Theoretical Analysis of Track Generation in 3d Method of Characteristics

ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method · Nashville, Tennessee · April 19–23, 2015, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2015)

THEORETICAL ANALYSIS OF TRACK GENERATION IN 3D METHOD OF CHARACTERISTICS

Samuel Shaner, Geoffrey Gunow, Benoit Forget, and Kord Smith
Department of Nuclear Science & Engineering
Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA 02139
shaner@mit.edu; geogunow@mit.edu; bforget@mit.edu; kord@mit.edu

ABSTRACT

Generating the tracks to use in a 3D Method of Characteristics (MOC) simulation is not a trivial task. The method used to generate tracks has significant implications on the memory and compute requirements for a problem and the current track generation methods have shortcomings. In this study, we provide a detailed description and analysis of the current state of the art method to generate tracks for direct 3D MOC, the Modular Ray Tracing (MRT) method. Additionally, a new global method for generating tracks is presented that is generalizable to many geometries, domain decomposition schemes, and quadrature sets. The main difference between the global and modular track generation approaches is that the global approach does not require any knowledge of the underlying geometry discretization and is therefore more flexible in domain decomposing the geometry. Some considerations with memory requirements and general applicability that we and others have found are discussed.

Key Words: Method of Characteristics, ray tracing

1 INTRODUCTION

The Method of Characteristics (MOC) has seen significant interest in the past several decades owing to the computational efficiency and flexibility of the method to model complex geometries. Several production neutronics codes utilize 2D MOC for lattice physics or full core neutronics calculations. Full core analysis using MOC has typically followed a 2D/1D approach [1–4]. In this approach, 2D MOC is used in the radial direction while transport in the axial direction is modeled with a low order diffusion operator that uses transverse leakage to couple the radial planes. Recently, several codes has been written or extended to perform direct MOC calculations in 3D [5–7]. Extending MOC to 3D comes with a wide array of challenges.

One of the areas that has been investigated recently is the development of procedures to create 3D tracks. The modular ray tracing (MRT) method developed by Filippone [8] has seen wide adoption in codes including MPACT [5, 9], DeCART [2], DRAGON [6], CRX [7], PEACH [10], and nTRACER [1] as a way to reduce the memory and compute time required to generate tracks for a large 3D geometry. While simple in theory, MRT requires all domains to be the same size which can lead to load imbalance issues during segmentation if the minimum repeating domain is too large. Additionally, when certain simplifications are used, the z-spacing between 3D tracks can be limited for shallow polar angles [5]. This results in additional memory and compute requirements that could prove burdensome when trying to produce high fidelity solutions to full core PWR problems. Studies performed on 3D MOC often use pin cell domains during the MRT segmentation process when simplified problems such as the 3D C5G7 benchmark are solved [5]. With the small domain sizes, the load imbalance issues are trivial. However, in real PWR problems such as the BEAVRS benchmark [11], the strict pin cell structure is no longer preserved due to the presence of the grid sleeve and inter-assembly water gap. Furthermore, the presence of grid spacers, flow nozzles, and the bottom support plate result in uneven axial zone heights in the geometry. These characteristics complicate MRT and suggest a more streamlined, global approach to generating tracks might be more practical.

In this paper, we describe the conditions required for cyclic ray tracing and present a new procedure for creating global 3D tracks that is independent of the domain decomposition scheme used. This new method, which we call 3D Global Tracking (3DGT), uses information about the 2D track cycles to perform a global fitting to ensure cyclic wrapping of 3D tracks. In order to highlight the benefits of this method, we present the number of tracks required for the MRT, simplified MRT, and 3DGT methods using anticipated parameters for a converged 3D MOC solution of a PWR core.

2 2D TRACKING

The Method of Characteristics is a widely used method for finding the solution of partial differential equations in which a two or three dimensional problem is approximated by solving the transport equation along constant angle tracks that traverse the geometry. While one dimensional, each track represents a particular three dimensional volume and angular subspace. The first step in setting up the MOC problem is creating tracks that span the spatial and angular space of the problem. In Fig. 1, a sparse 2D and 3D track laydown for a homogeneous cube geometry is illustrated.

