

# Optimization of neutron scattering instruments using different optimization routines

**K Lieutenant**

Hahn-Meitner-Institut Berlin, Abteilung SF1, Glienicker Straße 100, D-14109 Berlin, Germany

E-mail: lieutenant@hmi.de

Received 11 November 2004, in final form 11 November 2004

Published 21 January 2005

Online at [stacks.iop.org/JPhysCM/17/S167](http://stacks.iop.org/JPhysCM/17/S167)

## Abstract

A description of how the Monte Carlo simulation of a neutron scattering instrument can be combined with an optimization routine in order to design an instrument with the best performance is given. The parameters optimized here describe the converging guide exit of an instrument. A least-square fitting routine and a Metropolis algorithm (as used in the reverse Monte Carlo method) were used to reach this goal. The advantages and disadvantages of the use of these routines are discussed. A combination of both methods is proposed to find the optimal values with high accuracy. In general, the numerical optimization yields results that can sometimes not be obtained by varying the parameters by hand.

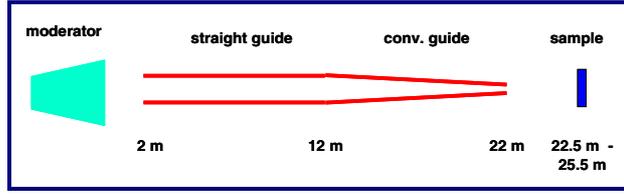
(Some figures in this article are in colour only in the electronic version)

## 1. Introduction

In recent years, several simulation packages have been developed to simulate neutron scattering instruments, such as NISP [1], RESTRAX [2], McStas [3], VITESS [4] and Ideas [5]. Their reliability has been proved in a benchmarking simulation of the triple-axis spectrometer H8 [6]. Many MC simulations have been performed to improve the design of existing instruments or to design new instruments, e.g. [7–11]. According to the ideas of instrument scientists or those responsible for instruments, some set-ups were simulated and compared.

The next step is to optimize instrument parameters by numerical optimization routines. First attempts in this direction have already been done: the guide exit of a planned spin echo instrument was optimized [12, 13], and general features of guide exits were examined [14]. The optimization method used was a standard least-square fitting routine with a special damping algorithm. This fitting program was used successfully for data evaluation in many applications, e.g. [15]. But in the studies mentioned above, it has been found that the statistical nature of the Monte Carlo method changes the optimization process for the worse.

Before continuing to use the method, it seems necessary to check the reliability of the results. Therefore, the topic of this paper is to study the optimization process in detail and



**Figure 1.** Layout of the instrument used in the simulations.

**Table 1.** Simulation data: all parameters are fixed during optimization except those defining shape and exit size  $a \times a$  of guide 2 (which has a quadratic cross section at any point).

Component	Position (m)	Parameters
Source	0	Continuous source $12 \times 12 \text{ cm}^2$ , 1–12 Å moderator temperature 45 K, total flux $3 \times 10^{13} \text{ n cm}^{-2} \text{ s}^{-1}$
Guide 1	2–12	Straight, $6 \times 6 \text{ cm}^2$ $m = 1$ , waviness $0.005^\circ$
Guide 2	12–22	Converging/diverging $6 \times 6 \text{ cm}^2 \rightarrow a \times a$ $m = 3$ , waviness $0.005^\circ$
Sample	23.5	$2 \times 2 \text{ cm}^2$

to compare the results to those obtained by another way of finding optimal parameters. The alternative method used is the Metropolis algorithm [16], as it is used in the reverse Monte Carlo method [17].

## 2. Methods

### 2.1. MC simulation

The simulation package VITESS [4] was used to perform the simulations. A typical instrument was simulated consisting of a continuous source with a cold moderator, a guide of  $6 \times 6 \text{ cm}^2$  cross section and 20 m length with a waviness of  $0.005^\circ$  beginning 2 m from the moderator, and a sample of  $2 \times 2 \text{ cm}^2$  size 1.5 m behind the guide exit (see figure 1; details are given in table 1). The first 10 m of the guide were of constant size with a normal Ni coating, the second 10 m had a  $m = 3$  supermirror coating and were varied in shape. The size decreased either linearly over the 10 m, or consisted of 5 pieces of 2 m length. All components had the same sizes in the horizontal and vertical direction. To compare the results, a transmission  $T$  was defined as the ratio of the count rate on the sample to the count rate at the end of the straight guide:

$$T(\lambda) = I_{\text{sample}}(\lambda)/I_{\text{guide}}(\lambda). \quad (1)$$

Three million trajectories were started at the source and entered the guide, more than half a million of them reached the end of the first guide and were treated then in the variation of the guide exit. First tests showed that this number is necessary for a successful optimization.

