Available online at www.sciencedirect.com
ScienceDirect
Journal of Radiation Research and Applied Sciences
journal homepage: http://www.elsevier.com/locate/jrras

ERSN-OpenMC, a Java-based GUI for OpenMC Monte Carlo code

Jaafar EL Bakkali [a,*], *Tarek EL Bardouni* [a], *Seyedmostafa Safavi* [b], *Hamid Boukhal* [a], *Mohamed Kaddour* [a], *Khalid Benaalilou* [a], *Essaid Chham* [a]

[a] *ERSN/LMR, University Abdelmalek Essaadi, Faculty of Sciences, Tetouan, Morocco*
[b] *Cyber Security Unit at the National University of Malaysia, Malaysia*

ARTICLE INFO

Article history:
Received 21 September 2015
Received in revised form
2 November 2015
Accepted 9 December 2015
Available online 21 December 2015

Keywords:
Neutronic
Monte Carlo
k-Eigenvalue criticality
OpenMC
Java
GUI

ABSTRACT

OpenMC is a new Monte Carlo transport particle simulation code focused on solving two types of neutronic problems mainly the k-eigenvalue criticality fission source problems and external fixed fission source problems. OpenMC does not have any Graphical User Interface and the creation of one is provided by our java-based application named ERSN-OpenMC. The main feature of this application is to provide to the users an easy-to-use and flexible graphical interface to build better and faster simulations, with less effort and great reliability. Additionally, this graphical tool was developed with several features, as the ability to automate the building process of OpenMC code and related libraries as well as the users are given the freedom to customize their installation of this Monte Carlo code. A full description of the ERSN-OpenMC application is presented in this paper.

1. Introduction

Monte Carlo methods have been widely used in different areas of physics particularly in nuclear physics applications. Some of its applications are as follows: energy sources, nuclear forensics, imaging and diagnostics, radiation treatment and material science. Nowadays, there are several Monte Carlo codes: Geant4 (Agostinelli et al., 2003), MARS (Mokhov, 1995) and FLUKA (Ferrari, Sala, Fassoand, & Ranft, 2005), which have been developed to simulate a wide range of nuclear physics. On the other hand, there are other ones, which are currently in development and are relatively nascent such as OpenMC Monte Carlo simulation code (Romano & Forget, 2013).

Starting in 2011, this new Monte Carlo code was created and spearheaded by the members of the Computational Reactor Physics Group (CRPG), at the Massachusetts Institute of Technology. Although it has high-reliability factor enabling the modeling of nuclear reactors and others fissile nuclear systems. A major drawback of this code is that it isn't a user-friendly application (as it's only a shell program and it lacks a graphical user interface).

The OpenMC code uses a solid modeling technique known as constructive solid geometry (CSG) to build complex three-

[*] *Corresponding author.*
E-mail address: bahmedj@gmail.com (J. EL Bakkali).
Peer review under responsibility of The Egyptian Society of Radiation Sciences and Applications.

http://dx.doi.org/10.1016/j.jrras.2015.12.002

dimensional objects in Euclidean space. This is achieved by using Boolean operators. It uses the concept of lattice and universes that are implemented in MCNP and SERPENT to model repetitive geometry, making the code more structured and thereby simplifying user's work. The OpenMC code does not have its own standalone visualization system. Hence it uses external packages to visualize the overall geometry as well as the output data of a given simulation. Currently there are two different plotting capabilities implemented in OpenMC. The first one is a 2D slice plotting system, which allows users to create geometry along plan cut, and it can be visualized using a PPM viewer. The second one is a 3D voxel plotting system, allowing user to create a 3D plotting file of their geometry. The generated file is an arbitrary binary voxel format which can be converted to VTK format in order to be visualized by one of powerful viewers (as ParaView or VisIt). The plotting of geometry is carried out by creating a plot.xml optional input file and running OpenMC executable in plotting mode with –plot or -p command-line flag.

