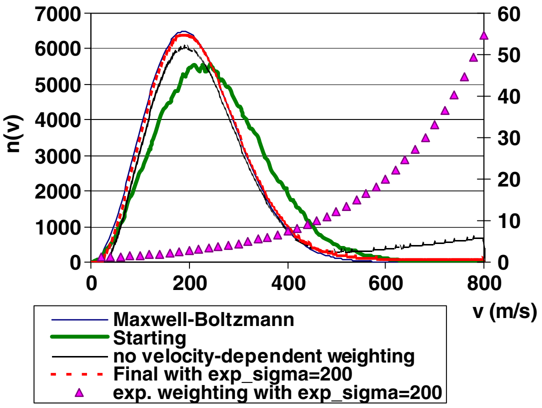

Figure 5. Velocity distribution histograms for $S(Q, \omega)$ fitting (run 5) that demonstrate the necessity of the velocity-dependent weighting. Triangles (right y axis) show the applied exponential weighting with $\sigma^{\exp} = 200$.

3.4. Velocity-dependent weighting

The application of the velocity distribution constraint defined by

$$\chi^2_{\text{vel}} = \sum_s^{n_{\text{vc}}} \frac{\sum_u^{n_{\text{vbins}}} \left(V^C_{s,u} - V^{\text{MB}}_{s,u}\right)^2 w_{s,u}}{\sigma_s^2} \tag{3}$$

(where $V^C_{s,u}$ is the calculated, $V^{\text{MB}}_{s,u}$ the theoretical velocity distribution histogram for the uth velocity bin of the sth velocity distribution constraint and σ_s is the control parameter) with $w_{s,u} = 1$ significantly improves the agreement between the velocity distributions of the model system and the theoretical distribution (see figure 5). However, there is a tendency for a 'tail' at higher velocities to remain in the model distribution. To decrease the tail, which results from the superposition of the moved distance distribution of the accepted moves on the existing average velocity distribution of the configuration ensemble, a velocity-dependent weighting factor $w_{s,u}$ was introduced during the χ^2 calculation. After some experimenting the following formula was adopted:

$$\begin{aligned} w_{s,u} &= \exp\left(\frac{v_u^{\text{mean}}}{\sigma_s^{\text{exp}}}\right) \qquad \text{if } \sigma_s^{\text{exp}} > 0; \\ w_{s,u} &= 1 \qquad \text{if } \sigma_s^{\text{exp}} \leqslant 0; \end{aligned} \tag{4}$$

where v_u^{mean} is the mean velocity of the uth velocity bin of the sth constraint.

An additional parameter σ^{exp} has therefore to be provided for each velocity distribution constraint. The lower the value of σ^{exp}, the more strongly differences at larger velocities contribute to the χ^2, making the moves contributing to the 'tail' less desirable.

As can be seen in figure 5, the tail has disappeared by the end of the modelling as a result of velocity-dependent weighting, resulting in virtually identical calculated and theoretical velocity distributions.

3.5. Connection between the maximum movement parameter and the distribution of average velocities

The course of the simulation, and the ratios of accepted, tried and generated moves, are strongly affected by the value of the allowed maximum atomic movements, i.e. the maximum distance an atom is allowed to move from its original position inside a single time frame. It is obvious that there will be a (complex) relationship between the maximum movement, the maximum displacement and the velocity distribution. If too large movements are tried then most of the moves are rejected, as the moved particle has a high probability of overlap with another particle. If, on the other hand, the moves are very small then the ratio of acceptance is higher, but the effective change is very small. The application of a well-chosen maximum movement value was therefore important during normal RMC modelling. With the introduction of dynamics (time dependence), and especially with the application of the average velocity distribution constraint, the choice of the maximum movement parameter in RMC*t* becomes even more critical.

During RMC modelling the sequence of configurations as they travel through the configuration space is not important, only the quality of the fit: χ^2 has to decrease according to the applied σ parameter. However, during the dynamic RMC*t* modelling it is not just the sequence of configurations (time frames) in the ensemble, representing the time evolution of the system, that is constrained by the experimental data, but also the way the configuration ensemble travels through the configuration space, that is constrained by the dynamical characteristics of the system. This is described below.

To change from one representation of the system (ensemble 1) to the next (ensemble 2), a randomly chosen particle (or particles) is (are) moved in a randomly chosen time frame. The distances of the moved particle to all the other particles in all the time frames is calculated, including the distance to the same particle in different time frames (self part). The change caused by the movement appears most markedly in the self part of the calculated data, particularly in its short-time part. Its effect on the average velocity distribution is extremely strong as the latter reflects the fine-resolution, small-distance range distribution. So the move itself is incorporated into the system and preserved in it, making the choice of an adequately chosen maximum movement crucial.

When the velocity distribution constraint is introduced a 0.1 Å maximum movement value results in almost all moves being rejected. The maximum displacement for this maximum movement corresponded to a velocity of 2000 m s^{-1}, whereas the experimental 86.3 K Lennard-Jones MD system has its maximum velocity around 600 m s^{-1}. So, to set a guideline for how an adequate maximum movement parameter can be chosen for a given system, we first have to see in more detail what distribution will arise from using a specific value of the maximum movement parameter.

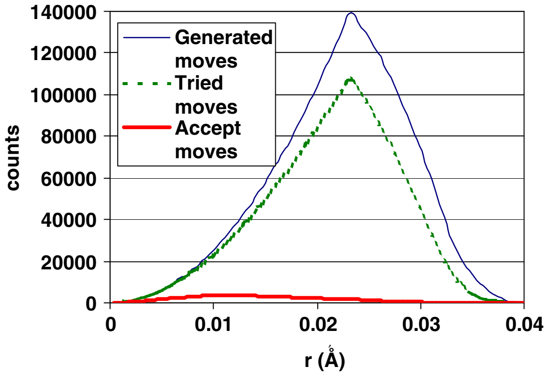

Figure 6. The distribution of generated, tried and accepted moves from the middle part of an RMC*t* run. The original $M_{\max} = 0.023$ Å value was still applied but the maximum of the distribution of accepted moves already moved to lower distances, indicating that it is advisable to decrease the value of the $M_{\max}$ parameter during the course of a run to achieve a higher acceptance ratio.

Moves are generated as a combination of separately generated random displacements in x, y, z directions up to a maximum moved distance $M_{\max}$, i.e., the move vectors are uniformly distributed in a cube with sides $M_{\max}$. The probability of a generated move to distance r is the part of the surface area of a sphere with radius r which is inside the cube. Therefore the generated moved distance distribution has a pointed maximum shape, reaching zero just beyond the largest possible maximum movement $\sqrt{3}M_{\max}$ Å (figure 6).

It has to be noted that the distance distribution of the accepted moves can have a different shape, though of course it must lie within the distribution of generated moves. The distribution of the generated, tried and accepted distances of the moves are saved during the simulation, as they can give information about choosing the right value for $M_{\max}$. Test simulations showed that at the beginning of a simulation, where the difference between the experimental and the calculated data is very large, the distance distribution of the accepted moves resembles very closely that of the generated moves. As χ^2 decreases it gets more and more difficult to find a move which decreases χ^2 further, so the shape of the accepted distance distribution changes: it becomes flatter and the maximum shifts towards lower distance values. Examples of the distributions of the generated, tried and accepted moves, taken from the middle part of an RMC*t* run, are shown in figure 6.

