Annals of Nuclear Energy 114 (2018) 437–450

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

An angular biasing method using arbitrary convex polyhedra for Monte Carlo radiation transport calculations

Joel A. Kulesza [a,b,*], Clell J. Solomon Jr. [b], Brian C. Kiedrowski [a]

[a] University of Michigan, Department of Nuclear Engineering & Radiological Sciences, 2355 Bonisteel Boulevard, Ann Arbor, MI 48109, United States
[b] Los Alamos National Laboratory, Monte Carlo Methods, Codes, and Applications Group, PO Box 1663, Los Alamos, NM 87545, United States

ARTICLE INFO

Article history:
Received 18 October 2017
Received in revised form 7 December 2017
Accepted 15 December 2017
Available online 2 January 2018

Keywords:
MCNP6
DXTRAN
Angular biasing
Convex hull

ABSTRACT

This paper presents a new method for performing angular biasing in Monte Carlo radiation transport codes using arbitrary convex polyhedra to define regions of interest toward which to project particles (DXTRAN regions). The method is derived and is implemented using axis-aligned right parallelepipeds (AARPPs) and arbitrary convex polyhedra. Attention is paid to possible numerical complications and areas for future refinement. A series of test problems are executed with void, purely absorbing, purely scattering, and 50% absorbing/50% scattering materials. For all test problems tally results using AARPP and polyhedral DXTRAN regions agree with analog and/or spherical DXTRAN results within statistical uncertainties. In cases with significant scattering the figure of merit (FOM) using AARPP or polyhedral DXTRAN regions is lower than with spherical regions despite the ability to closely fit the tally region. This is because spherical DXTRAN processing is computationally less expensive than AARPP or polyhedral DXTRAN processing. Thus, it is recommended that the speed of spherical regions be considered versus the ability to closely fit the tally region with an AARPP or arbitrary polyhedral region. It is also recommended that short calculations be made prior to final calculations to compare the FOM for the various DXTRAN geometries because of the influence of the scattering behavior.

1. Introduction

This paper presents a new method for performing angular biasing into a region of interest defined with an arbitrary three-dimensional convex polyhedron (hereafter polyhedron with 3-D and convex implied) in Monte Carlo radiation transport codes. Angular biasing is particularly useful when a tally exists at a position that has a low probability of being sampled by particles because of a low probability of the particles scattering into the vicinity of the tally. One such case is a radiation source and distant detector in an urban environment, or more generally, scenarios dominated by particle streaming. In these cases, sampling near the tally is increased by deterministically drawing weight toward a region of interest around the tally and then continuing Monte Carlo tracking. This work introduces a new method for defining such regions of interest using polyhedra so that particles can be drawn as close to the tally as possible.

Angular biasing capabilities already exist for several Monte Carlo radiation transport codes. Only spherical regions of interest (referred to as DXTRAN regions because they *deterministically transport* particles to the region of interest) can be defined in MCNP6 as of version 6.2; however, other DXTRAN geometries have been suggested by Booth (1986, 1994); Gardner et al. (1987). Hereafter, angular biasing regions of interest will be referred to as DXTRAN regions because (1) there is no agreed upon terminology for such regions and (2) this term is as terse as possible. MCBEND (Richards et al., 2013, Shuttleworth et al., 2000) has the ability to perform angular biasing (referred to as forced flight) using circular, elliptical, rectangular, or spherical DXTRAN regions. The only prior work implementing a method applicable to arbitrary convex regions is by Tickner (2009) (referred to as forced scattering) in EGSnrc (Kawrakow et al., 2017). EGSnrc is open source (https://github.com/nrc-cnrc/EGSnrc), but this capability was implemented by Tickner in a research version and is not publicly available. His work implemented arbitrarily aligned right parallelepiped (RPP), right circular cylinder (RCC) and spherical DXTRAN regions but

MCNP® and Monte Carlo N-Particle® are registered trademarks owned by Los Alamos National Security, LLC, manager and operator of Los Alamos National Laboratory. Any third party use of such registered marks should be properly attributed to Los Alamos National Security, LLC, including the use of the ® designation as appropriate. Any questions regarding licensing, proper use, and/or proper attribution of Los Alamos National Security, LLC marks should be directed to trademarks@lanl.gov.

* Corresponding author at: Los Alamos National Laboratory, PO Box 1663, Mailstop P225, Los Alamos, NM, 87545, United States.

E-mail address: jkulesza@umich.edu (J.A. Kulesza).

https://doi.org/10.1016/j.anucene.2017.12.045
0306-4549/

was only demonstrated with an axis-aligned RPP (AARPP). The work herein differs from Tickner's because his approach is based on the view of the particle from the DXTRAN region requiring surface calculations on the DXTRAN region and possible front- and back-face corrections. This work follows the same overall geometric approach as MCNP6 based on the view of the DXTRAN region from the particle. Furthermore, this approach is demonstrated for a large variety of DXTRAN region shapes.

First, the method is derived in Section 2 for DXTRAN regions defined using polyhedra with additional consideration given to AARPPs. Both geometries are described because AARPPs can often conform relatively well to oblong regions of interest and are much less computationally expensive to use than arbitrary polyhedra. Section 3 gives a brief description of the method's implementation. Finally, Section 4 defines a series of test problems and Section 5 gives accompanying results using a research version of the MCNP® version 6.2 code system (Goorley et al., 2012).

The forthcoming discussion requires some knowledge of computational geometry algorithms. The three main computational geometry algorithms required for this work and some recommended references are:

1. Determining the closest point on the boundary of a polyhedron to an external point (a degenerate polyhedron) such as with the GJK (Gilbert et al., 1988) or RGJK (Ong and Gilbert, 2001) algorithms,
2. Determining the vertices and edges of a convex hull in a plane, and how to order the vertices/edges into a cyclic set such as with algorithms by Jarvis (1973); Kirkpatrick and Seidel (1983); Barber et al. (1996); Chan (1996), and
3. Determining the first point of intersection on the boundary of a polyhedron and an external directed ray such as with the Cyrus-Beck algorithm (Cyrus and Beck, 1978) or another approach as discussed by Jiménez et al. (2001).