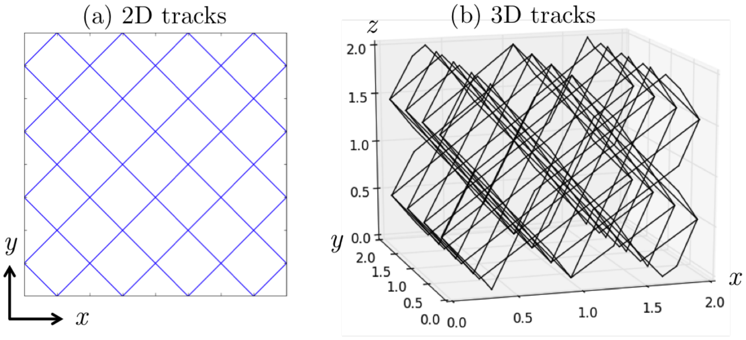

Figure 1. Pictures illustrating 2D (a) and 3D (b) tracks for a homogeneous cube geometry.

When creating the tracks, it is important to consider the boundary conditions as these determine whether the outgoing angular flux needs to be passed as the incoming flux to another track. In this study, we focus on generating tracks that are cyclic and can therefore be used for reflective or vacuum boundary conditions. While analysis of full-core problems typically involves only vacuum boundaries, it is helpful to have the option for reflective boundaries to compare the results of small-core benchmarks, such as the C5G7 benchmark, to other codes.

In our discussion, it is important that we make clear the distinction between tracks and segments (sometimes referred to as intersections). Tracks are defined to span an entire geometry or geometry subdomain and pass through region boundaries. When setting up a problem, tracks are decomposed into segments that span only a single region. Fig. 2 illustrates a fine track and segment laydown for a pin cell geometry. In this study, we focus only on the track generation procedure.

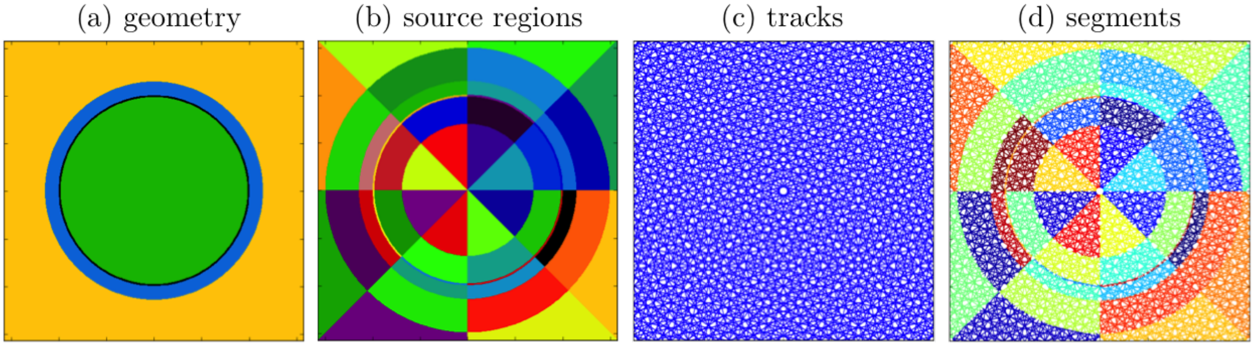

Figure 2. Pictures illustrating the geometry (a), the geometry subdivided into source regions (b), a fine 2D track laydown (c), and the corresponding segment laydown colored by source region (d) for a 2D pin cell geometry containing fuel, gap, clad, and moderator materials.