The theoretical maximum value of $M_{\max}$ allowed by the position of an atom and its images in the consecutive configurations for a given $D_{\max}$ value is $M_{\max} = 2D_{\max}/\sqrt{3}$. Such large moves are, however, extremely rare.

The following considerations may help in choosing a maximum movement parameter:

- When there is a big difference between the calculated and the experimental data the recommended value of $M_{\max}$ can be determined as $M_{\max} = D_{\max}/\sqrt{3}$.
- During modelling it is recommended to check the saved 'distance of the moves statistics' and decrease the applied value of the $M_{\max}$ parameter accordingly. In principle it would be possible to provide a default automation of this choice within the program.

If the average velocity distribution constraint is applied, the σ (and logarithmic sigma) parameter(s) is (are) advised to be chosen to give a χ^2 contribution 1–2 orders of magnitude smaller then the χ^2 coming from the experimental data. Such a choice helps to find the appropriate configuration ensemble but it does not put too strict a constraint in itself on the

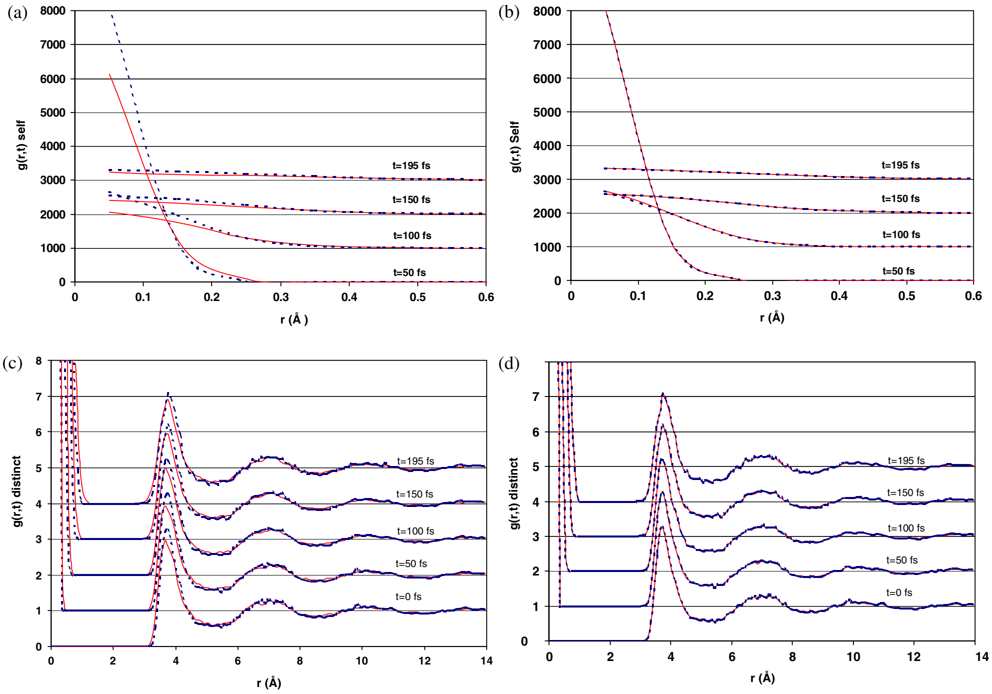

Figure 7. Values of the starting and final calculated (solid) and 'experimental' (dotted) $g(r, t)$ at $t = 0$, 50, 100, 150 and 195 fs for run 6. (a) $g(r, t)$ is shown for $0 < r < 0.6$ Å for the starting ensemble to emphasize details coming from the self part. (b) As for (a) for the final ensemble. (c) $g(r, t)$ is shown for $0 < r < 14$ Å for the starting ensemble to emphasize details coming from the distinct part; (d) same as (c) for the final ensemble. The series belonging to different time values are displaced up the y-axis for clarity.

modelling. Too strict a velocity distribution constraint prevents the rearrangements of the atoms and the model becomes 'frozen'. When the difference between calculated and experimental data ($g(r, t)$, $S(Q, \omega)$) has decreased considerably then it is advisable to decrease the σ for the velocity distribution constraint, in order to achieve as good a fit for the velocity distribution as possible.

4. Results: tests with model ('quasi-experimental') data

During these tests the quasi-experimental data were calculated from a Lennard-Jones MD configuration ensemble. The starting configuration ensemble also came from a similar MD simulation; however, the temperatures of the two systems were different in order to make the quasi experimental and initial model $g(r, t)$ or $S(Q, \omega)$ functions different. Parameters of the models are given in table 1.

4.1. Modelling $g(r, t)$

This test refers to run 6. Due to the two independent variables (r, t) in $g(r, t)$ it is difficult to show clearly both the full calculated and the 'experimental' data on the same figure. We have instead shown data at only a fixed set of time values in figure 7.

As can be seen, the huge initial difference virtually disappears during the modelling run. At first acceptance was dominated by χ^2_{lg}, then, after a substantial decrease of the difference

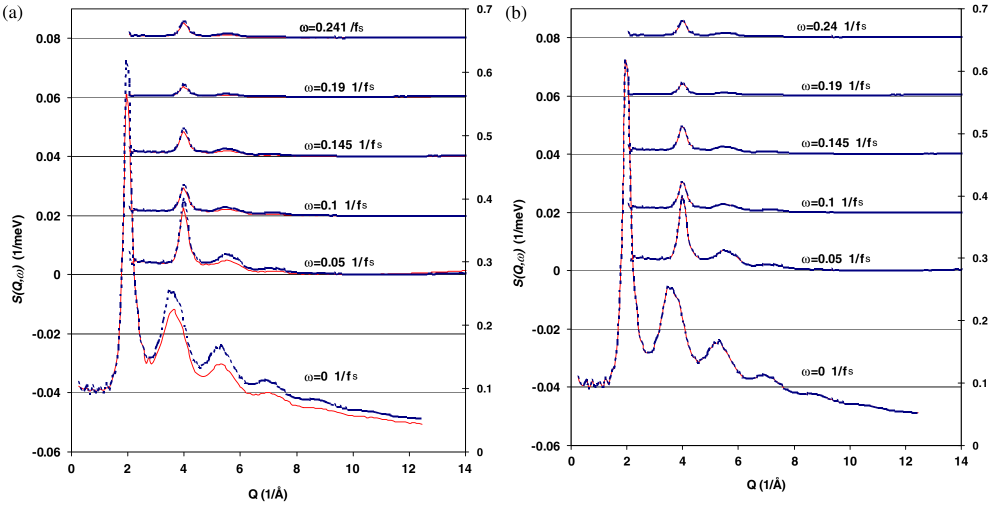

Figure 8. Values of the starting and final calculated (red solid) and 'experimental' (blue dotted) total $S(Q, \omega)$ at $\omega = 0, 0.05, 0.1, 0.145, 0.19$ and 0.24 fs^{-1} for run 5. (a) The calculated and experimental $S(Q, \omega)$ for the starting ensemble. (b) The final calculated and experimental $S(Q, \omega)$. The series for different frequencies are displaced up the y-axis for clarity.

Table 2. Numerical values for the standard and the logarithmic χ^2 for run 6 (modelling $g(r, t)$).

χ^2 given for $\sigma = 1$	$\chi^2 g(r, t)$	$\chi^2_{\text{lg}} g(r, t)$
Starting	4.5×10^7	13.59
Final	1.2	0.06

coming from the self part, the standard χ^2 was applied for $g(r, t)$. A velocity distribution constraint was also used during the run.

The initial and the final χ^2 for the $g(r, t)$ data set are given in table 2.