### 2.2. Optimization

The simulation of the instrument was combined with an optimization routine to find the shape and the exit size of guide 2 yielding best results. All other parameters were kept fixed.

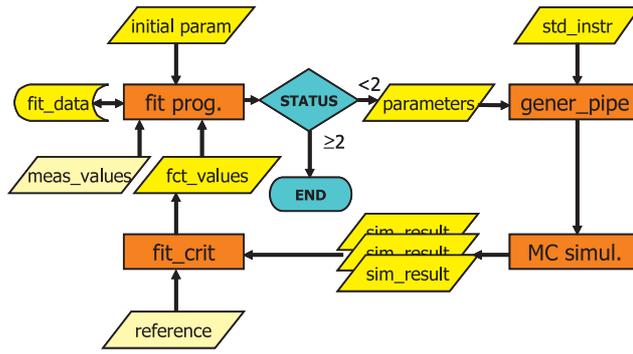


Figure 2. Flow chart of the optimization program.

The method is shown in figure 2:

- (1) The first set of parameters  $\underline{x}_1$  (and some parameters controlling the simulation procedure) have to be given in the file 'start\_param'.
- (2) Depending on the optimization algorithm, one of the parameters or all parameters to be optimized are varied giving sets  $\underline{x}_{1,j}$  that are written to the file 'parameters'.
- (3) For each parameter set in this file, the tool *gener\_pipe* generates a batch file containing the MC simulation by combining the unchanged instrument described in 'std\_instr' with the actual parameter set.
- (4) Each MC simulation is performed and the results are written to the files 'sim\_results'.
- (5) The tool *criterion* evaluates the simulation results and writes the determined values to a file 'fct\_values'. In this tool the optimized magnitude is defined. (The simulation results can be compared to a 'reference' function, e.g. a noise function.)
- (6) The resulting values  $f_i(\underline{x})$  are compared now to measured or theoretical values  $y_i$  stored in 'meas\_values': The sums of squared errors (over all points  $i = 1$  to  $i$  max)

$$Q(\underline{x}_{1,j}) = \sum_{i=1}^{i \max} (f_i(\underline{x}_{1,j}) - y_i)^2 \quad (2)$$

for the varied parameter sets  $\underline{x}_{1,j}$  are compared to that of the first set  $Q(\underline{x}_1)$ . Either the optimization is stopped now or a new set  $\underline{x}_2$  is created by the fitting procedure *fit\_prog*. In the latter case, the procedure is continued with steps (2), (3), ... using the new set  $\underline{x}_2$  instead of the previous one, and so on.

The instrument can be optimized in many aspects, e.g. highest flux, best signal-to-noise ratio, a certain resolution function, etc. As a criterion for best results, the highest transmission  $T$  averaged over the whole wavelength range  $w_s$  chosen, i.e.

$$T_{av} = \int T(\lambda) d\lambda / (\lambda_{\max} - \lambda_{\min}). \quad (3)$$

This means that each wavelength has got the same weight in the optimization. (If, e.g., a maximal flux is the criterion, the optimization will mainly take into account the wavelength range with the highest flux.)

Using this criterion, there is only one value calculated for every simulation. This value should be as large as possible. So the situation is simplified compared to the more general treatment above:  $i \max = 1$  in equation (2);  $f_1(\underline{x}) = 1/T_{av}$ ;  $y_1 = 0$ .

The status of the minimizing process is stored in the file 'fit\_data'; this is necessary because the minimization and the simulation program are running alternately.

**Table 2.** Exit sizes of a linearly converging guide obtained in runs with different random seeds and initial values using the least-square fitting method.

	Initial value		
	3 cm	6 cm	9 cm
Random seed	Exit size (cm)	Exit size (cm)	Exit size (cm)
0	3.95	3.98	3.96
1	4.01	3.83	4.05
2	3.82	4.03	3.89

**Table 3.** Properties of a converging guide with elliptic shape obtained in runs with different initial values using the least-square fitting method and the Metropolis algorithm.

