Monte Carlo simulations of neutron scattering instruments*

Per-Olof Åstrand[1,2], Kim Lefmann[1] and Kristian Nielsen[1,†]
[1] Materials Research Department, Risø National Laboratory, DK-4000 Roskilde, Denmark
[2] Department of Chemistry, University of Copenhagen, Universitetsparken 5, DK-2100 Copenhagen Ø, Denmark

Abstract

A Monte Carlo simulation is an important computational tool used in many areas of science and engineering. In our presentation, we will describe how Monte Carlo techniques can be used for simulating neutron scattering instruments. The basic ideas, techniques and approximations are presented. Since the construction of a neutron scattering instrument is very expensive, Monte Carlo software used for design of instruments have to be validated and tested extensively. The McStas software was designed with these aspects in mind and some of the basic principles of the McStas software will be discussed. Finally, some future prospects are discussed for using Monte Carlo simulations in optimizing neutron scattering experiments.

Introduction

Neutron scattering is a powerful tool for characterizing various properties of materials and many of the techniques employed give characteristics of a material that currently cannot be obtained with other experimental characterization methods. Neutron scattering is an expensive technique, however, which is due to the large-scale facility required for generating the neutrons. On the other hand, the grand challenges facing the scientific community in materials science compensate more than well for the large investment. For example, the ever-increasing demand for better computers has lead to a miniaturization of computer components down to an atomistic scale consisting of only a few atom layers [1] and the potential development of superconductors at room temperature would solve a large part of the current problems with storing and transporting energy. New functional materials are required and these materials need to be characterized. To meet these requirements several large-scale facilities are currently upgraded, constructed or planned. See for example the contributions in this report regarding the upgrade of the neutron source at KFKI in Budapest [2] and the plans regarding a second target station at ISIS and the European spallation source (ESS) [3].

Because of its complexity any assistance in the design of a neutron scattering instrument or in planning and and analyzing experiments is valuable, and in this respect in particular Monte Carlo simulations have turned out to be useful. A Monte Carlo simulation is a computational tool for calculating multidimensional integrals that are too complex to be calculated analytically or by simpler numerical methods. Monte Carlo techniques are used in many areas of science and engineering, amongst others in the study of molecular liquids [4],

*Paper presented at *Workshop on New Opportunities in Single Crystal Spectroscopy with Neutrons*, Balaton (Révfülöp), Hungary, April 19-22, 2001

†Present address: Sifira A/S, Fredericiagade 16, DK-1310 Copenhagen K, Denmark

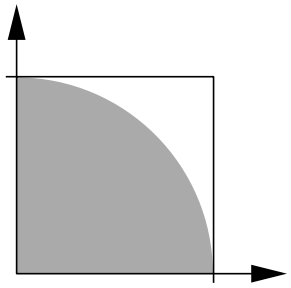

Figure 1: A circle segment with unit radius inscribed in a square.

polymer science [5], quantum mechanics [6], materials science [7], and in simulations of neutron scattering instruments, which is the topic of the present paper. This paper will be organized as follows: the Monte Carlo technique is introduced; Monte Carlo simulations of neutron scattering instruments are discussed; software requirements and in particular the design of the McStas software are discussed; and finally, some future prospects of Monte Carlo simulations of neutron scattering instruments are discussed.

Monte Carlo simulations: a gentle introduction

The Monte Carlo technique is a computationally demanding tool for calculating multidimensional integrals. Its name *Monte Carlo* was coined because of the extensive use of random numbers by Metropolis and Ulam in a paper describing the early work at Los Alamos [8]. The Monte Carlo technique is illustrated by a simple example, namely the calculation of the value of π [4]. As illustrated in Figure 1, the value of π may be given from the area of a circle segment with unit radius inscribed in a square as

$$\pi = \frac{4 \times \text{area of circle segment}}{\text{area of square}} \tag{1}$$

If two random numbers, (x,y), are generated from an *even* distribution of random numbers within the range [0,1] (assuming a unit radius of the circle), each point (or each *event*) will be found within the square but only the points fulfilling the trivial constraint $x^2+y^2<1$ will be found within the circle segment. Furthermore, if the procedure is repeated many times, all the events will cover the unit square evenly if an even distribution of random numbers is employed. If the number of points found within the circle segment is counted, it is realized that the relation between the area of the circle segment and the square and consequently the value of π can be obtained as

$$\pi = \frac{4 \times \text{number of counts within the circle segment}}{\text{total number of events}} \tag{2}$$