Before discussing the methodology for 3D track generation, we discuss how global tracks are generated for 2D problems. Tracks for a 2D MOC problem are typically laid down using the cyclic tracking approach as illustrated in Fig. 3. Users input a desired azimuthal track spacing, $\boldsymbol{\delta_\phi}$, and number of azimuthal angles, n_ϕ, in $0 < \phi < 2\pi$. With this approach, azimuthal angles are created to evenly subdivide the angular space such that each azimuthal angle in $0 < \phi < \frac{\pi}{2}$ is paired with a complementary azimuthal angle,

$$\boldsymbol{\phi^i} = \frac{2\pi}{n_\phi}(0.5 + i) \qquad \forall \qquad i = \left[0, \frac{n_\phi}{2}\right) \tag{1}$$

$$\boldsymbol{\phi^{\frac{n_\phi}{2}-i-1}} = \pi - \boldsymbol{\phi^i} \qquad \forall \qquad i = \left[0, \frac{n_\phi}{4}\right), \tag{2}$$

where $\boldsymbol{\phi^{\frac{n_\phi}{2}-i-1}}$ is the complement of angle $\boldsymbol{\phi^i}$. Other valid angular quadrature sets have also been used [7]. By tracking both forward and backward along a track, the full 2π angular space is covered as shown in Fig. 3 for four azimuthal angles. Tracks are laid down such that they intersect with a complementary track at the boundaries.

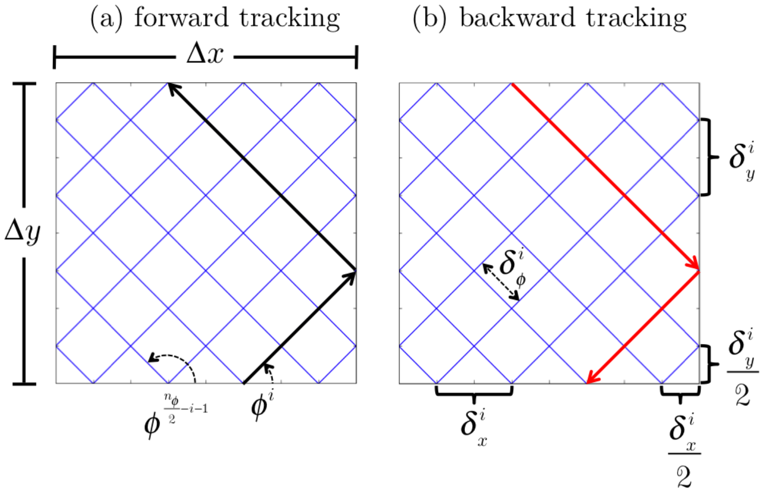

Figure 3. Pictures illustrating forward (a) and backward (b) tracking for 2D MOC track laydown. The geometry width Δx, geometry height Δy, and track angles and spacing (collectively ϕ^i, $\phi^{\frac{n_\phi}{2}-i-1}$, δ^i_ϕ, δ^i_x, and δ^i_y) have been denoted on the pictures.

In order to guarantee cyclic wrapping of 2D tracks, there must be an integer number of tracks on x and y axes for a particular angle, $\boldsymbol{\phi}^{\boldsymbol{i}}$,

$$\delta^i_x = \frac{\Delta x}{n^i_x} \qquad \delta^i_y = \frac{\Delta y}{n^i_y} \tag{3}$$

where n^i_x is the integer number of tracks along the x axis for angle $\boldsymbol{\phi}^{\boldsymbol{i}}$. The same notation applies to the y direction. Using the input value of $\boldsymbol{\delta}_{\boldsymbol{\phi}}$, the integer number of tracks along the axes for a particular angle $\boldsymbol{\phi}^{\boldsymbol{i}}$ are computed,

$$n^i_x = \left\lceil \left(\frac{\Delta x \cdot \sin(\boldsymbol{\phi}^{\boldsymbol{i}})}{\boldsymbol{\delta}_{\boldsymbol{\phi}}} \right) \right\rceil \qquad n^i_y = \left\lceil \left(\frac{\Delta y \cdot \cos(\boldsymbol{\phi}^{\boldsymbol{i}})}{\boldsymbol{\delta}_{\boldsymbol{\phi}}} \right) \right\rceil \tag{4}$$