Method	Run (m)	Long axis (cm)	Short axis (cm)	Shift (m)	Exit size (cm)	Average	Flux on
						transmission (%)	sample ( $\text{n cm}^{-2} \text{s}^{-1}$ )
Least-square 1	13.75	6.50	6.50	-1.14	3.81	18.3	$2.01 \times 10^9$
Least-square 2	13.77	7.48	6.50	-1.30	4.28	18.2	$1.95 \times 10^9$
Least-square 3	11.90	6.46	6.50	0.47	3.87	18.7	$2.05 \times 10^9$
Least-square 4	8.63	6.46	6.50	3.09	3.87	20.2	$2.21 \times 10^9$
Least-square 5	7.80	6.58	6.50	3.83	4.03	20.5	$2.28 \times 10^9$
Metropolis <sup>a</sup> 1	10.61	6.33	6.50	1.68	3.93	19.2	$2.10 \times 10^9$
Metropolis <sup>b</sup> 1	7.94	6.60	6.50	3.69	4.00	20.6	$2.28 \times 10^9$
Metropolis <sup>c</sup> 1	$7.96 \pm 0.23$	$6.54 \pm 0.09$	6.50	$3.66 \pm 0.18$	4.00	20.6	$2.28 \times 10^9$

<sup>a</sup> Best parameter set within first 1000 steps.

<sup>b</sup> Best parameter set during run (stopped after 4100 steps).

<sup>c</sup> Average over the last 388 executed steps (limit of maximum set to 20%, see text).

In practice, a numerical optimization can at best be done by performing the first part of the instrument simulation only once and repeating only the simulation of the second part of the instrument many times.

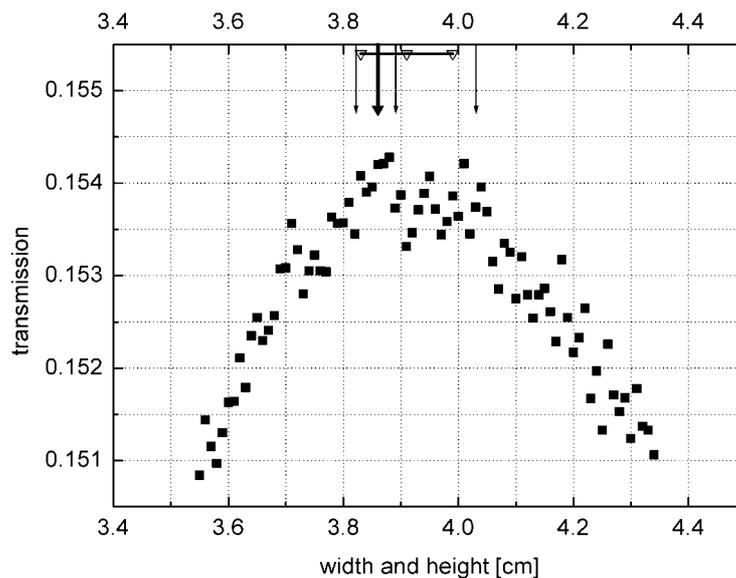
One program for optimization is a usual least-square fitting routine working with the matrix of partial derivatives. These derivatives are approximated by quotients of differences; values used in the present optimization are:  $\pm 0.1$  cm or  $\pm 0.1$  m, respectively (see units in tables 2 and 3). In the case of linear functions, the inverted matrix determines the position of the minimum. For the real functions, the direction of change is kept but the length may be reduced until it obeys a special criterion [18].

As an alternative optimization method, the Metropolis algorithm was used [16]. In each step the parameter to be changed, as well as the magnitude of change (within a certain limit), are chosen by a Monte Carlo choice. The maximal step sizes in these optimizations are  $\pm 0.2$  cm or  $\pm 0.2$  m, respectively (see units in tables 2 and 3). In the case of an improvement, the change is accepted. In the case of a change for the worse, the change is accepted with probability

$$p = \exp(-(Q_c - Q_0)/(2\sigma^2)) \quad (4)$$

with  $Q_0$  and  $Q_c$  being the error square sums before and after the change;  $\sigma$  was set to 0.3 in the runs. This algorithm is a central part of the RMC method [17]. It has two advantages compared to the least-square fit: a local minimum can be traversed to find the absolute minimum, and it is possible to build an average value over its random path within a minimum. The weights

$$p_i = Q_{\text{limit}} - Q_i \quad (5)$$



**Figure 3.** Transmission as a function of exit width (full squares). The arrows mark the sizes found in optimization runs using least-square fits (thin arrows) and Metropolis algorithm (thick arrow). The indicated range shows average value and standard deviation (for the size) of the random walk within the maximum using the Metropolis algorithm.

were used to calculate a weighted average and its standard deviation. The minimum is thus defined as the range giving error square sums below  $Q_{\text{limit}}$ .