The calculation of the value of π is obviously a constructed example, but it illustrates the basic idea of a Monte Carlo simulation. The sampling of events has to be repeated many times (*e.g.* 10^6 - 10^9 times) to get any reasonable precision of π. Consequently, a Monte Carlo

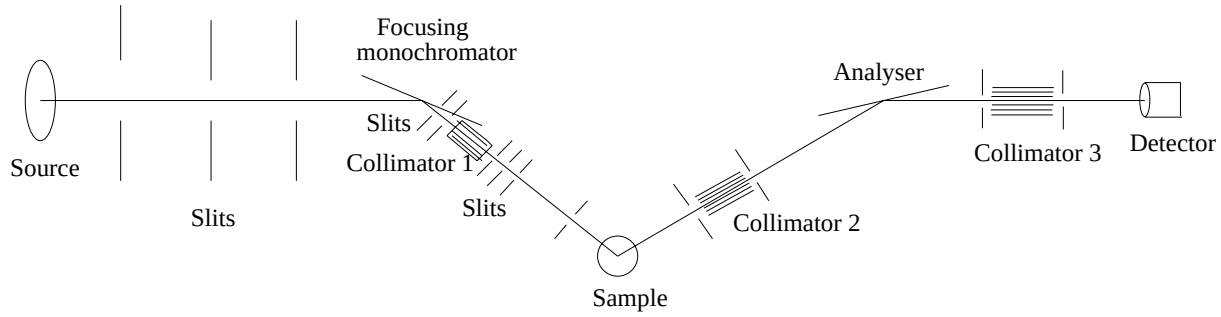

Figure 2: A sketch of the TAS1 instrument at Risø.

simulation is a computer-demanding, brute-force and not very elegant method to calculate integrals that we cannot calculate in any other way.

The hit and miss example in the previous paragraph is conceptually easy to understand and it can be generalized to any integral [4]. An integral $I = \int_{x_i}^{x_f} f(x)\, dx$ may be rewritten as

$$I = \int_{x_i}^{x_f} \left(\frac{f(x)}{\rho(x)} \right) \rho(x)\, dx \tag{3}$$

where $\rho(x)$ is an arbitrary probability distribution. By performing τ trials, where each trial consists of choosing a random number τ_i from the distribution $\rho(x)$ in the range (x_i, x_f), the integral may be rewritten as

$$I = \frac{1}{\tau} \sum_{i=1}^{\tau} \frac{f(\tau_i)}{\rho(\tau_i)} = \langle \frac{f(\tau_i)}{\rho(\tau_i)} \rangle_\tau \tag{4}$$

where the brackets, $\langle \ldots \rangle$, represent an average over all trials.

Monte Carlo simulations of neutron scattering instruments

A Monte Carlo simulation of a neutron scattering instrument (see *e.g.* the TAS1 instrument in Figure 2) resembles to a large extent the experimental situation. A random *state* of a neutron is generated according to a probability distribution defined by the type of neutron source. The neutron is traveling from the source towards the detector and each beam component in turn modifies the state of the neutron. If the neutron is not absorbed or scattered away, it will eventually reach the detector and its final state is recorded for further analysis. To obtain a reasonable representation of the characteristics of the neutron source, the sampling of random neutron states is carried out repeatedly (normally more than 10^6 times) until the desired statistics is obtained. Finally, the recorded states are analyzed in very much the same way as a real experiment, and Monte Carlo simulation of a neutron scattering instrument may thus be regarded as a *virtual experiment*.

The *state* of a neutron has to be represented in some way in the model, and normally the so-called semi-classical approach is adopted which means that wave mechanics is used

to describe the neutron and its interaction with different beam components [9, 10]. The representation of neutron may thus be written as a six-dimensional vector; either as ($\mathbf{r}$, E, Ω), where $\mathbf{r}$ is the position vector, E is the energy, and Ω is the traveling direction, or as ($\mathbf{r}$, $\mathbf{v}$) where $\mathbf{v}$ is the velocity vector. In addition, if polarization experiments are considered [11], the spin of the neutron, $\mathbf{s}$, has to be included. Since the length of the spin vector for a neutron is $\frac{1}{2}$, only a two-component representation is required for describing its orientation. However, a full three-component representation of the spin is preferred since it allows for an extension to neutron rays. Finally, if we would like to model also time-of-flight experiments, the traveling time t, for example since emission from the source, has to be stored.