where the ceiling is taken to ensure at least one track intersects with each axis. The azimuthal angle is then corrected,

$$\phi^i = \tan^{-1}\left(\frac{\delta^i_y}{\delta^i_x}\right) \tag{5}$$

where ϕ_i is used to denote the corrected azimuthal angle. Using the corrected azimuthal angle, the azimuthal track spacing for each angle is also corrected,

$$\delta_\phi^i = \delta_x^i \cdot \sin(\phi^i) \tag{6}$$

where δ_ϕ^i is used to denote the corrected azimuthal track spacing. Using the corrected values, ϕ^i and δ_ϕ^i, the 2D tracks are laid down on the geometry.

3 3D TRACKING

3.1 Requirements for Cyclic Track Laydown in 3D

In generating 3D tracks, we assume that 3D tracks are laid down as z-stacked arrays of tracks that project down onto the 2D track layout. Before discussing the conditions required to generate cyclic tracks in 3D, we need to understand the concept of a reflective track cycle. Fig. 4 decomposes a sample 2D geometry into 2D reflective track cycles with $T_{R,k}^i$ denoting the k^{th} reflective track cycle for azimuthal angle ϕ^i.

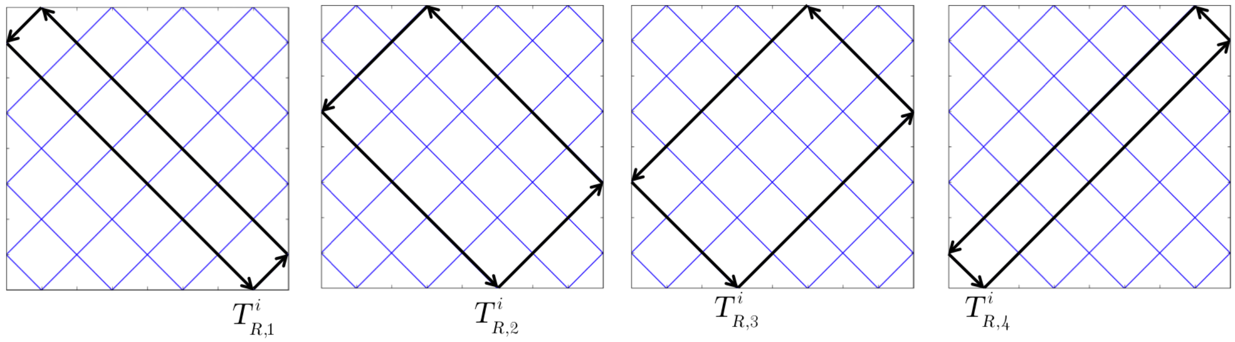

Figure 4. Each plot highlights one of the 2D track cycles contained in a sample geometry. The reflective track cycles are labeled with $T_{R,k}^i$ denoting the k^{th} track cycle for azimuthal angle ϕ^i.

To generate 3D tracks, users input a desired polar track spacing, $\boldsymbol{\delta_\theta}$, and number of polar angles, n_θ, in $0 < \theta < \pi$ in addition to the parameters required to generate 2D tracks. With this approach, polar angles are created to evenly subdivide the angular space such that each polar angle in $0 < \theta < \frac{\pi}{2}$ is paired with a complementary polar angle,

$$\boldsymbol{\theta^{i,j}} = \frac{\pi}{n_\theta} \cdot (0.5 + j) \qquad \forall \qquad j = \left[0, \frac{n_\theta}{2}\right) \tag{7}$$

$$\boldsymbol{\theta^{i,n_\theta - j - 1}} = \pi - \boldsymbol{\theta^{i,j}} \qquad \forall \qquad j = \left[0, \frac{n_\theta}{2}\right), \tag{8}$$

where $\boldsymbol{\theta^{i,n_\theta - j - 1}}$ is the complement of angle $\boldsymbol{\theta^{i,j}}$. Note that the polar angles are initially defined to be independent of azimuthal angle index, i. Later, we use the azimuthal angle to correct the polar angle to ensure the tracks are cyclic. Following the same notation used to describe the azimuthal angles, the corrected polar angles will be denoted by $\theta^{i,j}$. By tracking both forward and backward along a track, the full 4π angular space can be covered as shown in Fig. 1 for four azimuthal and two polar angles.

