Development of Monte Carlo code for the estimation of neutron multiplication factor in fissile materials

Sachin Shet[1] K.V. Subbaiah[*1] S. Anand[2] and Kapil Deo Singh[2]

[1]Manipal Centre for Natural Sciences, Centre of excellence,
Manipal Academy of Higher Education (MAHE), Manipal - 576104, Karnataka, India.
[2]Health Physics Division, Bhabha Atomic Research Centre, Mumbai 400085, India.
kv.subbaiah@manipal.edu

Abstract. Uranium and Plutonium are the elements that undergo neutron-induced fission and are used as nuclear fuels for power production. Fissile isotopes of these elements which are in solid or liquid form are processed in nuclear fuel cycle facilities to obtain the required purity. As these materials are arranged in various arbitrary geometry manner unlike the reactor geometry, the criticality safety during the process of fissile material is very essential and it needs the precise estimation of neutron multiplication factor (k_{eff}) of the system. A new Monte Carlo (MC) code namely Monte Carlo for Neutron Simulation of k_{eff} (MCNS_k_{eff}) has been developed indigenously for the estimation of k_{eff} of nuclear fuel cycle facilities incorporating the continuous energy neutron cross sections. MCNS_k_{eff} has been developed with several desirable features, which make the code faster and user-friendly. In the present work, the code has been tested against a variety of benchmark problems and for other incorporated features. The details of the development of the code along with the benchmark results are presented.

Keywords: Monte Carlo, Neutron Transport, Criticality Safety

1 Introduction

Fissile isotopes of Uranium and Plutonium are used in nuclear industries for power generation. Improper handling or storage of fissile materials may lead to criticality accidents [1] resulting in the release of a large quantity of energy thereby causing damage to the systems and the surroundings. In general, such systems are designed with a lot of safety margin causing the economic burden on the projects. To circumvent the cost of the project, it is necessary to estimate precisely the neutron multiplication factor (k_{eff}) of the system. Although many methods of calculating k_{eff} are in vogue. Monte Carlo (MC) methods (MCNP [2], OpenMC [3]) are preferred due to the capability of these methods in handling complex arrangement of geometrical shapes and in accounting detailed physics. The basic of MC method as applicable to neutron transport is the sampling of events randomly that may happen with nuclei while neutron traversing through the material media. The computer codes based on the MC procedure that exists today need more of Central Processing Unit (CPU) time and

computer memory prohibiting the usage in general. However, rapid developments both in software and hardware of computers, in addition to the development of new algorithms in MC procedures attracting the users for solving variety of problems. To reduce CPU time (a process which consumes more CPU time identified to be search algorithm in large arrays), new algorithms are being used such as binary search method and logarithmic energy grid method besides making use of parallel computers with multiple nodes. Further to limit the memory usage dynamic allocation is adopted avoiding the reservation of large fixed memory. In the present work, the MC code namely Monte Carlo for Neutron Simulation of k_{eff} (MCNS_k_{eff}) has been developed in house incorporating the above-advanced features for the estimation of neutron multiplication factor in fissile materials configured in complex geometrical shapes.

2 Methods

Neutron multiplication factor of a system is defined as the ratio of the number of neutrons produced in a generation to that of number started in the previous generation of the system. To estimate neutrons produced, it is necessary to account all predominant modes of interaction of neutrons with the nuclei present in the system. Generally, neutron interactions with nuclei are elastic scattering, inelastic scattering, radiative capture, nuclear fission, and particle emission reactions such as (n,2n), (n,3n), (n,d) and (n,p) etc. The interaction probabilities (cross sections) depend on the neutron energy and these exhibit sharp resonances demanding the tabulation at very close energy intervals. Such tabulations are known as continuous-energy cross section data or pointwise data (linearly interpolable between the points) available as ACE files (A Compact ENDF). Such data generated using NJOY [4] code for several commonly used nuclides available and is used for testing the code developed. In addition to tabulated cross sections for each of the reaction, secondary angle and energy distributions are also provided.