In a real semi-classical simulation, a neutron is either traveling towards the detector interacting with each component or it has been absorbed or scattered away. This can be represented by a neutron weight p, which is either 1 if the neutron is still present or 0 if the neutron is lost. For some beam components, *e.g.* transmission through filters, reflection from monochromators, scattering from samples, etc., it is possible to define a neutron weight in the range $0 < \mathsf{p} < 1$. For example, if the probability for transmission through a filter is 10%, p is multiplied by 0.1 and the neutron continues to the next component with this new weight. Equivalently, one could allow 10% of the neutron to pass the filter and 90% of the neutrons would be absorbed, but the usage of the weight p improves the statistics of the simulation considerably. The resulting state vector describing a neutron is thus

$$(\mathbf{r}, \mathbf{v}, \mathbf{s}, \mathrm{t}, \mathrm{p}) \tag{5}$$

i.e. the position, velocity, spin, time and weight of the neutron.

Software requirements

It is in principle rather easy to write a simple Monte Carlo program in Fortran or C for simulating for example the TAS1 instrument in Figure 2. However, to write a general simulation program for various types of neutron scattering instruments and that is used by a large part of the neutron scattering community, some aspects have to be considered in detail [12–14]. For example, if the ESS project is realized, the simulation programs used by the neutron scattering community for designing the instruments will be the programs that fulfil some conditions.

- The program has to be **validated** extensively. Since neutron scattering is an expensive technique, any mistakes in the design of the instruments would be devastating.
- The program has to be **modular**. Many different kinds of instrument and beam components are of interest and new types are constructed all the time. It must be easy for the *user* to extend the simulation program with additional instruments and beam components.
- The program has to be **efficient**. Long simulations are required (normally cpu-hours to cpu-days), and the simulation program therefore has to be coded in a programming language that provides an efficient executable resulting in fast simulations.

- The program has to be **user-friendly**. It must be straightforward to start to use the program and the usage should be more or less intuitive. Secondly, it must be easy for the user to try new ideas, and again it should be straightforward for the user to include minor changes and extensions in the program.

An important initiative is to define a set of standard instrument models for which a Monte Carlo program can be tested against, where the first instrument to be included is a canonical triple-axis spectrometer [15]. Such a suite of standard models would certainly be an important part of the validation procedure for a simulation program, and the initiative is currently supported by five Monte Carlo groups.

The data obtained from a simulation of an instument is analyzed to a large extent in the same way as an experiment. It would therefore be preferred if the same tools could be used to analyze all simulations carried out with various software as well as all experiments carried out at various facilities around the world. The NeXus data format [16] provides such an option and hopefully it will be fully supported in the future.

Design of McStas

The McStas (Monte Carlo Simulation of Triple Axis Spectrometer) software was designed already from the start with the aspects discussed in the previous section in mind, and a version 1.0 was released in October 1998 [13, 14, 17]. Already from the start, other groups were invited to collaborate on the development of components to the software [13], and McStas is currently used extensively not only at Risø.

A meta-language has been constructed especially designed for the task of modelling neutron scattering instruments. The physics of each beam component is defined in a separate file by using this meta-language. Furthermore, an instrument file consisting of a list of components and component parameters is also specified with the meta-language. Many standard components are available in the McStas software (currently around 50 in version 1.4.2 [17]), but it is in principle straightforward for users to implement additional components. The most recent contributions include components for moderators at ESS. It should be noted that all the physics of a neutron scattering instrument are included in the component and instrument definitions and thereby separated from the kernel containing the actual simulation code, which facilitates for the user to make minor changes and additions. Furthermore, McStas contains some components that do not correspond to a physical beam component as for example components for adaptive importance sampling (*vide infra*) and an interface between the McStas and VITESS [18] programs.

The instrument as defined by the meta-language is compiled by using the facilities provided by FLEX (fast lexical analyzer generator) and a Bison parser generator. The compilation checks the semantics of the instrument and component definitions and thus it ensures that these definitions are correct. Furthermore, the compilation results in an ANSI-C code which provides an efficient and fast simulation code.

As discussed in the previous section, the validation of a simulation software is of fundamental importance. Simulations comparing McStas and Restrax [19] with experiments have

been carried out for the cold-neutron triple-axis spectrometer IN14 at ILL, and very close agreement was found between the simulations and experiment [20]. In addition, results from McStas and Restrax have been compared for the new structure powder diffractometer (SPODI) at FRM-II with good agreement between the two simulation programs [21].

A criterion for the success of a simulation is to what extent it has been used by others than the original authors. The McStas software has been used extensively for modelling instrument upgrades at ILL [20, 22, 23], design of instruments at the new facilities FRM-II [21, 24–29], and SNS [30].