4.2. Modelling $S(Q, \omega)$

Similarly to modelling $g(r, t)$, both the ensembles producing the 'experimental' $S(Q, \omega)$ and the starting configurations came from MD configurations; details are given in table 1. The run code for this study was run 5.

Even with this increased system size, correlations in time could not decay entirely within the simulated ensemble. As truncation errors resulting from this shortcoming affect both the 'experimental' and the RMC$t\,S(Q, \omega)$ the same way, no serious inconsistencies were found. However, in the case of real experimental data the finite (r, t) model size would have to be taken into account by convoluting the experimental data with an appropriate function.

To accelerate convergence, static $S(Q)$ fitting was also applied. The 'experimental' $S(Q)$ was calculated using the same ensemble as for the 'experimental' $S(Q, \omega)$. A velocity distribution constraint was applied with $\sigma_{\text{exp}} = 200$ velocity-dependent weighting at the final stage of the simulation; the result of this on the velocity distribution was shown earlier (figure 5).

It can be seen that the initial difference between data sets decreased substantially during the simulation, as is also apparent from table 3. Comparison of the $S(Q, \omega)$ values at a set of ω is made in figures 8, and 9 compares the 'experimental' and model (static) $S(Q)$.

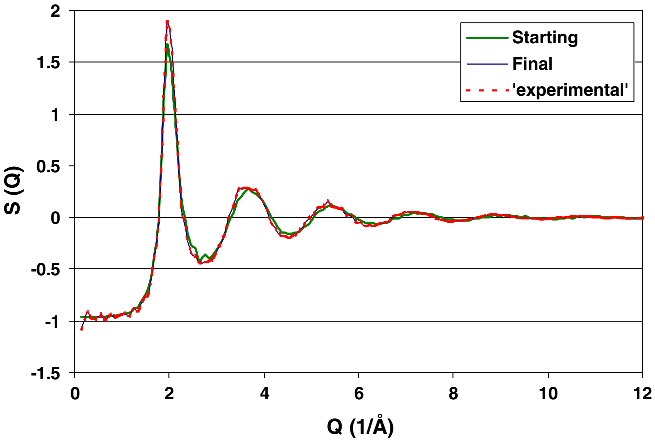

Figure 9. Comparison of static $S(Q)$ for the starting and final ensemble of run 5.

Table 3. The change in χ^2 for run 5 ($S(Q,\omega)$ modelling).

χ^2 given for $\sigma = 1$	$\chi^2 S(Q,\omega)$	$\chi^2 S(Q)$	χ^2 velocity distribution	Number of generated moves	Number of accepted moves
Starting	6.22×10^{-2}	4.51×10^{-1}	9.26×10^{7}	0	0
Final	2.10×10^{-8}	1.43×10^{-8}	2.05×10^{6}	3.7×10^{8}	6.7×10^{6}

4.3. Calculation of dynamic properties

Some properties related to the dynamics of the system have been calculated for the starting and the final configuration ensembles of run 5, as well as for the MD system providing the ‘experimental’ data. One of these properties was the squared displacement (SD) for each atom between the first and the last (100th) time frames in the ensembles. The atoms were sorted (in increasing order) according to their SDs to make the comparison possible. As is visible from figure 10(a), the larger displacements of the starting (higher-temperature) system have decreased during the RMC*t* calculation and the final SDs are running together with the SDs of the 86.3 K ‘experimental’ system. The good agreement may be explained by the fact that the SD is strongly determined by $S(Q,\omega)$, through $g(r,t)$.

The (initial parts of the) velocity autocorrelation function (VACF) [9, 12], a quantity that is frequently quoted in molecular dynamics studies [9], could be calculated, too. Note that this calculation can only serve as a demonstration, due to the rather small system size (in terms of the time variable; up to 395 fs). Results are shown in figure 10(b). The agreement between RMC*t* and ‘experimental’ functions may be termed as semi-quantitative. The first point of the VACF, corresponding to $t = 0$ fs, is strongly constrained by the velocity distribution constraint: it moved from the higher starting value closer to the lower ‘experimental’ value. The $t > 0$ VACF values are only weakly constrained by the velocity distribution constraint (mostly through $D_{\max}$) and by $S(Q,\omega)$ (due to the missing detailed information, as discussed in sections 3.2–3.3). For a thorough comparison (and analysis), much longer calculations (requiring much increased computational resources) would be necessary.

5. Conclusions

A new method for modelling atomic dynamics based on experimental data, RMC*t*, has been presented. This has so far been shown to work for ‘ideal’ data. We have learnt that control of the

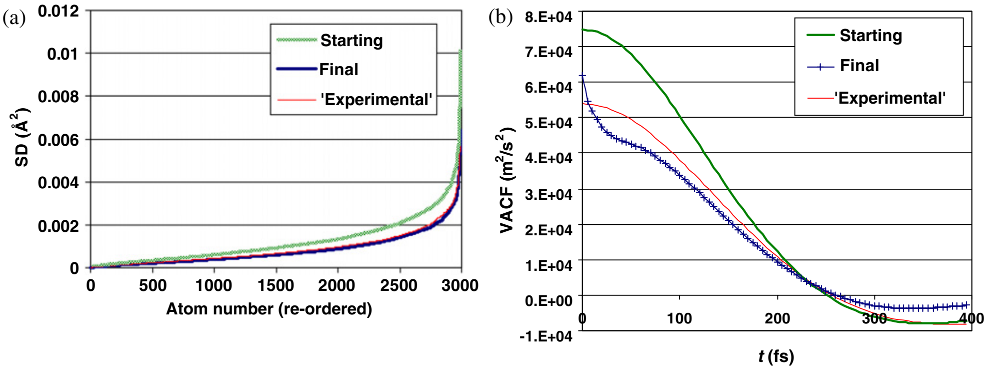

Figure 10. (a) Squared displacement of each atom (sorted, in increasing order, according to their displacements) between the first and the last time frames in the starting and final configuration ensembles of run 5 and of the 'experimental' ensemble. (b) The VACF for the starting and final stages of run 5, as well as for the 'experimental' system.

velocity distribution (related to the temperature) is necessary via the application of a suitable constraint. This can be traced to a lack of detailed information in the data at short times and distances. As could be expected, the method is considerably more computationally demanding than RMC structural modelling, so efficient coding using parallelization is necessary if the model sizes and timescales are to be increased to those necessary for modelling real experimental data. Possible routes to efficient parallel codes are described in appendix C.

Appendix A. Theoretical background

Theoretical background can be found elsewhere [11, 12] in detail; here, only the necessary pieces are mentioned.

The *dynamic structure factor* $S(\mathbf{Q},\omega)$ can be obtained from time-of-flight neutron scattering experiments [13], where not only the momentum transfer ($\mathbf{Q}\hbar$), but the energy transfer ($\omega\hbar$) is detected, as well. During the experiment the partial differential cross section $\left(\frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega \mathrm{d}E'}\right)$ is measured.

