ICANS-XIII
13th Meeting of the International Collaboration on Advanced Neutron Sources
October 11-14, 1995
Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

A USER-FRIENDLY, GRAPHICAL INTERFACE FOR THE MONTE CARLO NEUTRON OPTICS CODE MCLIB

Thierry Thelliez *, Luc Daemen†, Phil A. Seeger ‡ and Rex P. Hjelm †

* Applications Programming Group, Computing, Information and Communications Division, Los Alamos National Laboratory, Los Alamos, NM 87545 USA
† Neutron Scattering Center, Los Alamos Neutron Science Center, Los Alamos National Laboratory, Los Alamos, NM, 87575 USA
‡ 239 Loma del Escolar, Los Alamos, NM 87544 USA

ABSTRACT

We describe a prototype of a new user interface for the Monte Carlo neutron optics simulation program MCLIB. At this point in its development the interface allows the user to define an instrument as a set of predefined instrument elements. The user can specify the intrinsic parameters of each element, its position and orientation. The interface then writes output to the MCLIB package and starts the simulation. The present prototype is an early development stage of a comprehensive Monte Carlo simulations package that will serve as a tool for the design, optimization and assessment of performance of new neutron scattering instruments. It will be an important tool for understanding the efficacy of new source designs in meeting the needs of these instruments.

1. Motivation

The process of designing new, advanced neutron sources, or even upgrading present sources, must start with the identification of an instrument suite to address the scientific and technical issues likely to be important in the future. The job of designing the instrument suite and neutron source involves extensive collaboration between instrument designers and target-moderator-reflector-shield system designers. The collaboration must start at an early stage of the facility design if the facility is to be properly optimized and its potentials fully realized.

As with any optimization the instrument suite and source design is an iterative process at several interacting levels, and there are important compromises at each level that must be assessed. For example, each instrument must be optimized to its resolution, intensity and measurement domain requirements, and the source must be optimized and matched to these needs. At a higher level conflicts need to be resolved between instruments that share moderators and even between instruments that are on different moderators on the same target.

New instrument and source designs should advance the state of the art, beyond the limits where we have operational experience. Thus, in order carry through the task of facility design

Keywords: Instruments, Simulations, Monte Carlo, MCLIB, User Interface

a means must be provided to predict instrument performance at each stage of the process. This is all the more important, as once the facility is built cost, inaccessibility and shear size make modification to correct mistakes difficult, if not impossible.

Computational assessment of instrument performance requires the calculation of complex integrals, which are currently best handled using Monte Carlo integration. The implementation of this technique requires detailed modeling of the neutron optics of the instrument elements, of the scattering laws of the sample and of detectors. The production of code for any one instrument is quite a considerable undertaking. The development of modular code libraries to simulate the different instrument elements makes the task considerably easier, as the instrument designer can string elements together to produce the needed code. This is the approach used in the modular instrument element Monte Carlo library, MCLIB, described in another contribution in this volume [1]. MCLIB is based on the MCLIB Monte Carlo library developed at the Rutherford Laboratory [2]. This library has been used extensively at Los Alamos to design, optimize and simulate instrument performance of SANS [3-6], neutron reflectometers [7] and chopper spectrometers [8].

The use of this package still requires considerable hand coding to simulate an instrument. Clearly, if research on instrument performance is to be done quickly and effectively, tools must be provided to the instrument designer that will allow him or her to understand the performance of an instrument design, and optimize the design quickly and accurately. This is essential in the development of effective procedures to optimize the instrument with the source and deal with issues that occur in designing the instrument suite. Thus we need to go one level further to provide a simulations package that will remove altogether the need for the end user to do coding. This implies that a user interface be written that will allow a user to build an instrument from predefined instrument elements, and specify the size, shape and position of these elements using simple dialog boxes. The user interface should allow the end user to control the operation of the simulation, then display data. The interface should also provide data assessment capabilities as well. Such an interface would not only be useable as a tool for instrument design, but also would be valuable in modeling for data analysis once the instrument is built, as instrument effects would be accurately included in the scattering simulations.

Our interest at Los Alamos in developing a simulations tool is driven by the need to understand instrument performance with different neutron source designs, to guide us in the design of new SPSS and LPSS facilities, and to understand the relative merits of instruments built for each kind of source. We see such a tool as being invaluable in coming to a consensus as what neutron scattering facilities should be built.

2. Approach

2.1. Basic Requirements

The basic features that must be present in the simulations package is that the user interface be easy to use and that it be interactive. The simulations must be fast to assure reasonable turn around times so that the instrument parameter space can be completely probed. Finally the system needs to be open. This final criterion implies the specification of standards to which others can write and thus make contributions to the program. In this report we address only certain aspects of the user interface, but the design strategy takes into account the other essential aspects that must be present in the entire system.