Similar to other Monte Carlo codes, OpenMC reads continuous-energy cross sections from Compact ENDF (ACE) data libraries, which can be generated with the NJOY nuclear data processing code. Since this data format is shared with other Monte Carlo codes (such as MCNP and SERPENT), any ACE format data library generated for these codes can be used with OpenMC.

Unlike other Monte Carlo codes, which use an arbitrary-format ASCII file to specify the input files, in OpenMC each input file has a modern and portable input format with structured content in a set of XML data. The structure of each file is defined by the corresponding OpenMC XML schema. The syntax of OpenMC xml language can be easily understood with basic knowledge of XML and regular expressions. The main advantage of using OpenMC is reduced time when writing the components of simulation. Thus, the users do not need to re-compile and re-link any of their applications if they made any amount of changes. The OpenMC code can produce a modern and portable binary output format, by using the HDF5 library, which allows this code to faster write the output data.

The OpenMC code can resolve two types of neutronic problems the k-eigenvalue criticality fission source and the external fixed fission source problems during Monte Carlo calculations. The major problem encountered in the k-eigenvalue neutronic problem is the criticality calculation of the convergence of the fission source. To solve this problem, several methods have been implemented in various Monte Carlo codes – like extrapolation method (Toth & Griesheimer, 2007), matrix method (Carney, Brown, Iciedrowski, & Martin, 2012), Weilan method (Yamamoto & Miyoshi, 2004) and the Croash Mesh Finite Difference (CMFD) method (Lee, Joo, Lee, & Smith, 2010). The last one is used by OpenMC code as the main factor to determine the source convergence acceleration. To enable CMFD acceleration, the OpenMC uses an external package called PETSc (Portable, Extensible Toolkit for Scientific Computation).

The present paper details the development of a simple and highly flexible java-based GUI Front-End application, which is programmatically wrapped around OpenMC. It is only available as shell program which can be accessible through command line interface (CLI). This java-based application has been designed and implemented to automate efforts needed for compiling and installing OpenMC code and its pre-requisites packages along with creating, modifying and validating a Monte Carlo simulation with this code.

2. Software architecture

The source of ERSN-OpenMC software was written using Java object-oriented language. Moreover, a few Linux bash scripts are also developed and are invoked directly by the ERSN-OpenMC application, to make particular tasks. The reason we have chosen to create a java-based application is because it can be run on various platforms such as Linux, MS-Window and Mac OS X, and there is a Java virtual machine for almost every platform. In addition, Java provides a very rich and powerful API producing the most portable and flexible applications. The GUI application described in this paper was developed under NetBeans IDE. This GUI application calls directly a number of external programs. Some of these programs are as follows: openMC which will be used for neutron calculations engine, xterm for application console, eog (Eyes of GENOME) for viewing PPM image and Paraview for viewing 3D model and plotted data.

The GUI source is divided into four major parts as follows:

2.1. *Java sources*

main.java: the main window of ERSN-OpenMC java-based GUI application.

new_project.java: a dialog box to create a new OpenMC project.

project_three.java: an OpenMC project explorer view.

isotopes_tables.java: a dialog window showing a periodic table and lets users generate XML code for multiple isotopes or natural elements selected by them.

scorers.java: a dialog window for generating XML code of list of desired physical quantities to be scored, such as total flux, total reaction rate, etc.

output.java: a dialog window contains tab based pane displaying three kinds of output data: tallies, summary and cross sections. The user can choose which one to be viewed, by selecting the tab which corresponds to particular involved output data.

rbg_color.java: a dialog window used for generating XML code of a custom RBG color for a given cell or a given material.

run_openmc.java: a dialog box which lets users to select the run options they wish to use and run the OpenMC executable.

get_*openmc.java:* a dialog window for getting OpenMC code and it prerequisites libraries.

2.2. *Linux bash scripts*

get_openmc.sh: gets the last version of OpenMC from OpenMC official website, compiles its source in sequential mode.