Geometric cells bounded by analytical surfaces containing materials are specified by combining the surfaces with the use of binary operators such as union, intersection, and complement. It is possible to construct any complex configurations of materials with a limited number of surfaces. In the present code, we use 21 basic surfaces of first and second degree which are enough to describe any conceivable geometry.

k_{eff} of a system is calculated iteratively starting with a finite number of neutrons in cells containing fissile materials (source points) and tracking them from its birth to its death through the system while scoring the events of interest namely fission. Although an event may not have any physical meaning, the average over a collection of a large number of events represents the behavior of the system, which is the required answer for the problem. In MC simulation birth of a neutron is taken to be the source point or fission site, and death of a neutron taken to be either absorption or escape from the system. At the end of tracking of a predetermined number of neutrons, k_{eff} is estimated and fission sites are stored for starting subsequent iterations. The iterations are stopped when the value of k_{eff} of a system converges. To ensure that the neutron population does not grow exponentially, the number of fission sites are controlled if

found in excess and sampled over and again if the number of fission sites is less than the predetermined number.

3 Details of development

The MCNS_k_{eff} is cross-platform code developed for both Linux and Windows operating systems (OS) written in Fortran language with Fortran 2008 features. The notable features are:

- Input file – divided as 3 blocks of data and the code accepts in a free format
- Debug feature – input is processed and if found any warning and fatal error messages are displayed
- Parallel mode run – By default, MCNS_k_{eff} uses all the available number of threads in a computer.
- A user can run with the specified number of threads with the use of command line arguments
- Continue Run –Large CPU problems can be broken up into smaller ones and can be run as restart runs.
- S (α, β) treatment - For neutrons energies, less than 4 eV, the free gas treatment of kinematics is not valid. Scattering kinematics is different for crystalline solids and bound molecules and these are taken care of by using S (α, β) cross-section tables
- Unresolved resonance region probability tables – To account for the presence of a large number of resonances in the energy region (keV), probability table method is employed.
- Repeated structure handling – Input can be specified in terms of universes and lattices, which can be repeated at ease.
- Geometry viewer –Interactive of 2D geometry plots can be made to identify geometry errors. A typical 2D plot of a rectangular lattice is shown in Fig.1

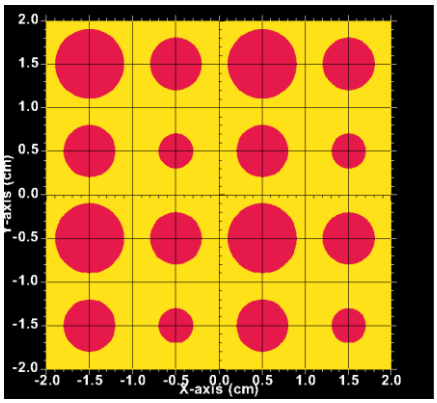

Fig. 1. 2D plot of rectangular lattice with three different universes (red-fuel, yellow-water)

4 Results

MCNS_k_{eff} has been indigenously developed in Fortran language [2008] for operating systems both for Linux and Windows. To validate all the features of the code, a variety of criticality benchmark problems are chosen representing solid, liquid fuels and their arrangement in terms of lattices. k_{eff} can be estimated by three different scoring functions namely absorption, collision and tracklength:

$$\text{Collision estimator} \quad k_{eff}^{c} = \frac{1}{N}\sum_i W_i \left[\frac{\Sigma_k f_k \upsilon_k \sigma_{f_k}}{\Sigma_k f_k \sigma_{T_k}}\right]$$

$$\text{Absorption estimator} \quad k_{eff}^{A} = \frac{1}{N}\sum_i W_i \, \upsilon_k \frac{\sigma_{f_k}}{\sigma_{c_k} + \sigma_{f_k}}$$

$$\text{Tracklength estimator} \quad k_{eff}^{TL} = \frac{1}{N}\sum_i W_i \, \sigma d \sum_k f_k \upsilon_k \sigma_{f_k}$$

The above equations contain standard symbols and details are given in Reference 2. By combining these 3, the average k_{eff} of the system is estimated which will have a less standard deviation. All the 3 k_{eff} for Godiva reactor obtained with present code versus cycle number is shown in Fig.2 along with one standard deviation [σ]. The σ is very large at the beginning of cycles indicating the system has not reached its equilibrium state. The σ softens off after about 100 cycles indicating the convergence of k_{eff} value beyond this point.