2.2 Package Design Philosophy:

In order to meet the needs of an easy to maintain, easy to use interactive user interface we use the object oriented paradigm. However, there are considerable drawbacks in run time overhead that would strongly impact performance if the entire Monte Carlo package were developed using this approach. Modern FORTRAN compilers produce highly optimized code for highly efficient and fast numerical calculations. Thus we choose a hybrid approach where the user interface is written in a language designed to support object oriented programming, Smalltalk, and the high performance part of the code is written in FORTRAN. The user interface defines the overall architecture of the package, provides for library management and supports the graphical interface. The interface thus functions to manage the difference modules of the FORTRAN code. This approach also allows us to use the considerable development in instrument simulations that has already been done in MCLIB [1].

2.3 The Architecture of the User Interface:

An instrument is an ordered set of elements, such as a source, a collimator, choppers, a sample and detectors. The elements operate on a neutron, in turn, in the Monte Carlo simulation. It is the function of the interface to allow the end user to define an instrument by specifying the instrument elements the element intrinsic properties positions and orientations and transfer this information to the simulation code. How this information is used and the internal architecture of the Monte Carlo simulation package can be found in reference [1]. These functions of the user interface are carried out by four managers as shown in Fig. 1.

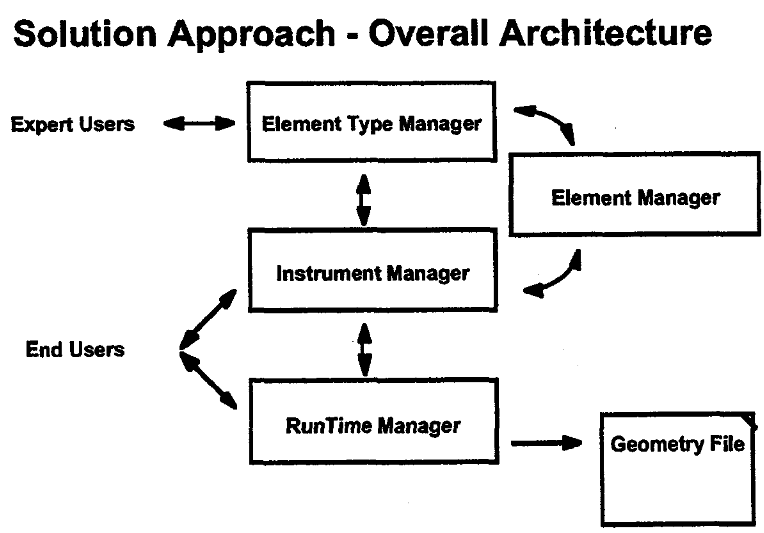

Figure 1: Overall architecture of the user interface and relationships of the managers with one another, the users and the simulation code.

The end user creates and manages the instruments using the Instrument Manager (Fig. 1). The instrument manager allows the user to define an instrument as a set of elements, chosen from a predefined element list. The instrument elements themselves are created and managed thorough the Element Manager (Fig. 1). The Element Manager allows the end user to select from a set of predefined instrument elements, specify the intrinsic parameters of the element (for example the radius of an aperture), and the extrinsic parameters (position and orientation).

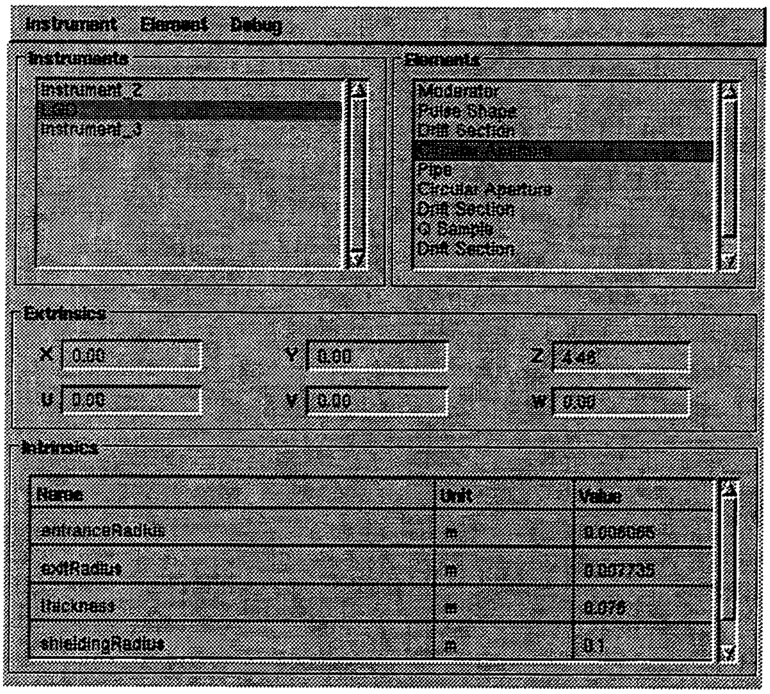

Figure 2: The Instrument Browser.