If not all the nuclei have the same scattering properties due to the presence of different isotopes and/or different spin states, which can happen even in a one-component system, incoherent scattering arises. The partial differential cross section can then be separated into coherent and incoherent parts, which are proportional to the $S(\mathbf{Q},\omega)$ and its incoherent (or self) part $S^{\mathrm{inc}}(\mathbf{Q},\omega)$, and the relationship is given for a mono-atomic system by

$$\left(\frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega\,\mathrm{d}E'}\right) = \frac{k'}{k}\frac{N}{4\pi}\left(\sigma_c S(\mathbf{Q},\omega) + \sigma_{\mathrm{inc}} S^{\mathrm{inc}}(\mathbf{Q},\omega)\right) = \frac{k'}{k}\frac{N}{4\pi} S_{\mathrm{T}}(\mathbf{Q},\omega) \tag{5}$$

where k and k' are the magnitudes of the incident and scattered wavevector, respectively, and N is the number of atoms. The scattering vector can be given by $\mathbf{Q} = \mathbf{k} - \mathbf{k}'$. The coherent and incoherent cross sections σ_{c} and σ_{inc} are related to the bound scattering lengths b according to the following formulae:

$$\sigma_c = 4\pi \left|\bar{b}\right|^2 \qquad \sigma_{\mathrm{inc}} = 4\pi \left[\overline{|b|^2} - \left|\bar{b}\right|^2\right]. \tag{6}$$

The total *dynamic structure factor* $S_{\mathrm{T}}(\mathbf{Q},\omega)$ is the inverse Fourier transform in space and the Fourier transform in time of the *spin- and isotope-dependent correlation function* $\Gamma(\mathbf{r},t)$, which is related to the self and distinct parts of the *time-dependent correlation function* (*van Hove correlation function*) [14], $G(\mathbf{r},t) = G^{\mathrm{s}}(\mathbf{r},t) + G^{\mathrm{d}}(\mathbf{r},t)$, and the relationship can be expressed for a multi-component system in the following way:

$$\Gamma\left(\mathbf{r},t\right) = \sum_{\alpha\beta} c_\alpha \overline{b}_\alpha \overline{b}_\beta G^{d}_{\alpha\beta}(\mathbf{r},t) + \sum_{\alpha} c_\alpha \overline{b^2}_\alpha G^{s}_{\alpha}(\mathbf{r},t) \tag{7}$$

where α and β denote the components of the system.

The Fourier transformation relationship between $S(\mathbf{Q},\omega)$ and $G(\mathbf{r},t)$ and their incoherent part given by equation (8) is the theoretical basis of RMC*t*.

$$\begin{aligned} S\left(\mathbf{Q},\omega\right) &= \frac{1}{2\pi\hbar}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G\left(\mathbf{r},t\right) \mathrm{e}^{\mathrm{i}(\mathbf{Q}r-\omega t)}\mathrm{d}\mathbf{r}\,\mathrm{d}t; \\ S^{\mathrm{inc}}\left(\mathbf{Q},\omega\right) &= \frac{1}{2\pi\hbar}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G^{s}\left(\mathbf{r},t\right) \mathrm{e}^{\mathrm{i}(\mathbf{Q}r-\omega t)}\mathrm{d}\mathbf{r}\,\mathrm{d}t. \end{aligned} \tag{8}$$

It has to be noted that $G(\mathbf{r},t)$ is a density distribution function, and the dimension of $G(\mathbf{r},t)$ is (volume)$^{-1}$, whereas of $S(\mathbf{Q},\omega$ it is (energy)$^{-1}$.

For a disordered material, with radial symmetry around particles, the integration over the angular components of the vector **r** can be carried out explicitly, leaving just the magnitudes, r and Q.

For practical reasons, not the $G(r,t)$ but the particle correlation function-type pair correlation function is used during the simulation. Extending the definition of the static pair correlation function, the *dynamic pair correlation function*, $g(r,t)$, for a multi-component system may be defined in the following way:

$$\begin{aligned} g^{d}_{\alpha\beta}(r,t) &= \frac{P_{\alpha\beta}(r,t)}{P^{H}_{\alpha\beta}(r,t)} = \frac{G^{\mathrm{d}}_{\alpha\beta}(r,t)}{\rho^{0}_{\beta}(r,t)}, \qquad \alpha \geqslant \beta; \\ g^{s}_{\alpha}(r,t) &= \frac{P_{\alpha}(r,t)}{P^{H}_{\alpha}(r,t)} = \frac{G^{s}_{\alpha}(r,t)}{\rho^{0}_{\beta}(r,t)} \end{aligned} \tag{9}$$

where, for the distinct partial, $P_{\alpha\beta}(r,t)$ is the probability of finding a particle type β at time t in volume element $\mathrm{d}V$ at a distance r from a particle type α at the origin at time $t = 0$; $P^{H}_{\alpha\beta}$ is the same probability in a homogenous material. For the self partial the probabilities can be interpreted similarly.

$g(r,t)$ can be calculated from the positions; the exact method of this computation is described in appendix B.2.

For a multi-component system the total time-, spin- and isotope-dependent particle correlation function based on the Faber–Ziman formalism can be derived as

$$g_{\mathrm{T}}\left(r,t\right) = \frac{\Gamma(r,t)}{\rho^{0}} = \sum_{\alpha\beta} c_\alpha c_\beta \overline{b}_\alpha \overline{b}_\beta g^{d}_{\alpha\beta}(r,t) + \sum_{\alpha} c^{2}_{\alpha} \overline{b^2}_\alpha g^{s}_{\alpha}(r,t). \tag{10}$$

It is visible that the first sum describing the contributions coming from the distinct part would be the same for $t = 0$ after normalization as $g^{\mathrm{RMC}}_{\mathrm{T}}(r)$.

Rearranging equation (10) to be more suitable for computation gives

$$\begin{aligned} g_{\mathrm{T}}\left(r,t\right) = {} & \sum_{\alpha} c^{2}_{\alpha}\left(\overline{b}_\alpha\right)^{2} g_{\alpha\alpha}\left(r,t\right) + \sum_{\alpha}\sum_{\beta<\alpha} 2c_\alpha c_\beta \overline{b}_\alpha \overline{b}_\beta g^{d}_{\alpha\beta}\left(r,t\right) \\ & + \sum_{\alpha} c^{2}_{\alpha}\left[\overline{b^2}_\alpha - \left(\overline{b}_\alpha\right)^{2}\right] g^{s}_{\alpha}\left(r,t\right) \end{aligned} \tag{11}$$

where

$$g_{\alpha\alpha}\left(r,t\right) = g^{s}_{\alpha}\left(r,t\right) + g^{d}_{\alpha\alpha}\left(r,t\right). \tag{12}$$

Dependency on the nuclei means that each chemical element, their different isotopes and their different spin states possess different scattering properties, so their contributions to the total scattering differ and they have to be weighted according to the different scattering lengths.

In RMCt, $g_{\mathrm{T}}(r,t)$ and $S_{\mathrm{T}}(Q,\omega)$ are renormalized to remove the dependence of their values from the actual values of the scattering lengths, leaving just the proportion of the contributions of the different partials. This ensures that the value of the $g(r,t)$ will tend to one for larger r and t values:

$$g_{\mathrm{T}}^{\mathrm{RMC}t}(r,t)=\sum_{\alpha}\kappa_{\alpha\alpha}^{\mathrm{coh}}g_{\alpha\alpha}^{\mathrm{RMC}t}(r,t)+\sum_{\alpha}\sum_{\beta<\alpha}\kappa_{\alpha\beta}^{\mathrm{coh}}g_{\alpha\beta}^{\mathrm{RMC}t,d}(r,t)+\sum_{\alpha}\kappa_{\alpha}^{\mathrm{inc}}g_{\alpha}^{\mathrm{RMCT},s}(r,t). \tag{13}$$

