Annals of Nuclear Energy 100 (2017) 128–149

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

On the calculation of angular neutron flux in MCNP

Jure Beričič, Luka Snoj *

"Jožef Stefan" Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 18 March 2016
Received in revised form 1 September 2016
Accepted 19 October 2016

Keywords:
Angular flux
Neutron flux
Neutron transport
Monte Carlo
MCNP
Computer code
Solid angle partitioning

ABSTRACT

Modern Monte Carlo neutron transport codes offer many options for neutron flux and spectra calculations, however, they often lack the option to obtain the angular neutron flux in a region of the problem. The angular flux can also be obtained from deterministic programs, however, it includes biases due to discretization and other physical approximations. Therefore, a novel method for determining the angular neutron flux from the standard output of the MCNP is proposed in this paper. The method was also implemented as a set of Python libraries and tested in several examples. The results were then used to investigate the self-shielding effect in a realistic angular profile of the flux, i.e., the TRIGA research reactor.

1. Introduction

The angular neutron flux presents a complete information about a population of neutrons. It depends on the position in space $\vec{r}$, the direction of movement $\vec{\Omega}$, the neutron energy E and the time t. It is basically seven dimensional and as such a relatively complex quantity. The practical usefulness of the angular neutron flux depends on the problem at hand and on the method used to solve it. Deterministic codes for neutron transport like PARTISN (Alcouffe et al., 2008) directly solve transport equations for angular flux. On the other side there are Monte Carlo neutron transport codes such as MCNP (Goorley et al., 2012) which transport each neutron separately and the angular flux is not calculated explicitly. Some of the codes, like TRIPOLI (TRIPOLI-4 Project Team, 2013) and KENO (Goluoglu et al., 2011), also provide solid angle binning for flux tallies. TRIPOLI offers track length and collision estimators for volume flux tallies. The angular flux can be determined on a custom defined $(\cos(\theta), \phi)$ angular grid, where the angles are given in the global frame of the geometry. In KENO, angular fluxes are computed with a track length estimator for a symmetric level quadrature set (see also Section 5.1). There are also efforts to combine the two approaches, like the TORT/MCNP coupling (Kurosawa, 2005) or the ADVANTG (Mosher et al., 2015). On the experimental side, it is difficult to measure the angular neutron flux in general. There were, however, several experiments performed measuring the angular neutron flux with a time-of-flight technique for slabs of different materials (Oyama et al., 1988, 1992).

While the information about angular neutron flux is not explicitly needed in the Monte Carlo codes it can still prove to be useful by generating additional insight into the problem under study, especially for neutron streaming calculations (Snoj et al., 2012). In addition, the Monte Carlo calculation of angular neutron flux can be used as a reference or benchmark calculation for deterministic methods. This is why a method for calculating the angular neutron flux from the standard MCNP output is proposed in this work. The method was also implemented as a set of Python libraries and tested in several examples. Moreover, the proposed method can be directly used in the MCNP and a new tally, e.g., the track length estimator of angular neutron flux, could be implemented into the MCNP code.

This paper is structured as follows. Section 2 describes the basics of the angular flux calculations. Section 3 presents the current MCNP capabilities with regards to the angular flux. In the Section 4 a track length estimator for flux calculation is introduced. Section 5 presents our method for determining the angular neutron flux, including the partitioning of the solid angle, track length calculation for few representative geometries, tallying procedure, and two visualization techniques. Section 6 discusses the results obtained with this method in case of a graphite and water block, and for various positions inside the JSI TRIGA research reactor. Lastly, in the Section 7 the application of the angular neutron flux calculation to the problem of sample self-shielding is presented.

* Corresponding author.
E-mail address: luka.snoj@ijs.si (L. Snoj).

http://dx.doi.org/10.1016/j.anucene.2016.10.014
0306-4549/

2. Angular flux

The neutron population can be described by the angular neutron density $N(\vec{r}, \vec{\Omega}, E, t)$, where the expression $N(\vec{r}, \vec{\Omega}, E, t)\,\mathrm{d}V\,\mathrm{d}\vec{\Omega}\,\mathrm{d}E$ measures the expected number of neutrons in a volume element dV about $\vec{r}$ having direction within $\mathrm{d}\vec{\Omega}$ about $\vec{\Omega}$ and energies in dE about E. Similarly, an angular neutron flux can be used, which is defined as Bell and Glasstone (1970):

$$\Phi(\vec{r}, \vec{\Omega}, E, t) \equiv v(E) N(\vec{r}, \vec{\Omega}, E, t). \tag{1}$$

The average flux inside a cell of volume V is given by Goorley et al. (2012):

$$\bar{\Phi}_V = \frac{1}{V} \int \mathrm{dE} \int \mathrm{dV d\Omega} \int \mathrm{dt\, v(E) N(\vec{r}, \vec{\Omega}, E, t)} \tag{2a}$$

$$= \frac{1}{\mathrm{V}} \int \mathrm{dE} \int \mathrm{d\Omega} \int \mathrm{dV} \int \mathrm{ds\, N(\vec{r}, \vec{\Omega}, E, t)}, \tag{2b}$$

where a relation d$s = v$dt was used in the second line.