The interaction between the two managers and the end user is through the Instrument Browser (Fig. 2).

The instrument elements themselves are defined through the Element Types Manager (Fig. 1). This manager allows the expert user to create, define and manage element types through the Element Type Browser (Fig. 3). The physics model for an element is encoded in an MCLIB module [1], and this manager defines the element for the user interface and provides a means for specifying the surfaces and regions that define the element, as well as the intrinsic parameters that are needed to completely specify the surfaces [1]. This manager is maintained by expert users, those who also are also responsible for writing and maintaining the MCLIB codes. This part of the interface is not available to the end user.

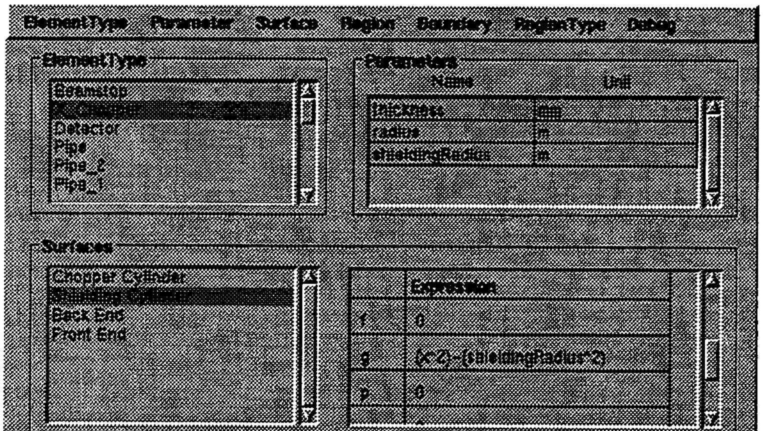

Figure 3: The Element Type Browser.

Finally, once an instrument is set up through the Instrument Browser, the Runtime Manager, which functions to provide for the selection of an instrument and the simulation parameters, writes a geometry file as an output and starts the simulation. The geometry file completely specifies the elements and their arrangement in the instrument [1].

3. Future Work

The present geometry interface is an early prototype in the development of a comprehensive package for instrument design. Future work will expand on the capabilities of the user interface to control execution of the Monte Carlo simulation and provide facility for the display and assessment of results. The next prototype of the package will also provide for a graphical representation of the instruments. We plan to investigate the generation of highly optimized FORTRAN code by the Run Manager as a strategy toward improving the runtime speed and efficiency of the simulation.

4. References

[1] P. A. Seeger, "The MCLIB Library: Monte Carlo Simulation of Neutron Scattering Instruments," Proceedings of ICANS-XIII, Paul Scherrer Institut, Villigen, Switzerland, Oct. 11-14, 1995.

[2] M. W. Johnson and C. Stephanou, "MCLIB: A Library of Monte Carlo Subroutines for Neutron Scattering Problems," Rutherford Laboratory report RL-78-090.

[3] P. A. Seeger and R. Pynn, "Resolution of Pulsed-Source Small-Angle Neutron Scattering," Nuclear Instruments and Methods in Physics Research A245 , 115-124 (1986).

[4] P. A. Seeger and R. P. Hjelm, "A Very-Low-Q-Diffractometer for an Advanced Spallation Source," Proceedings of ICANS-XII, Abingdon, U.K., May 24-28, 1993.

[5] P. A. Seeger and R. P. Hjelm, "Design and Implementation of Low-Q Diffractometers at Spallation Sources," American Crystallographic Association Symposia Proceedings, Albuquerque, NM, May 24-28, 1993.

[6] G.A. Olah, R.P. Hjelm and P.A. Seeger, "Small-angle Scattering Instrumentation on a 1 MW Long Pulse Spallation Source", Proceedings of ICANS-XIII, Paul Scherrer Institut, Villigen, Switzerland, Oct. 11-14, 1995.

[7] M.R. Fitzsimmons and R. Pynn, "the Performance of Reflectometers at Continuous Wave and Pulsed Neutron Sources", Proceedings of ICANS-XIII, Paul Scherrer Institut, Villigen, Switzerland, Oct. 11-14, 1995.

[8] L.L. Daemen, J. Eckert., R. Pynn and P.A. Seeger, "Monte Carlo Study of the Performance of a Time-of-Flight Multichopper Spectrometer", Proceedings of ICANS-XIII, Paul Scherrer Institut, Villigen, Switzerland, Oct. 11-14, 1995.