### 3. Results and discussions

#### 3.1. Stability of the method in the one-dimensional case

To study the stability of the routine and the influence of the MC random events in the instrument simulation, the instrument was simulated with three different random seeds. For each simulation, the minimization was started with three different initial values using the normal least-square fitting routine.

The minimum was found in each case, but the result depended on the random sample used in the simulation as well as on the initial value (see table 2, these values differ from those found earlier [14] because of a different damping criterion). The first finding is natural; the second is a bit surprising. A close look at the minimization process showed that intermediate values were indeed checked, but were found to be worse than higher or lower values for the exit size. This means that several maxima for the average transmission exist. To check this, a scan of the average transmission around the maximum was carried out, varying the exit size in steps of 0.1 mm. Random seed 2 was used for these simulations (cf table 2). The result is shown in figure 3 (black squares). There is clear maximum of the average transmission around 3.9 cm but with large fluctuations causing several local maxima. So the optimizing procedure ended in one of these local maxima (thin arrows).

The Metropolis algorithm has no end, because it can enter and leave any minima. Therefore, the size giving the highest average transmission during the whole run was searched for. Indeed the absolute maximum (according to the scan) was found (thick arrows in figure 3).

In addition, the average value of the random walk around the optimal size and its standard deviation were calculated. The limit was set to  $T_{av,limit} = 0.1515$ . The result of the run was:  $3.91 \pm 0.08$  cm in 177 steps (three connected triangles in figure 3). This was already found within the first 500 steps, starting with the initial value 6.0 cm (straight guide). The result is just what could be expected from the shape of the curve (average transmission versus exit size). This means that this method is well suited to searching for an optimum which is disturbed by fluctuations. Therefore, it should be used in cases where the number of random events (in a Monte Carlo simulation) is small.

### 3.2. Behaviour in multi-dimensional cases

To check the optimization of several free parameters, the exit was divided into five pieces of 2 m length each. The shape was described by an ellipse with three parameters: long diameter (along the neutron flight direction), short diameter (corresponding to maximal guide width) and shift (along the flight direction).

If the least-square fitting procedure is used here, the optimum is no longer found in each case. If the initial values are too far away from the optimal values, the optimization process often ends in a local minimum far from the absolute minimum (see table 3). This is mainly due to the fluctuations caused by the limited number of random events in the MC simulation; the same program hardly ever caused such problems when it was used in data evaluation, where the calculated functions are smooth.