The following expressions are applied during the renormalization:

$$\begin{aligned}&\kappa_{\alpha\alpha}^{\mathrm{coh}}=c_{\alpha}^{2}\overline{b}_{\alpha}^{2}, &&\kappa_{\alpha\beta}^{\mathrm{coh}}=2c_{\alpha}c_{\beta}\overline{b}_{\alpha}\overline{b}_{\beta} &&\alpha\neq\beta, &&\kappa_{\alpha}^{\mathrm{inc}}=c_{\alpha}^{2}\left[\overline{b_{\alpha}^{2}}-\overline{b}_{\alpha}^{2}\right]\\ &\Xi=\sum_{\alpha}\sum_{\beta\leqslant\alpha}\kappa_{\alpha\beta}^{\mathrm{coh}} &&\kappa_{\alpha\beta}^{\mathrm{coh}}=\frac{\kappa_{\alpha\beta}^{\mathrm{coh}}}{\Xi}, &&a\leqslant\beta, &&\kappa_{\alpha}^{\mathrm{inc}}=\frac{\kappa_{\alpha}^{\mathrm{inc}}}{\Xi}.\end{aligned} \tag{14}$$

The 'RMCt' index indicates that the definition of the total $g(r,t)$ given by equation (13) is developed especially for RMCt, where $g_{\alpha}^{\mathrm{RMC}t,s}(r,t)$ is defined only for $t>0$. $g_{\mathrm{T}}^{\mathrm{RMC}t}(r,0)$ does not contain the contribution coming from the self part, which is handled separately.

As the double Fourier transformation is a time consuming process, instead of the transformation of the partial $g_{\alpha\beta}(r,t)$ into partial $S_{\alpha\beta}(Q,\omega)$ and their weighted linear combination to give $S_{\mathrm{T}}(Q,\omega)$, the total $g_{\mathrm{T}}^{\mathrm{RMC}t}(r,t)$ is transformed directly to $S_{\mathrm{T}}^{\mathrm{RMC}t}(Q,\omega)$ in two steps. The calculation is performed according to equation (15) after the discretization of the integral and carrying out the integration over the discrete intervals:

$$\begin{aligned}S_{\mathrm{T}}^{\mathrm{RMC}t}(Q_m,\omega_n)={}&\frac{4\pi\rho^{0}}{Q_m^{2}}\frac{1}{\pi\hbar\omega_n}\sum_{q=0}^{W_{\mathrm{tframes}}-1}\left(\sum_{l=0}^{l_{\max}}\left[g_{\mathrm{T}}^{\mathrm{RMC}t}(r_l,t_q)-1\right]\left[\frac{\sin(Q_m r)}{Q_m}\right.\right.\\&\left.\left.-r\cos(Q_m r)\right]_{r_l}^{r_{l+1}}\right)\left[\sin(\omega_n t)\right]_{q\Delta t}^{(q+1)\Delta t}+\frac{1}{\pi\hbar\omega_n}\frac{\sum_{\alpha}c_{\alpha}\overline{b_{\alpha}^{2}}}{\left(\sum_{\alpha}c_{\alpha}\overline{b}_{\alpha}\right)^{2}}\sin(\omega_n\Delta t)\end{aligned} \tag{15}$$

where m is the index of the Q points ($0\leqslant m<\Lambda$); n is the index of the ω points ($0\leqslant n<\Omega$); l is the index of the r points ($0\leqslant l\leqslant$ number of $r$points -1); and q is the index of the t points ($0\leqslant q<W_{\mathrm{tframes}}$).

First the $r\rightarrow Q$ transformation of $[g_{\mathrm{T}}^{\mathrm{RMC}t}(r,t)-1]$ is performed by the calculation of the inner sum in the round bracket which, including the first fraction of the constant, gives $S(Q,t)$ (sometimes denoted as $I(Q,t)$), the intermediate function. (To obtain the intermediate function is the reason for π appearing in the numerator of the first fraction and the denominator of the second.) It is worth mentioning that $S(Q,t=0)=S(Q)$. $S(Q,t)$ is further transformed according to the rest of equation (15) to yield the total dynamic structure factor. It has to be noted that $g_{\mathrm{T}}^{\mathrm{RMC}t}(r,t)$ and, consequently, $S(Q,t)$ does not contain the contribution coming from the self $g_{\alpha}^{s}(r,0)$. As the delta function corresponding to $G_{\alpha}^{\mathrm{RMC}t,s}(r,0)$ can be explicitly integrated over r giving $+1$, the total contribution coming from all the particle types is $\sum_{\alpha}c_{\alpha}\overline{b_{\alpha}^{2}}/(\sum_{\alpha}c_{\alpha}\overline{b}_{\alpha})^{2}$, which is added to $S(Q,t=0)$ before it is transformed to $S(Q,\omega)$. This contribution is the last part of equation (15).

The discrete interval has to be small enough, in order to give an appropriate estimate for the integral. Even for the largest frequencies $Q_{\max}$ and $\omega_{\max}$ there have to be at least five data points over a complete wave to ensure adequate sampling; that is, the following relations have to be fulfilled:

$$\Delta r\leqslant\frac{\pi}{\gamma Q_{\max}},\qquad\Delta t\leqslant\frac{\pi}{\varepsilon\omega_{\max}} \tag{16}$$

where γ is the measure of the oversampling, the number of discretization points of r for a full wave (2π) in case of the $r \rightarrow Q$ transformation; ε is the same for t in case of the $t \rightarrow \omega$ transformation.

In order to decrease the time requirements of the calculation, it is advisable to build two two-dimensional matrices, one depending only on the (r, Q) values, the other on the (t, ω) values. These matrices have to be calculated only once, at the beginning of the simulation. The speed of the integral calculation in every step could be further increased, if one, (r, Q, t, ω) four-dimensional matrix could be used instead of the two two-dimensional ones. However, the memory requirements (minimum around $200 \times 100 \times 200 \times 100 = 400\,000\,000$ matrix element/data set) would make it impossible to run the program on a normal PC.

Although the partial $S_{\alpha\beta}(Q, \omega)$ functions are not used directly in the iteration process, they are calculated from time to time similarly to equation (16), only using the *partial pair correlation functions*.

Appendix B. Calculation details

B.1. Histogram calculation

At the beginning of the simulation all of the interatomic distances have to be calculated, both within the configurations and between them. During the simulation the distance of the moved atom to all of the other atoms in the same and different configurations has to be calculated.

For the simulation of τ time frames we need τ simulation cells. The time difference between consecutive configurations is Δt; the full (length of τ) trajectory encompasses $t = (\tau - 1)\Delta t$ from the system's lifetime. For the calculation of the $q\Delta t$ time difference we would have $\tau - q$ configuration pairs $(0 \leqslant q < \tau)$. It can be seen that the larger the time difference is, the smaller the number of available configuration pairs is. For the greatest time difference $(\tau - 1)\Delta t$ there is only one configuration pair left (consisting of the first and the last configuration of the set). This problem could be solved by applying systems consisting of a few thousand particles/configuration, but then the duration of the calculation and the memory usage would be too high.

As, for the time being, the number of atoms in a simulation cell will be relatively low (around 500), the statistics of the greater time differences will be poor.

To make the statistics of the smaller and larger time differences balanced, and to avoid the use of data with poor statistics, the concept of a calculation window consisting of W_{tframes} time frames, sliding through all the τ time frames of the configuration ensemble was introduced. Now the simulation of the system trajectory goes up to the $t_{\max} = (W_{\mathrm{tframes}} - 1)\Delta t$ largest time difference, not using the largest time differences with the poorest statistics, which could impair the quality of the $S(Q, \omega)$ data in each data point due to the double Fourier transformation. The size of the calculation window is a parameter; the maximum of it is the number of configurations used by the simulation: $W_{\mathrm{tframes}} \leqslant \tau$.