Tracks are laid down such that they intersect with a complementary track at the boundaries. Selecting an arbitrary cycle, $T^i_{R,k}$, we follow a set of 3D tracks as they complete one 2D cycle. Fig. 5 highlights a particular 2D track cycle and a set of 3D tracks projected along that cycle.

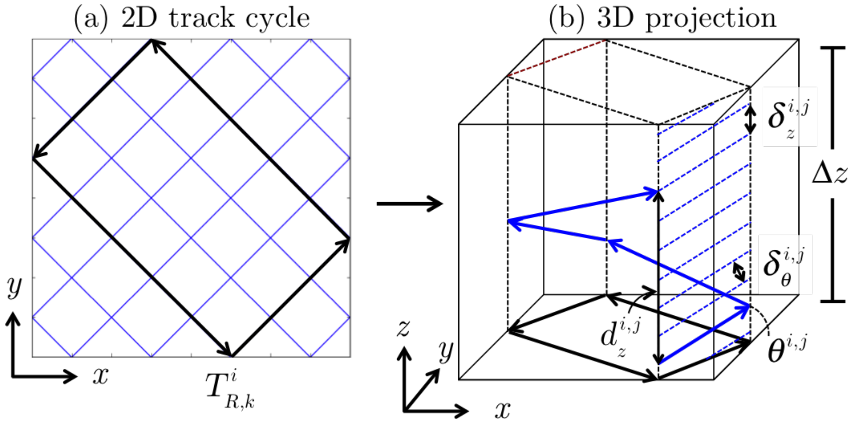

Figure 5. Illustration of an arbitrary 2D track cycle (a), $T^i_{R,k}$, and a set of 3D tracks projected along the 2D track cycle (b).

To guarantee cyclic track wrapping of the 3D tracks, two conditions must be met:

1. For each azimuthal angle, ϕ^i, polar angle, $\theta^{i,j}$, and 2D track cycle, $T^i_{R,k}$, the distance between the beginning and end of a 3D track projection along a 2D track cycle must be an integer number of track spacings.

2. For each azimuthal angle, ϕ^i, and polar angle, $\theta^{i,j}$, there must be an integer number of track spacings along the z axis over the depth of geometry, Δz.

The first condition guarantees that a 3D track cycle wraps back onto another 3D track when the 2D reflective cycle is completed. The second condition guarantees that a 3D track cycle that contains a reflection off a top or bottom surface still reflects into an existing 3D track when the 2D cycle is completed. In the next two sections, we show how both the 3DGT and MRT methods comply with these conditions and what additional assumptions they make.

3.2 The 3D Global Tracking Method

The 3DGT method uses the conditions stated in the previous section for 3D cyclic ray tracing to generate tracks for a rectangular geometry independent of any domain decomposition scheme.