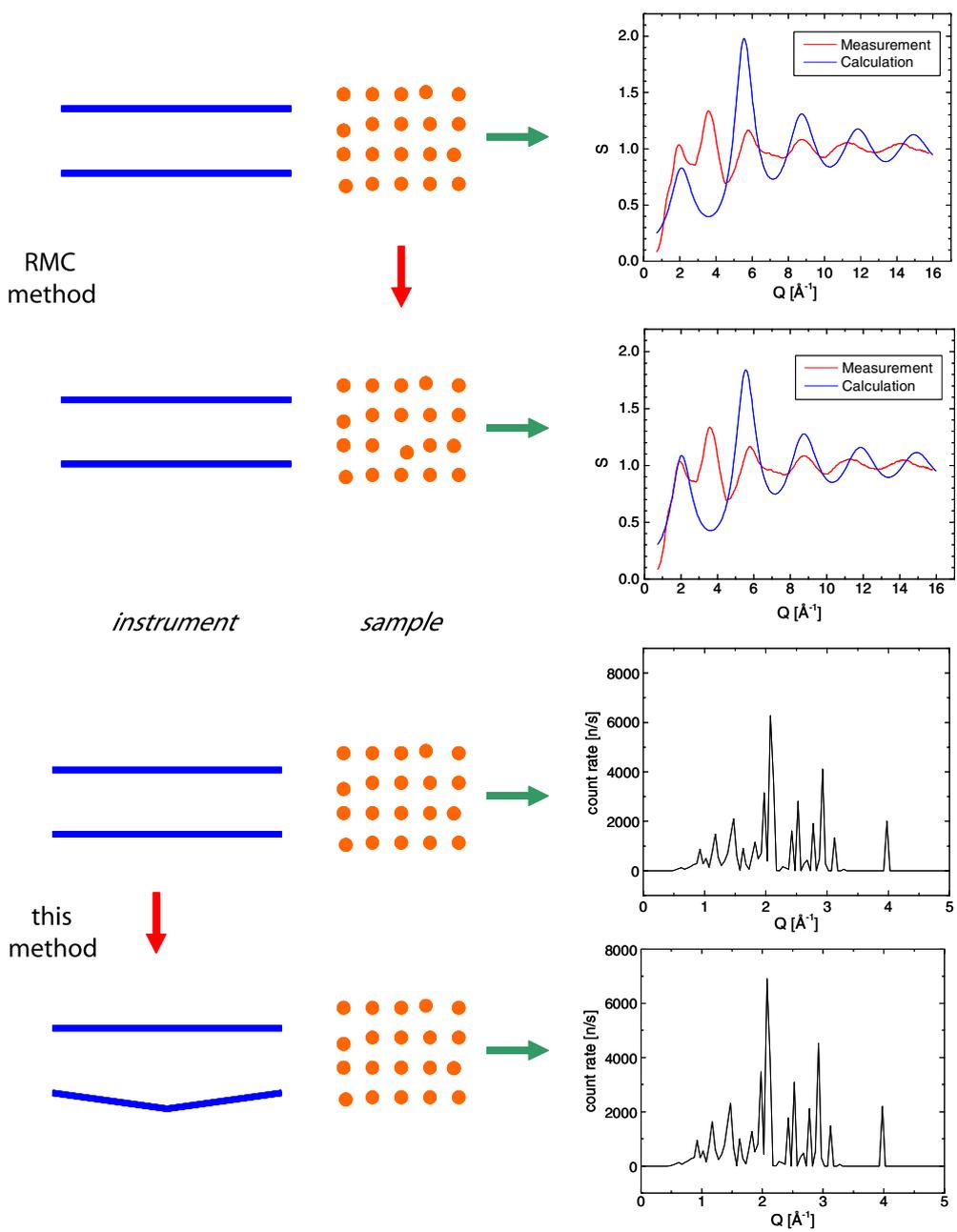
On the other hand, the Metropolis algorithm approaches this optimum slowly. Starting with the same initial values as the first and second least-square fit (15 m, 6 cm, 0 m), the best set found within the first 1000 steps is better than the result of the first three least-square fits, but worse than the last two fits (cf table 3); after 1500 steps the final values are reached. The slow convergence can be a problem, if a larger part of the instrument has to be simulated in each step. Here it was not; about 1500 steps could be run per day on my computer; a good 2 GHz PC might be a factor of four faster.

In the end, both methods yielded the same values for the optimal parameter set (see table 3). It is important to note that this result could not be achieved by varying the parameters by hand. If only one parameter is changed, it is of course a good solution to scan this parameter, as the first example showed. However, this is practically impossible for several free parameters. This means that this method allows the treatment of more sophisticated solutions. As an example, the best linearly converging exit generates a flux of  $1.77 \times 10^9$  n cm<sup>-2</sup> s<sup>-1</sup> [14] for the instrument simulated here, while the best ellipsoidal shape yields  $2.28 \times 10^9$  n cm<sup>-2</sup> s<sup>-1</sup> (see table 3), which is very close to that of the best shape found so far giving  $2.32 \times 10^9$  n cm<sup>-2</sup> s<sup>-1</sup> [14].

Comparing the methods, it can be said that neither of the methods is in general favourable. If the position of the optimum is completely unknown, it seems reasonable to start with the Metropolis algorithm to approach the optimum. If it takes too much computing time to find the optimum, the best solution to use is the least-square fitting routine as a second step to find the optimum. The last step is another run using the Metropolis algorithm with the option of averaging over the positions during the random walk, because this method delivers the best guess of the optimal values and in addition their standard deviations.

### 3.3. Comparison to standard applications of the RMC method

This application of the Metropolis algorithm is quite different from the reverse Monte Carlo method, as it has been used up to now. However there also some similarities—apart from the same optimization algorithm. Figure 4 tries to illustrate this.



**Figure 4.** Comparison of this application to the standard application of the RMC method. If the system instrument + sample is seen as a unit, these two application are similar in the way that the system is varied step by step to achieve a good agreement between the simulated spectrum and the measured or desired one.