Figure B.1 shows how the calculation window consisting of six time frames slides through the configurations of a $\tau = 10$ configuration ensemble. For the configurations with lower indices the entire calculation window is available (denoted by shading). No shading is applied when only a part of the calculation window could be used; this happens when the starting point was a higher-index configuration.

If the histogram were calculated only in the shaded entire calculation window then the statistics for each time difference would be the same: each is calculated five times in our example. In this case, however, the higher-index configuration would contribute to a smaller extent to the calculation, with the result that a movement of an atom in configurations 6–10 would not be 'felt' by the system as strongly, since the small time difference(s) is (are)

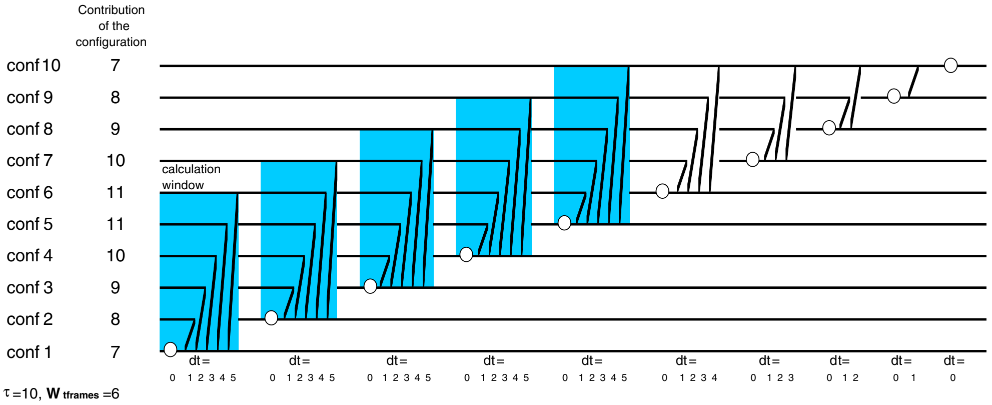

Figure B.1. The way the calculation window slides through the configurations for a $\tau = 10$ configuration ensemble with $W_{\text{tframes}} = 6$. The circles denote the d$t = 0$ configuration pairs (the two configurations are the same), the tilted lines connect the configurations of a configuration pair for d$t > 0$. The contribution of the configuration means how many times a configuration is involved in building a configuration pair. Any configuration is counted twice at forming the d$t = 0$ configuration pair. The time difference between the configurations (time frames) is denoted as d$t = 0, 1, 2\ldots$ meaning $t = 0$, $t = \Delta t$, $t = 2\Delta t$ and so on. Shading denotes the full calculation window. No shading means that only part of the calculation window could be used in case of a higher-index starting configuration.

not calculated for these configurations. As our aim was to provide better statistics for the greater time differences, and not to make the statistics equal, the calculation is performed for all the partial calculation windows, as well. With this method the contribution of the low- and high-index configurations pairwise are the same, introducing a kind of symmetry into the system: an atomic movement made in a low- or high-index configuration is 'felt' equally by the system. (The contribution of the configuration with the lowest index equals the highest-index configuration, the lowest index + 1 the highest index − 1 and so on, as it is shown by the 'Contribution of the configuration' in figure B.1.)

This means that, for example, for $W_{\text{tframes}} = 6$, all the available configuration pairs making time differences up to $t_{\max} = 5\Delta t$ are used in the calculation of the histogram. If the size of the calculation window is changed for the same configuration ensemble, it only means that $t_{\max}$ changes. The total number of configuration pairs is $\sum_{q=0}^{W_{\text{tframes}}-1}(\tau - q)$.

The program is capable of handling simulations with $3 \leqslant W_{\text{tframes}} \leqslant \tau$ time frames; usually $W_{\text{tframes}} = 0.8\tau$ was applied so far.

During the calculation the—otherwise—continuous variable distance will be discretized. The discrete interval for the histogram is s, which has the dimension of distance; the number of histogram bins is n_{bin}, covering the sn_{bin} maximum distance between atoms. This maximum distance is most usually set to $L/2$; however, it may be larger, with a maximum of $\sqrt{3}L/2$. During the histogram calculation the number of distances falling between $r \rightarrow r + s$ will make up the counts of the histogram bin $l = (r + s)/s(1 \leqslant l \leqslant n_{\text{bin}})$. In a (static) RMC simulation of N particles with n_{moved} moved particles, this will make up $2(n_{\text{moved}}N - \sum_{i=1}^{n_{\text{moved}}} i)$ distances between the moved particle(s) and the others (N is the number of atoms in the simulation cell). In an RMCt simulation this figure will be multiplied by a parameter depending on τ and W_{tframes} and the t value of the configuration containing the moved particle(s). (Multiplication by two comes from the fact that both the new and old distances of the moved atom(s) from the others have to be computed.)

Concerning the histogram, another new feature will appear in the case of RMCt, beside the appearance of the time dimension. During RMC modelling of multi-component systems two kinds of histogram exist:

- The $H_{\alpha\alpha}$ type 'clean' partial contains the counts of distances between particles from the same type in the configuration. Here the distance of a particle to itself obviously has no meaning.
- The $H_{\alpha\beta}$ type 'mixed' partial contains the counts of distances between particles from different types in the configuration.

The above-mentioned two types of histogram will be referred to as 'distinct' in the case of RMCt, and denoted as $H^{\mathrm{d}}_{\alpha\alpha}$ and $H^{\mathrm{d}}_{\alpha\beta}$.

In RMCt we have to calculate the distance of the same particle in two different configurations. These counts will be kept in the 'self' type histogram, denoted as $H^{\mathrm{s}}_{\alpha\alpha}$.

B.2. Calculation of the dynamic pair correlation function

Based on equation (9), the calculation of $g(r,t)$ is defined the following way for the RMCt simulation. This definition ensures that the dynamic pair correlation function can be divided into distinct and self parts, similarly to the *van Hove correlation function*: $g(r,t) = g^{\mathrm{d}}(r,t) + g^{\mathrm{s}}(r,t)$.

$$g_{\alpha\beta}^{\mathrm{RMC}t,d}(r,t) = n_{\alpha\beta t}\frac{1}{n_{\mathrm{cp}}t_f^0 N_\alpha N_\beta}\frac{V_{\mathrm{cell}}}{V(r)} \qquad \begin{array}{ll} t_f^0 = 1 & \text{if } q = 0 \\ t_f^0 = 2 & \text{if } q > 0 \end{array} \tag{17}$$

$$g_{\alpha\alpha}^{\mathrm{RMC}t,d}(r,t) = n^d_{\alpha\alpha t}\frac{1}{n_{\mathrm{cp}}t_f^0 N_\alpha^2}\frac{V_{\mathrm{cell}}}{V(r)} \qquad \begin{array}{ll} t_f^0 = \frac{1}{2} & \text{if } q = 0 \\ t_f^0 = 1 & \text{if } q > 0 \end{array} \tag{18}$$

$$g_{\alpha}^{\mathrm{RMC}t,s}(r,t) = n^s_{\alpha\alpha t}\frac{1}{n_{\mathrm{cp}} N_\alpha^2}\frac{V_{\mathrm{cell}}}{V(r)} \qquad q > 0$$

$$\begin{array}{llll} t = q\Delta t, & n_{\mathrm{cp}} = \tau - q; & \text{for } g_{\alpha\alpha}^{\mathrm{RMC}t,d} \quad \text{and} & g_{\alpha\beta}^{\mathrm{RMC}t,d} \\ & 0 \leqslant q < W_{\mathrm{tframes}} & & \end{array} \tag{19}$$

where $n_{\alpha\beta t}$, $n^{\mathrm{d}}_{\alpha\alpha t}$ és $n^{\mathrm{s}}_{\alpha\alpha t}$ are the counts in the appropriate histograms, and q is the index of the time frame. The factor n_{cp} is the actual number of available configuration pairs for the calculation of the $q\Delta t$ time difference. The other factor, t_f^0, is adjusting ('normalizing') the number of atom pairs used in the calculation of the 'distinct' histograms to the theoretical $N_\alpha N_\beta$ pairs applied for the calculation for the ideally homogenous material.