Importance sampling

For some instruments, only very few neutrons reach the detector which gives poor statistics and the simulation (or the experiment) has to be run for a very long time. In Monte Carlo simulations, but not in real experiments, the situation can be improved by using *importance sampling*.

In a regular Monte Carlo simulation, the neutron events are sampled from a probability distribution describing the actual neutron source, $\rho_{\mathtt{source}}(\mathbf{r}, \mathbf{v}, \mathbf{s})$. Since all sampled neutron corresponds to experimentally equally possible states, the neutrons may be regarded to have equal weight, $w = 1$. In a simulation, we can equally well sample the neutrons from any distribution, $\rho_{\mathtt{any}}(\mathbf{r}, \mathbf{v}, \mathbf{s})$, but then the weight w has to be modified accordingly. Obviously, it is preferable to sample neutrons more frequently that improve the statistics of the requested result, where a necessary but in principle not sufficient condition is that the neutrons reach the detector. To *a priori* determine the suitable distribution that optimizes the sampling is a very difficult problem, and requires in principle that the results of the simulation are known on beforehand. The solution is to use an *adaptive* procedure where $\rho_{\mathtt{any}}(\mathbf{r}, \mathbf{v}, \mathbf{s})$ is modified on-the-fly during the simulation.

In McStas, *adaptive importance sampling* can be utilized in two different ways by using virtual components: one way is to use the `Source_adapt` component and the other way is to use the `Source_Optimizer` and `Monitor_Optimizer` components and both procedures are described in detail in the McStas manual [14].

It is crucial to at least consider various ways of importance sampling in Monte Carlo simulations. Otherwise, one may spend the simulation calculating the number zero with a very high precision, *i.e.* the neutrons do not reach the detector. Furthermore, one may actually learn something from the resulting probability distribution, which may provide information on how the set-up of the experiment or the design of the instrument may be improved.

Prospects

The information obtained from simulations may be used for several purposes. For example, the design of a new instrument can be optimized, experiments may planned in detail, and the simulation data may assist in the analysis of experimental data [31, 32]. Many other

possibilities exist and here some suggestions of extensions of McStas are listed that more or less corresponds to to the discussion at the McStas workshop at Risø National Laboratory in January 2001 [17, 33].

- Virtual monitors can be placed in principle at each beam components. If the neutron states at each component are stored, it would be possible to correlate any unexpected results with the properties of the neutrons at some component. Consequently, Monte Carlo simulations can be used to obtain information that is attainable in experiments. Some support for calculating correlation functions exist in McStas in the components `Monitor_nD and PreMonitor_nD`. Furthermore, if neutron states at a component are stored, they may be used as a virtual source where a simulation may be restarted.

- In some cases it would be useful to group components in a so-called supercomponent. The most trivial example is that the same setup of components for a part of an instrument is used repeatedly and it would then be appreciated if one can refer to this part by one entry in the instrument definition. Another example is if the position and the relative order of some components are given by an external parameter. This external parameter would be given as input to the supercomponent, which then provided the components in the correct order. Other cases that may be treated within a supercomponent are *e.g.* to allow for the neutron to interact many times with each component, to handle splitting of neutron rays, and decoupling shape and funtion of a sample component.

- It would be useful if an instrument parameter could be optimized automatically in a single simulation. One way to do this would be to modify the neutron state vector as ($\mathbf{r}$, $\mathbf{v}$, $\mathbf{s}$, t, p, Q), where Q is an instrument parameter. Also the values of Q can be sampled according to the Monte Carlo procedure, and an optimum value of Q may be obtained from adaptive importance sampling.

- Currently, the length of a simulation is specified by the number of neutron events. In line with the efforts by R. McGreevy and coworkers in developing software for using other stopping conditions in experiments (such as the precision for a requested property), suitable stopping conditions should be investigated by simulations.

Acknowledgement

The McStas project is funded by grants from the Eurpean Union under the RTD and SCANS programs. Kurt N. Clausen is acknowledged for initiating the project and for continuous support. Heloisa Bordallo and Emmanuel Farhi are acknowledged for providing articles prior to publication. Larry Passell is acknowledged for information regarding the standard Monte Carlo models in Ref. [15].

References

[1] D. A. Muller, T. Sorsch, S. Moccio, F. H. Baumann, K. Evans-Lutterodt, and G. Timp. *Nature*, **399**, 758–761, 1999.