In the case of the RMC, the sample is varied step by step to achieve better agreement between the measured and calculated spectra. The instrument is not treated explicitly, only the instrument resolution has an influence on the result.

In a more general case than the one treated here, a sample is also part of the instrument simulation. Usually a sample with a well-known spectrum is used. In this case, the instrument could be varied until the response of the system instrument + sample fits to this spectrum. Alternatively a certain, e.g. symmetric, resolution is given, and the instrument is optimized to reach such a resolution curve.

In this view, instrument and sample are a unit; by performing a simulation, we make a virtual experiment. Both parts contribute to the signal and both can be varied to improve the agreement between calculation and experiment, i.e. instrument parameters as well as sample parameters can be fitted. The two applications mentioned are thus approximations to the evaluation of such a virtual experiment.

#### 4. Conclusions

In general, the combination of MC simulation and optimization routine works very well, yielding result that cannot be obtained by MC simulation. A least-square fit is preferable in the case of good statistics and long simulation times (for the variable part of the instrument). In the case of bad statistics or completely unknown position of the optimum, the Metropolis algorithm is better suited. Building the average of the random walk around the optimum yields the best estimate for the optimal values of the free parameters. In complicated cases a combination of the Metropolis algorithm and the least square fit is the best solution.

#### Acknowledgments

I would like to thank the RMC community, especially G Evrard and L Pusztai, for their support and helpful discussions.

#### References

- [1] Daemen L L, Seeger P A, Hjelm R P and Thelliez T G 1999 *Proc. SPIE* **3771** 80–9  
<http://strider.lansce.lanl.gov/NISP/Welcome.html>
- [2] Šaroun J and Kulda J 1997 *Physica B* **234–236** 1102–4 <http://omega.ujf.cas.cz/restrax/>
- [3] Nielsen K and Lefmann K 2000 *Physica B* **283** 426–32 <http://neutron.risoe.dk/mcstas/>
- [4] Zsigmond G, Lieutenant K and Mezei F 2002 *Neutron News* **13** 11–4 <http://www.hmi.de/projects/ess/vitess/>
- [5] Lee W-T, Wang X-L, Robertson J L, Klose F and Rehm Ch 2002 *Appl. Phys. A* **74** (Suppl.) S1502–4  
<http://www.sns.gov/ideas/>
- [6] Seeger P A, Daemen L L, Farhi E, Lee W T, Wang X L, Passell L, Šaroun J and Zsigmond G 2002 *Neutron News* **13** 24–9
- [7] Litrell K C, Thiyagarajan P, Carpenter J M and Seeger P A 2002 *Appl. Phys. A* **74** (Suppl.) S1480–2
- [8] Šaroun J, Pirling T and Rogge R B 2002 *Appl. Phys. A* **74** (Suppl.) S1489–91
- [9] Klausen S N, Lefmann K, McMorrow D F, Altorfer F, Janssen S and Lüthy M 2002 *Appl. Phys. A* **74** (Suppl.) S1508–10
- [10] Zsigmond G, Lieutenant K, Manoshin S, Bordallo H N, Champion J D M, Peters J, Carpenter J M and Mezei F 2004 *Nucl. Instrum. Methods Phys. Res A* **529** 218–22
- [11] Klose F and Lee W-T 2002 *Appl. Phys. A* **74** (Suppl.) S1486–8
- [12] Lieutenant K, Krist T, Monkenbusch M, Manoshin S, Ohl M, Zsigmond G and Mezei F 2003 *Proc. ICANS-XVI* ed G Mank and H Conrad (Jülich: Forschungszentrum Jülich) pp 505–12
- [13] Manoshin S, Lieutenant K, Zsigmond G, Clemens D and Mezei F 2003 *Proc. ICANS-XVI (FZ Jülich, 2003)* ed G Mank and H Conrad, pp 513–21
- [14] Lieutenant K and Zsigmond G 2004 *Physica B* **350** E687–9
- [15] Steinhoff H J, Redhardt A, Lieutenant K, Chrost W, Hess G, Schlitter J and Neumann H J 1990 *Z. Naturf. a* **45** 677–86
- [16] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 *J. Chem. Phys.* **21** 1087–92
- [17] McGreevy R L and Pusztai L 1988 *Mol. Simul.* **1** 359–67
- [18] Braess D 1966 *Computing* **1** 264–72