Simulation tools for detector and instrument design

Kanaki, Kalliopi; Kittelmann, Thomas; Cai, Xiao Xiao; Klinkby, Esben Bryndt; Knudsen, Erik B.; Willendrup, Peter Kjær; Hall-Wilton, Richard

Published in:
Physica B: Condensed Matter

Link to article, DOI:
10.1016/j.physb.2018.03.025

Publication date:
2018

Document Version
Version created as part of publication process; publisher's layout; not normally made publicly available

Link back to DTU Orbit

Citation (APA):
Simulation tools for detector and instrument design

Kalliopi Kanaki a,*, Thomas Kittelmann a, Xiao Xiao Cai a, b, Esben Klinkby b, a, Erik B. Knudsen b, Peter Willendrup b, a, Richard Hall-Wilton a, c

a European Spallation Source ERIC, 22-100, Lund, Sweden
b Technical University of Denmark, DTU, 2800 Kgs. Lyngby, Denmark
c Mid-Sweden University, SE-851 70 Sundsvall, Sweden

ARTICLE INFO

Keywords: Crystals, File formats, Monte Carlo simulations, Neutron detector, Neutron scattering

ABSTRACT

The high performance requirements at the European Spallation Source have been driving the technological advances on the neutron detector front. Now more than ever it is important to optimize the design of detectors and instruments, to fully exploit the ESS source brilliance. Most of the simulation tools the neutron scattering community has at their disposal target the instrument optimization until the sample position, with little focus on detectors. The ESS Detector Group has extended the capabilities of existing detector simulation tools to bridge this gap. An extensive software framework has been developed, enabling efficient and collaborative developments of required simulations and analyses – based on the use of the Geant4 Monte Carlo toolkit, but with extended physics capabilities where relevant (like for Bragg diffraction of thermal neutrons in crystals). Furthermore, the MCPL (Monte Carlo Particle Lists) particle data exchange file format, currently supported for the primary Monte Carlo tools of the community (McStas, Geant4 and MCNP), facilitates the integration of detector simulations with existing simulations of instruments using these software packages. These means offer a powerful set of tools to tailor the detector and instrument design to the instrument application.

1. Introduction

The neutron scattering community has been investing a large effort in developing new detector technologies that can tackle the needs of the upcoming European Spallation Source (ESS) [1]. The detector designs are diverse and target different sets of requirements [2, 3], e.g. high spatial resolution [4, 5], high rate capability [5, 6], large area coverage [7] or combinations thereof. Additionally, as instrument performance is typically defined by the signal to background ratio [8], the ability to predict and improve instrument backgrounds will enhance the capability of future instruments. To facilitate and accelerate the design process, the use of Geant4 [9–11] has been adopted. It is a powerful Monte Carlo simulation toolkit for the description of particle passage through matter, used by several scientific communities since decades. Specifically it provides step-based particle simulations in arbitrarily complex geometrical layouts, and with physics modeling capabilities extending beyond thermal neutrons to include also other particle types and a wide range of electromagnetic and hadronic processes – features all essential for any simulation of detector performance.

In recent years the capabilities of Geant4 have been extended to include an increasing number of neutron-related phenomena at lower energies. The ESS Detector Group continues this effort by creating easily integrable tools, modeling Bragg and inelastic processes in crystalline materials. This way, it is becoming possible to expand the relevant functionality of Geant4 and other software packages utilized by the neutron scattering community. A selection of these tools is presented in this article with focus on their functionality and not their technical implementation. The intention of these tool-sets is two-fold: firstly to enhance the capability and accuracy of the simulations; secondly to lower the entry barrier to utilization of the simulation codes and ensure their correct usage with modern code management and validation of the code and standard results.

2. The NCrystal project

A large fraction of the instrument components consists of crystalline materials, which makes it crucial for simulations to correctly model interactions of thermal neutrons in such materials. One of the first
Fig. 1. Back-scattered neutrons at the rear end of the Multi-Grid detector module (a) (from Ref. [14]) can be reproduced with Geant4 (b) (from Ref. [15]), provided the coherent scattering for the crystalline Al detector frame is enabled via NXSG4.

Fig. 2. Total neutron cross sections for Germanium powder as a function of neutron wavelength, overlaid with the Bragg and background contributions, provided by NCrystal.
3. The MCPL file format

Motivated by the need for simulation packages to efficiently exchange particle data, the ESS Detector Group developed the Monte Carlo Particle List (MCPL) file format [21]. It is a well-defined, binary format containing full particle state information. Interfaces are available for Geant4, McStas 2.4.1, McXtrace 1.4 [22], MCNP5 [23], MCNP6 and MCNPX, so that the communication between these packages is facilitated through a single standardized format. MCPL comes with C/C++/Python bindings that allow easy integration with software packages. It also constitutes a convenient way for storage of particle state information even for users of just a single software application. The stand-alone code can be downloaded from its GitHub repository [24], is open source and released under very liberal license conditions. The power of the MCPL format lies in its flexibility and efficient implementation but also in the utilities it ships with. Easy histogram visualization, file merging, filtering and modification are available via simple terminal commands [25]. The MCPL format allows the most appropriate or familiar simulation code to be used by the user for the application foreseen.

As a show case for the MCPL use, the interfacing of a McStas instrument simulation to a Geant4 detector simulation is presented here. As a part of the detector and instrument optimization processes, it is beneficial to use customized input for the detector simulation, e.g. a realistic distribution of neutrons scattered from a typical sample for the particular application. Such a scenario is shown in Fig. 5. Neutrons are simulated in a Small Angle Neutron Scattering instrument in McStas until after their scattering on 200 Å radius spheres. These neutrons are then saved with all their properties in the MCPL format and used as input to a Geant4 simulation, which contains a detailed detector model. This way, the user is able to look at the interesting scientific quantities both at the sample and after the detection or any other stage of the simulation. In this particular example, Fig. 5 depicts the Q intensity that contains instrument and sample effects in blue, while in red appears the result of the Geant4 detector simulation, which additionally convolves the detector effects.

4. The ESS simulation framework

It is customary for large scientific communities to create software frameworks, which provide support and utilities to all community members. These should incorporate modern software coding tools such as a versioned repository, documentation, issue tracking and validation [26–29]. Such a simulation framework [30] has been developed by the ESS Detector Group and is used at the moment within ESS and by in-kind collaborators in Europe. It facilitates a quick setup of simulation and analysis of new projects, reducing thus the overhead required to start a new detector simulation. Users can benefit from centralized support and utilities that are seamlessly integrated. To mention just a few of the latter, the framework ships with a user-friendly, dynamic build system, a flexible Python interface for combining geometries and particle generators, a dedicated 3D viewer based on OpenSceneGraph [31] that allows visualization and quick geometry debugging (see Fig. 6), a customized binary file format (GRIFF) [30] for efficient storage of simulation output and accompanying meta-data, and finally NumPy-compatible histogram classes to assist the user with the analysis of the results. In addition to these utilities, all the aforementioned libraries
**References**