The discrete interval of the partial pair correlation function does not have to coincide with the s interval of the histogram; usually the relation is $s \leqslant \mathrm{d}r$. That is, the dimensions of the arrays are not necessary the same.

First the partial pair correlation functions are calculated at the histogram grid points ($r = s$), so it should be denoted as $g(s,t)$. In the case of $g(r,t)$ fitting, first all the existing partial $g(r,t)$ are calculated at the experimental r data points from $g(s,t)$; then, from them the total $g_{\mathrm{T}}(r,t)$ is determined.

In the case of $S(Q,\omega)$ fitting, the partial $g(s,t)$ are combined to yield $g_{\mathrm{T}}(s,t)$, which is transformed into $S(Q,\omega)$.

B.3. Comparison of calculated and experimental data

For the comparison of calculated and experimental data it is necessary to have them at the same data points, which the program automatically ensures. The quantitative measure of the difference, χ^2, is calculated according to the following formula for the $g(r,t)$, $S(Q,\omega)$ and $S(Q)$ data series, respectively:

$$\chi^2 = \sum_{i}^{n_{\exp t}} \frac{\sum_j \left(a_i A_{i,j}^C + b_i - A_{i,j}^E\right)^2}{\sigma_i^2} + \sum_{m}^{n_{\mathrm{cc}}} \frac{\sum_p^\tau \left(\frac{N_{m,p}^s}{N_m^{\mathrm{c}}} - N_m^f\right)^2}{\sigma_m^2}$$

$$+ \sum_{n}^{n_{\mathrm{ac}}} \frac{\sum_p^\tau \left(\frac{N_{n,p}^{\mathrm{nc}}}{N_n^{\mathrm{c}}} - N_n^d\right)^2}{\sigma_n^2} + \sum_{s}^{n_{\mathrm{vc}}} \frac{\sum_u^{n_{\mathrm{vbins}}} \left(V_{s,u}^C - V_{s,u}^{\mathrm{MB}}\right)^2 w_{s,u}}{\sigma_s^2} \qquad (20)$$

where i is the index of the experimental data sets ($n_{\mathrm{expt}} = n_{grt} + n_{sqo} + n_{sq}$), j is the index of the data points, and $A_{i,j}^C$ and $A_{i,j}^E$ are the jth data point of the calculated and experimental ith data series. It can be seen that a (small!) renormalization of the experimental data can also be done according to equation (20) (a_i, b_i given by equations (21)–(23)). Renormalization is optional in the program. The contribution of the coordination number constraint to the χ^2 is given by the second sum and that of the average coordination number constraint by the third sum of equation (20). (n_{cc}: number of coordination number constraints; n_{ac}: number of average coordination number constraint;n_{vc}: number of velocity distribution constraints; $N_{m,p}^s$: number of central atoms for the mth coordination number constraint in the pth configuration, whose coordination number equals the desired coordination number; N_m^{c}: number of central atoms for the mth coordination or average coordination number constraint; N_m^{f}: fraction of central atoms for the mth coordination number constraint for which the constraint has to be fulfilled; $N_{n,p}^{\mathrm{nc}}$: total number of neighbours for all the central atoms of the nth average coordination number constraint in the pth configuration; N_n^d: desired average coordination number for the nth average coordination number constraint; σ is the weighting factor of the given constraint).

The last sum of this equation described earlier by equation (3) gives the contribution coming from the velocity distribution constraint. The reason for using the velocity histogram counts instead of the velocity distribution was to spare the time of calculating the velocity distribution from the histogram counts at each simulation step (by normalizing it with the number of atoms of the constrained type (N_α), the number of $t = \Delta t$ time differences ($\tau - 1$), and the size of the velocity bin, Δv: $V_{s,u}^{\mathrm{MB}} = f(v)_{s,u}^{\mathrm{MB}} N_\alpha (\tau - 1)\Delta v$.

The velocity-dependent weighting factor was introduced in order to increase the contribution of the differences coming from the high-velocity end of the velocity distribution. It presently has the form given earlier by equation (4).

As was mentioned earlier, it is possible to use additive (b_i) or/and multiplicative (a_i) renormalization during the χ^2 calculation. (In case of $g(r,t)$ only multiplicative renormalization is allowed.) The calculation of the renormalization constants is given by equations (21)–(23).

In the case of only additive renormalization:

$$a_i = 1 \qquad b_i = \frac{\sum_j A_{i,j}^E - \sum_j A_{i,j}^E}{n_i}. \qquad (21)$$

In the case of only multiplicative renormalization:

$$a_i = \frac{\sum_j A_{i,j}^E A_{i,j}^C}{\sum_j (A_{i,j}^C)^2} \qquad b_i = 0. \qquad (22)$$

In the case of both additive and multiplicative renormalization:

$$a_i = \frac{n_i \sum_j A_{i,j}^E A_{i,j}^C - \sum_j A_{i,j}^E \sum_j A_{i,j}^C}{n_i \sum_j (A_{i,j}^C)^2 - \sum_j A_{i,j}^C \sum_j A_{i,j}^C} \qquad b_i = \frac{\sum_j A_{i,j}^E - a_i \sum_j A_{i,j}^C}{n_i} \qquad (23)$$

where n_i is the number of data points for the ith series.

In the case of fitting $g(r,t)$, the criteria of acceptance had to be changed to give a greater weight to the differences between the small values coming from the distinct part of $g(r,t)$. Satisfactory results could be achieved by the introduction of logarithmic χ^2 applied with proper weighting together with the original χ^2 (defined by equation (20)). The contribution coming from the difference of the logarithmic $g(r,t)$ is defined in the following way:

$$\chi^2_{\mathrm{lg,grt}} = \sum_{k}^{n_{grt}} \frac{\sum_j \left(\lg\left[g(r,t)^C_{k,j}+1\right] - \lg\left[g(r,t)^E_{k,j}+1\right]\right)^2}{\sigma^2_{\mathrm{lg},k}}. \tag{24}$$

Renormalization is not defined here. According to the logarithmic χ^2, a move is accepted if χ^2_{lg} decreases; if it increases then the move will be accepted with a probability of $\exp[(\chi^2_{\mathrm{lgold}} - \chi^2_{\mathrm{lgnew}})/2]$. The same move is evaluated according to the normal χ^2; the move is acceptable only if it is accepted according to both the normal and logarithmic χ^2.