After the 2D track information has been generated, the 3D track information can be computed using the requirements for 3D cyclic global ray tracing. First, we note that all 2D track cycles for a particular azimuthal angle index, i, have the same cycle length that can be computed,

$$L_R^i = \mathrm{lcm}\left(2 \cdot n_x^i, \frac{2 \cdot \Delta y}{\tan(\phi^i) \cdot \delta_x^i}\right) \cdot \frac{\delta_x^i}{\cos(\phi^i)}, \tag{9}$$

where lcm is the least common multiple. There are several ways in which the 3D track information can be found given a user specified polar spacing and number of polar angles. The method we use below is well suited towards minimizing the correction to the desired polar angles while allowing the polar track spacing to change significantly. Using the reflective cycle length for each azimuthal angle, the integer number of track spacings between the beginning and end of a set of 3D tracks after one complete 2D cycle can be computed,

$$n_l^i = \left\lceil L_R^i \cdot \cot(\boldsymbol{\theta^{i,j}}) \cdot \frac{\sin(\boldsymbol{\theta^{i,j}})}{\boldsymbol{\delta_\theta}} \right\rceil, \tag{10}$$

where $L_R^i \cdot \cot(\boldsymbol{\theta^{i,j}})$ is the z distance between the start and end of the set of 3D tracks, $\frac{\sin(\boldsymbol{\theta^{i,j}})}{\boldsymbol{\delta_\theta}}$ is the z distance between tracks, and the subscript l is used to signify that n_l^i is measured along the direction of a track cycle. Next, the number of track spacings along the z axis needs to be set to an integer number. The number of tracks on the z axis can be found by considering the relation between the number of track spacings in the z direction and the spacing along the length of the 2D track.

$$\tan(\theta^{i,j}) = \frac{\delta_L^i}{\delta_z^{i,j}} = \frac{L_R^i}{n_l^i} \cdot \frac{n_z^{i,j}}{\Delta z} \tag{11}$$

Rearranging and inserting our approximation for the polar angle, $\boldsymbol{\theta^{i,j}}$, this equation gives us an expression for the number of tracks on the z axis,

$$n_z^{i,j} = \left\lceil \frac{\Delta z \cdot n_l^i \cdot \tan(\boldsymbol{\theta^{i,j}})}{L_R^i} \right\rceil, \tag{12}$$

where we take the ceiling to ensure at least one track crossing on the z axis. The z-spacing between 3D tracks is shown in Eq. 13.

$$\delta_z^{i,j} = \frac{\Delta z}{n_z^{i,j}} \tag{13}$$

Using the 2D cycle length and number of track crossings on the z axis and along the length of the 2D cycle, the polar angle can be corrected using Eq. 14.

$$\theta^{i,j} = \tan^{-1}\left(\frac{L_R^i}{n_l^i \cdot \delta_z^{i,j}}\right) \tag{14}$$

Similarly, the polar track spacing is corrected using Eq. 15.

$$\delta_\theta^{i,j} = \delta_z^{i,j} \cdot \sin(\theta^{i,j}) \tag{15}$$

In summary, the algorithm for generating tracks with the 3DGT method is described by Alg. 1.

Algorithm 1 3D track generation using the 3D Global Tracking Method

```
User specifies $n_\phi$, $\boldsymbol{\delta_\phi}$, $n_\theta$, and $\boldsymbol{\delta_\theta}$.
for all $i \in I$ do {Loop over azimuthal angles}
  Compute the # of tracks in x and y for $\boldsymbol{\phi^i}$, $n_x^i$ and $n_y^i$, and distance between tracks in x and y, $\delta_x^i$ and $\delta_y^i$ (Eqs. 3 and 4).
  Correct the azimuthal angle and azimuthal track spacing, $\phi^i$ and $\delta_\phi^i$ (Eqs. 5 and 6).
  Compute the length of the 2D reflective track cycles, $L_R^i$ (Eqs. 9).
  for all $j \in J$ do {Loop over polar angles}
    Compute the # of track spacings after one complete 2D cycle, $n_l^{i,j}$ (Eq. 10).
    Compute the # of tracks on the z axis, $n_z^{i,j}$ (Eq. 12).
    Compute the z distance between 3D tracks, $\delta_z^{i,j}$ (Eq. 13).
    Correct the polar angle, $\theta^{i,j}$ (Eq. 14).
    Correct the polar track spacing, $\delta_\theta^{i,j}$ (Eq. 15).
  end for
end for
```

3.3 The Modular Ray Tracing Method

Modular ray tracing relies on the principle that a geometry can be uniformly decomposed into a series of rectangular cuboids of equal size. Typically, the decomposition procedure is performed such that many of the cuboids have the same underlying geometry and therefore contain the same segment structure. This allows an identical set of tracks to be laid down on each domain and segmentation only performed on each unique domain type. Therefore, we only present the track generation procedure for a single domain.