The details of these algorithms are not covered in depth in this paper; however, as the overall method is described the reader is reminded of relevant algorithms.

2. Methodology

2.1. Overview of spherical DXTRAN region processing

Before presenting the new method, it is helpful to provide a brief and simplified overview of a basic MCNP6 DXTRAN process upon which the new method is based. DXTRAN processing initiates when a particle is emitted from the source or has a collision at spatial point $\mathbf{p}$ traveling with initial direction $\mathbf{\Omega}$. First, a polar axis is defined from $\mathbf{p}$ to the center of the spherical DXTRAN region $\mathbf{p}_S$. The maximum polar angle of scattering θ_O, which corresponds to the minimum polar cosine, is calculated based on the radius of the circular projection of the DXTRAN region as viewed from $\mathbf{p}$. The range of polar cosines defines the non-analog probability density function (pdf) $\tilde{p}(\mu = \cos\theta)$ that is then sampled to determine θ, the polar angle of scattering. An azimuthal angle is sampled about the polar axis uniformly in $[0,2\pi)$ to define the new direction of flight $\mathbf{\Omega}_{DX}$ for the particle. The value of the analog pdf $p(\mu = \mathbf{\Omega}\cdot\mathbf{\Omega}_{DX})$ is known based on the originating particle's incoming direction $\mathbf{\Omega}$ and forced outgoing direction $\mathbf{\Omega}_{DX}$. A new particle (the DXTRAN particle) is then created on the DXTRAN surface at $\mathbf{p}_{DX}$, which is where a ray from $\mathbf{p}$ along $\mathbf{\Omega}_{DX}$ first intersects the DXTRAN region. See Fig. 1. The DXTRAN particle's weight, w_{DX}, is then adjusted relative to the originating particles weight w to account for an optional user-defined space-dependent rouletting parameter $\beta(\mathbf{p})$, the non-analog direction change, and the spatial attenuation between $\mathbf{p}$ and $\mathbf{p}_{DX}$ (with optical distance ℓ) as

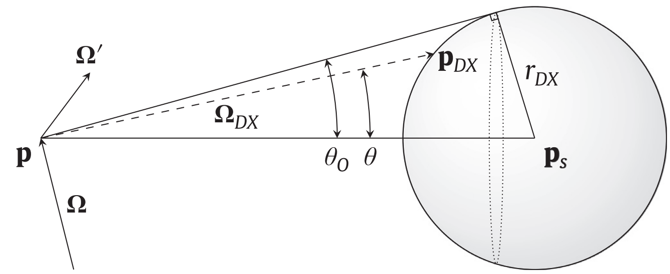

Fig. 1. Spherical DXTRAN region and initiation point **p**.

$$w_{DX} = \frac{w}{\beta(\mathbf{p})}\frac{p(\mu)}{\tilde{p}(\mu)}\exp(-\ell). \tag{1}$$

Once done, the initiating particle continues to be followed in its normal random walk (with outgoing direction $\mathbf{\Omega}'$) but is killed if it attempts to enter the DXTRAN region on its next free flight—this preserves the fairness of the variance reduction game and therefore the tally mean by balancing weight created and destroyed on the DXTRAN surface.

More details regarding DXTRAN processing are available in Section 3 and X-5 Monte Carlo Team (2008). Note that $\beta(\mathbf{p})$ (`DXC` in MCNP6) is included because it can be used to minimize the expensive computations associated with DXTRAN; however, its use is not explored further herein. Also, to avoid complicating this discussion, the DXTRAN regions herein do not have a concept of inner/outer surfaces like the spheres described in X-5 Monte Carlo Team (2008) and various treatments of weight are neglected other than that described in Eq. (1). Finally, this work focuses on MCNP6's DXTRAN process using a particle-centric approach but other approaches are available such as by Tickner (2009) that are DXTRAN region centric, as noted in Section 1.

2.2. Polyhedral DXTRAN region processing

The overall process for polyhedral DXTRAN is similar to spherical DXTRAN: the particle calculates a new direction of flight toward/into the DXTRAN region and projects to the surface of the region with that new direction. However, polyhedral DXTRAN region calculations are more complicated because (1) the projected view of the DXTRAN region is not circular (and is unpredictable because it is an explicit function of the position of the initiating particle relative to the DXTRAN region) and (2) the polar axis may not go through the center of the region. Thus, additional calculations must be made to (1) determine the projected view of the DXTRAN region, (2) determine the new direction of flight, and (3) determine the projection to the surface of the DXTRAN region for the DXTRAN particle.

To begin, assume the initiating particle (either source emission or non-absorptive collision) is at position $\mathbf{p} = (p_x, p_y, p_z)$. DXTRAN can only be initiated external to the DXTRAN region so $\mathbf{p}$ must not be within the DXTRAN region or on its surface. Rather than assuming a circular projection, calculate the 2-D projection of the DXTRAN region as viewed by the initiating particle. Find the 2-D projection by projecting the vertices of the polyhedron into a projection plane as viewed from $\mathbf{p}$ and calculating the 2-D convex hull. The convex hull gives the angular extents that a DXTRAN particle can be projected toward and the interior points of the convex hull are all valid points to project toward. To construct the convex hull, begin with a DXTRAN region with the boundary specified by an arbitrary convex polyhedron defined with vertices. Number the vertices $1, 2, \ldots, N$ in an arbitrary order which then have coordinates