The units of the angular flux Φ are $(\mathrm{s} \cdot \mathrm{cm}^2 \cdot \mathrm{sr} \cdot \mathrm{MeV})^{-1}$. However, the units of the average flux $\bar{\Phi}_V$ are $1/\mathrm{cm}^2$ due to the averaging over the cell volume and integrating over energy, time and solid angle.

3. MCNP and angular distributions

MCNP does not currently feature any option to directly assess the angular neutron flux. The scalar neutron flux as estimated by the track length estimator (F4 tally in MCNP) can be binned in energy and time, but not in solid angle. On the other hand, the surface current and surface flux estimators (F1 and F2 tallies in MCNP, respectively) also allow binning in cosine of the angle between particle direction and surface normal. However, it is of limited use. One can only bin in one angle and not two. Additionally, the binning angle is relative to the surface normal. When tallying on the curved surface, or two surfaces with different normals, the information about the absolute direction along which the particle is moving in is lost.

4. Flux tally

One method for calculating the scalar flux tally in MCNP is based on Eq. (2b), just using neutron density $N(\vec{r}, E, t)$ instead of angular neutron density (Goorley et al., 2012):

$$\bar{\phi}_V = \frac{1}{V} \int \mathrm{d}E \int \mathrm{d}V \int \mathrm{d}s\, N(\vec{r}, E, t). \tag{3}$$

The average flux is then estimated by the sum:

$$\bar{\phi}_V = \frac{1}{NV} \sum_{i=1}^{N} W_i T_l^i, \tag{4}$$

where W is the weight of the event, T_l is the track length of the neutron inside the cell and i runs over all neutron histories. The flux can be further divided into time and energy bins. The population variance for large N is given by Goorley et al. (2012):

$$S^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(\frac{W_i T_l^i}{V} - \bar{\phi}_V \right)^2 \approx \bar{\phi}_V^2 - \bar{\phi}_V^2, \tag{5a}$$

$$\bar{\phi}_V^2 = \frac{1}{NV} \sum_{i=1}^{N} \left(W_i T_l^i \right)^2. \tag{5b}$$

The estimated standard deviation of the $\bar{\phi}_V$ is then:

$$s_{\bar{\phi}} = \sqrt{\frac{S^2}{N}}. \tag{6}$$

It is important to note that in the MCNP the values $W_i T_l^i / V$ in Eqs. (4) and (5) are the total contribution from the ith starting particle and all resulting progeny. In each particle history, there may be multiple contributions to a tally bin, but they are correlated because they come from the same starting particle.

5. Angular flux tally

Even if the MCNP does not feature an option to tally the angular flux by itself, all the necessary information for calculating the angular neutron flux are provided in the particle track log file, containing the particle track information for each event of each history of the simulation. This is invoked by using the so-called `PTRAC` card in MCNP. Among other, the following information is available (Goorley et al., 2012):

- position of the event (e.g. collision),
- weight of the event,
- energy of the neutron,
- direction vector of the neutron.

The authors propose a method for tallying the angular neutron flux by using the above information. The procedure is analogous to the one used in the MCNP code for the scalar flux tally (see Section 4). The main difference with the scalar flux tally is the additional binning of the solid angle, hence the procedure is based on Eq. (2b) instead of Eq. (3).

A set of Python libraries was also developed to read the `PTRAC` files and tally the angular neutron flux. Details about the solid angle partitioning used, track length calculation and tallying are given in the subsequent subsections.

5.1. Solid angle partitioning

In order to make a binning in $\vec{\Omega}$, a partitioning of the solid angle or unit sphere must be selected. There are several schemes available, some of which are described in the text below. In this paper the EQ sphere partitioning algorithm by Leopardi (2006) is used.

5.1.1. Celestial sphere

An icosahedron-based method for pixelating the celestial sphere was developed for storing and analyzing sky maps obtained at astronomical and cosmological observations. The method consists of three steps (Tegmark, 1996). First, the sphere is inscribed in an icosahedron, whose faces are pixelized with a regular triangular grid. These points are then mapped radially onto the sphere. In the third step, the points are shifted around slightly to give all pixels approximately equal area. This results in a pixelation as shown in Fig. 1.

5.1.2. Goldberg polyhedra

A special family of convex polyhedra was described by Goldberg (1937) which can also be used to partition the solid angle. Goldberg polyhedra are composed of twelve regular pentagons and a number of hexagons. They can be denoted as $G(m, n)$, where m and n describe the form of the polyhedra. To go from one pentagon to the next a "chess knight move" must be taken: first take m steps along a direction, then turn for 60° counter-clockwise and take n steps. The simplest Goldberg polyhedron is a dodecahedron. An example of $G(4, 4)$ is shown in Fig. 2.