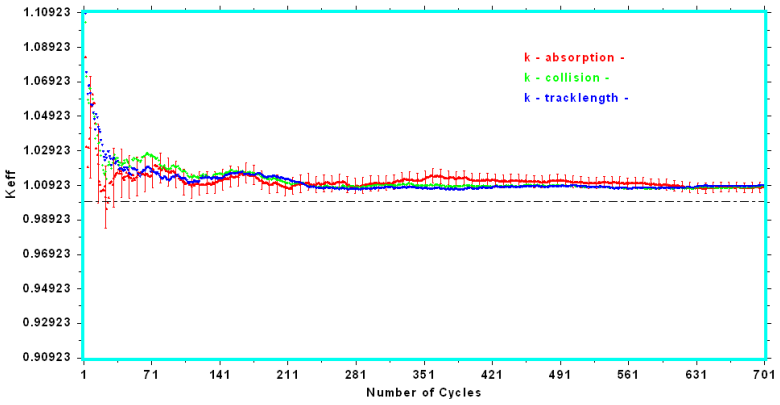

Fig. 2. k_{eff} vs Cycle for Godiva reactor assembly

For validation of the present results, the values reported in literature/generated using the existing standard codes are compared for several problems. Table 1 compares the typical results of average k_{eff} obtained from the present code with that of MCNP using the same cross section data and identical physics parameters, which has been validated by many other workers.

Table 1. Comparison of k_{eff}

Problem		Problem ID	MCNS_k_{eff}	MCNP
1. Godiva Reactor[solid-sphere]		HEU-MET-FAST-001 [6]	1.0070 ± 0.0006	0.9995 ± 0.0005
2. UO_2F_2 Solution [liquid Cylinder]	S (α, β)	Problem number 3 [7]	0.9967 ± 0.0008	0.9984± 0.0007
	No S (α, β)		0.9982 ± 0.0008	1.0013 ± 0.0007
3. Lattice [Rectangular]		Square lattice [8]	0.05956 ± 0.0001	0.05983 ± 0.0001
4. Lattice [Hexagonal]		Problem Number 6 [7]	0.0121 ± 0.0002	0.0114 ± 0.0001

From the table, it can be inferred that the values of k_{eff} for all the problems chosen are in excellent agreement with each other.

5 Conclusions

A new Monte Carlo code has been developed indigenously with several new features (Fortran 2008) to estimate k_{eff} of fissile systems. The new code MCNS_k_{eff} has been tested with several criticality benchmarks problems against standard code MCNP and shows excellent agreement.

6 References

1. Knief, Ronald Allen. "Nuclear criticality safety." Am. Nucl. Soc., Illinois (1985).
2. J.F. Briesmeister, Ed., “MCNP – A General Monte Carlo N-Particle Transport Code – Version 4C,” LA-13709-M, Los Alamos National Laboratory (March 2000).
3. Paul K. Romano, Nicholas E. Horelik, Bryan R. Herman, Adam G. Nelson, Benoit Forget, and Kord Smith, “OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development,” Ann. Nucl. Energy, 2015
4. R.E. McFarlane &D.W. Muir: The NJOY Nuclear Data Processing System, Version 91. LA-12740-M (October 1994)
5. DISLIN – Fortran graphics library - Max Planck Institute for Solar System Research – 2017.
6. Russell D. Mosteller et al, “An Expanded Criticality Validation Suite for MCNP”, International Conference on Nuclear Criticality 2011Edinburgh, Scotland September 19-22, 2011
7. Harmon, C.D. II, Busch, R.D., Briesmeister, J.F., & Forster, R.A. (Jun 1994). Criticality calculations with MCNP trademark: A primer (LA-SUB--94-108). United States.
8. The OpenMC Monte Carlo Code. Available: https://github.com/openmc-dev/openmc/tree/develop/examples/xml/lattice/simple

Acknowledgment. This work was supported by Department of Atomic Energy – Board of Research in Nuclear Sciences project reference no: 36(4)/14/40/2015/36007.