Appendix C. Parallelization

As the computational requirements have substantially increased, it was essential to increase the speed of the program; this was achieved by parallelization of the code. To make the best use of our computer facility, multi-threaded versions using the portable operating system interface (POSIX) applicable on shared-memory multi-processor computers, as well as network versions using the message passing interface (MPI) standard with the possibility of multi-threading (in case any computer of the network has more than one available processor) were developed. Both the multi-threaded and the MPI-multi-threaded RMC*t* exist in two versions, the 'short'-threaded and 'long'-threaded versions, depending on the lifetime of the created auxiliary threads. That is, in total, five different versions of the RMC*t* code exist, including the standard consecutive version. All of the program versions were tested both in Linux and Windows environment; in the case of Windows the POSIX for the Win32 interface was applied. The speedup $\{S(p)\}$ and efficiency $\{E(p)\}$, as defined in [15] and given by equation (25), for the different versions are shown in table C.1.

$$S(p) = \frac{C_1}{C_p}; \qquad E(P) = \frac{S(p)}{p} \times 100\% \tag{25}$$

where C_1 is the elapsed time for the one-processor application, C_P is the elapsed time for the parallel application using p processors. All the simulations used the same parameters, the random number generator was seeded the same way, and the same number of steps were generated.

Several methods can be found in the literature for the parallelization of Monte Carlo simulations [15]. During the RMC*t* simulation, similarly to RMC, distances between particles are calculated at least up to half the box length to give good statistics and make the pair correlation function decay to minimize truncation errors during Fourier transformation. This long-range correlation made parallelization achieved by spatial decomposition impossible. As the entire course of the RMC*t* simulation can be compared to the warm-up period of a Monte Carlo (MC) simulation, time decomposition by running independent, parallel simulations differing only in the random number seed was equally unsuitable.

As the computational requirements in each iteration loop are much higher than in the case of the static RMC (and MC), parallelization was achieved by using an unmodified Markov chain with the farm algorithm, simply distributing the most time-consuming calculation steps among the processors. These are the calculations of the histogram and its change and the double Fourier transformation; however, other calculation tasks and updating large arrays were also

Table C.1. The speedup and the efficiency of the parallel versions related to $p = 1$ standard consecutive RMCt. A total number of two threads ($p = 2$) was used for the multi-threaded versions. The MPI-multi-threaded versions were tested on two nodes, each of them using 2-2 threads ($p = 4$). It has to be mentioned that the speedup achieved strongly depends on the simulation parameters (system size, experimental data size) and the applied computer architecture. The simulation system used for testing consisted of 2992 atoms/configuration, with $T = 2000$ configurations, $W_{\text{tframes}} = 1600$. That is, in this case the thread lifetime was long enough even for the 'short'-threaded versions to land on the free processor by load-balancing and achieve speedup.

Version	p	S	E (%)
s-m_RMCt	2	1.7	85.0
l-m_RMCt	2	1.8	90.0
s-MPI_RMCt	4	2.8	70.0
l-MPI_RMCt	4	3.5	87.5

parallelized in certain versions. The task splitting to equal parts for the histogram (change) calculation was achieved by distribution of the time frames among the threads, so mutual exclusion (mutex) usage could be eliminated. The other task splits were based on splitting of the large arrays to calculate or copy without mutex usage.

C.1. Multi-threaded versions

The multi-threaded versions were created because nowadays lots of PCs have more than one processor or are capable of increasing the program's speed due to hyper-threading. This mostly means that two processors are available, but the program can work with more processors as well, since the total number of threads is a parameter.

All the threads of the program share the memory and the data segment. In case of the multi-threaded versions the main thread of the program takes care of the non-parallelized work, creates the auxiliary threads, makes the move(s) and splits the tasks among the threads while taking an equal share in the calculation, as well.

The 'short'-threaded version (s-m_RMCt) was created first. Here the auxiliary threads are created just for a specific task (histogram (change) calculation, or double Fourier transformation) and after completing their work they terminate and join the main thread. Although the thread creation and termination is a very quick process this approach on faster machines and for smaller system sizes could not achieve similar speed increase as on slower ones as the relatively short-lived auxiliary threads did not have enough time to land on the free processor by load balancing. This fact made the 'long' multi-threaded version (l-m_RMCt) necessary.

In the case of l-m_RMCt the task of the main thread is very similar to the s-m_RMCt, but the auxiliary threads are created at the beginning of the program and only join the main thread at the end. The synchronization of the tasks of the different threads is achieved by signals. Due to the constant existence of the auxiliary threads most of the work of the main loop is parallelized besides the tasks that were parallelized in the s-m_RMCt, including:

- copying the modified histogram parts;
- calculating and copying of the partial pair correlation function (ppcf);
- calculating and copying of the $g(r, t)$ partials in case of $g(r, t)$ fitting;
- calculating the total $g(r, t)$ in case of $S(Q, \omega)$ fitting.

C.2. MPI-multi-threaded versions

This version was designed to be used on computer networks, where each computer can access only its own memory and data segment. Multi-threading, either with 'short'-threading (s-MPI_RMCt), or 'long'-threading (l-MPI_RMCt) can be used as well, if more processors available. Multi-threading was applied at the same places as in the respective purely multi-threaded versions (with the exception that the calculation of $g_{\mathrm{T}}(r,t)$ both for $g(r,t)$ and $S(Q,\omega)$ fitting was also implemented in both cases).

Due to the nature of the problem the MPMD (Multiple Program Multiple Data) program structure was adopted, with one *boss* program executing the tasks that cannot be parallelized (such as splitting the work between the threads, generating the moves and broadcasting the new trial coordinates), and taking equal share in the parallel work, as well. $N_{\mathrm{nodes}} - 1$ *slave* programs execute the work assigned to them by the *boss*. The segmentation of the work is done by the *boss* at the beginning of the program, creating non-consecutive time frame segments for each node. This segmentation does not change during the simulation. Each node possesses all the particle coordinates and calculates its contribution to the χ^2 in the case of $g(r,t)$ fitting and its $S(Q,\omega)_{\mathrm{inode}}$ which are summed by the *boss*, and receives whether the move is acceptable or not, keeping message transfer minimal.

The nodes used in the testing of the MPI versions were connected by an ordinary Ethernet link with maximum speed of 100 Mb s^{-1}.

References

[1] Keen D A, Pusztai L and Dove M T (eds) 2005 The first 15 years of reverse Monte Carlo modelling *J. Phys.: Condens. Matter* **17** S1 special issue

[2] McGreevy R L and Pusztai L 1988 *Mol. Simul.* **1** 359

[3] McGreevy R L and Zetterström P 2003 *Curr. Opin. Solid State Mater. Sci.* **7** 41–7

[4] Zetterström P and McGreevy R L 2001 private communication

[5] Evrard G 2003 private communication

[6] Evrard G and Pusztai L 2005 *J. Phys.: Condens. Matter* **17** S1

[7] Zetterström P, Belushkin A V, McGreevy R L and Shuvalov L A 1999 *Solid State Ion.* **116** 321

[8] Mellergård A and McGreevy R L 1999 *Acta Crystallogr.* A **55** 783

[9] Allen M P and Tildesley D J 1987 *Computer Simulation of Liquids* (Oxford: Clarendon)

[10] see e.g. Wannier G H 1987 *Statistical Physics* (New York: Dover)

[11] Squires G L 1996 *Introduction to the Theory of Thermal Neutron Scattering* (London: Dover)

[12] Hansen J-P and McDonald I 1990 *Theory of Simple Liquids* (New York: Academic)

[13] see e.g. Dove M T 2003 *Structure and Dynamics* (Oxford: Oxford University Press)

[14] Van Hove L 1954 *Phys. Rev.* **95** 249–62

[15] Heffelfinger G S 2000 *Comput. Phys. Commun.* **128** 219–37