[2] L. Rosta. The new cold neutron research facilities at the Budapest Research Reactor. In This report, 2001.
[3] U. Steigenberger. Advances with Neutron Sources. In This report, 2001.
[4] M. P. Allen and D. S. Tildesley. *Computer Simulations of Liquids*. Clarendon, Oxford, 1987.
[5] C. R. A. Catlow, S. C. Parker, and M. P. Allen, editors. *Computer Modelling of Fluids Polymers and Solids*. Kluwer, Dordrecht, 1988.
[6] M. H. Kalos, editor. *Monte Carlo Methods in Quantum Problems*. Reidel, Dordrecht, 1982.
[7] M. Meyer and V. Pontikis, editors. *Computer Simulation in Materials Science. Interatomic Potentials, Simulation Techniques and Applications*. Kluwer, Dordrecht, 1991.
[8] N. Metropolis and S. Ulam. *J. Am. Stat. Ass.*, **44**, 335–341, 1949.
[9] G. E. Bacon. *Neutron Diffraction*. Oxford University Press, 1975.
[10] G. L. Squires. *Introduction to the Theory of Thermal Neutron Scattering*. Dover Publications, 1996.
[11] F. Tasset. Polarimetric Neutron Scattering. In This report, 2001.
[12] R. K. Crawford. Report on the Workshop on Monte Carlo Simulation of Neutron Scattering Instruments. Technical report, Argonne National Laboratory, 1998. Technical Note ANL/SNS/97-2.
[13] K. Lefmann and K. Nielsen. *Neutron News*, **10**, 20–23, 1999.
[14] K. Nielsen and K. Lefmann. *User and Programmers Guide to the Neutron Ray-Tracing Package McStas, Version 1.4*. Risø National Laboratory, Roskilde, Denmark, July 2000.
[15] L. Passell. Planned Monte Carlo Inter-comparisons. Private communication.
[16] P. Klosowski, M. Koennecke, J. B. Tischler, and R. Osborn. *Physica B*, **241-243**, 151–153, 1998.
[17] See http://neutron.risoe.dk/mcstas/.
[18] D. Wechsler, G. Zsigmond, F. Streffer, and F. Mezei. *Neutron News*, **11**, 25–28, 2000.
[19] J. Šaroun and J. Kulda. *Physica B*, **234-236**, 1102–1104, 1997.
[20] A. R. Wildes, J Šaroun, E. Farhi, I. Anderson, P. Høghøj, and A. Brochier. *Physica B*, **276-278**, 177–178, 2000.
[21] R. Gilles, G. Artus, J. Saroun, H. Boysen, and H. Fuess. *Physica B*, **276-278**, 87–88, 2000.
[22] E. Farhi. Results of Monte-Carlo simulations of the neutron flux along the H15 cold guide. ILL Internal Report, 2001.
[23] E. Farhi. About H15, H22 and H24 McStas simulations. ILL Internal Report, 2001.
[24] G. R. J. Artus, R. Gilles, F. Frey, and W. Sherer. *Physica B*, **276-278**, 77–78, 2000.
[25] Ph. Bernhardt, F. Demmel, and A. Magerl. *Physica B*, **276-278**, 114–115, 2000.
[26] P. Link, G. Eckold, and J. Neuhaus. *Physica B*, **276-278**, 122–123, 2000.
[27] S. Roth, A. Zirkel, J. Neuhaus, W. Schneider, and W. Petry. *Physica B*, **283**, 439–442, 2000.
[28] G. R. J. Artus, F. Frey, and W. Scherer. *Physica B*, **283**, 436–438, 2000.
[29] M. Rotter and R. Schedler. McStas simulation for PANDA. Part Ia. Primary Spectrometer. Technical report, Institut für Angewandte Physik, Technische Universität Dresden, Dresden, Germany, April 2001.
[30] H. N. Bordallo and K. W. Herwig. Monte-Carlo simulations of a high-resolution inverse geometry spectrometer in the SNS - Long Wavelength Target Station. In *ICANS XV - The 15$^{\text{th}}$ meeting on the International Collaboration on Advanced Neutron Sources*, Tsukuba, Japan, 2001. To appear.
[31] K. Lefmann, K. Nielsen, A. Tennant, and B. Lake. *Physica B*, **276-278**, 152–153, 2000.
[32] K. Nielsen and K. Lefmann. *Physica B*, **283**, 426–432, 2000.
[33] K. Lefmann. Summary of McStas workshop at Risø National Laboratory. Unpublished, January 2001.