The MRT method uses a procedure similar to that used in the 3DGT method for track generation. For a domain, the number of tracks and spacing between tracks in x and y are described in Eq. 16 and Eq. 17, respectively,

$$n_x^i = \left\lceil \left(\frac{\Delta x \cdot \sin(\boldsymbol{\phi^i})}{D_x \cdot \boldsymbol{\delta_\phi}} \right) \right\rceil \qquad n_y^i = \left\lceil \left(\frac{\Delta y \cdot \cos(\boldsymbol{\phi^i})}{D_y \cdot \boldsymbol{\delta_\phi}} \right) \right\rceil \tag{16}$$

$$\delta_x^i = \frac{\Delta x}{D_x \cdot n_x^i} \qquad \delta_y^i = \frac{\Delta y}{D_y \cdot n_y^i}, \tag{17}$$

where D_x and D_y are the integer number of domains in the x and y directions, respectively. This guarantees that an integer number of tracks lie along the x and y boundaries of each domain cell and that tracks on one surface line up with adjoining tracks in the neighbor domain cell.

When generating tracks using the MRT method, it is important to remember that a track crossing a domain interface needs to connect with another track on the neighboring domain. As we follow a series of tracks across the geometry, we notice that the series of tracks is periodic. A periodic track cycle is defined to be the series of domain tracks that repeats when a global track traverses a geometry. Fig. 6 shows the periodic track cycles for one azimuthal angle in our sample geometry when it is split into four domains.

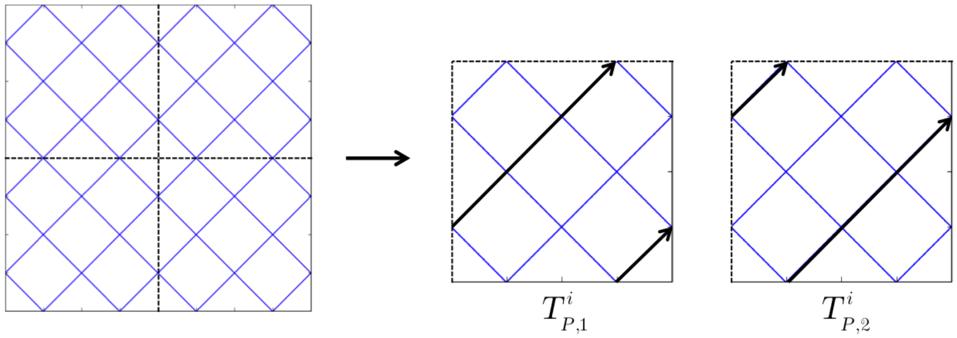

Figure 6. Each plot highlights one of the 2D periodic track cycles contained in a sample geometry split into four domains of equal size. The track cycles are labeled with $T^i_{P,k}$ denoting the k^{th} periodic track cycle for azimuthal angle ϕ^i.

We can define the length of the periodic track cycles for each azimuthal angle to be L_P^i. For the MRT method, the periodic 2D track cycles for a particular azimuthal angle, i, have the same cycle length computed in Eq. 18.

$$L_P^i = \text{lcm}\left(n_x^i, \frac{\Delta y}{D_y \cdot \tan(\phi^i) \cdot \delta_x^i} \right) \cdot \frac{\delta_x^i}{\cos(\phi^i)} \tag{18}$$

The procedure to find the corrected polar track spacing and corrected polar angles is identical to the procedure used in the 3DGT method, except that L_R^i is replaced with L_P^i and Δz is replaced with $\frac{\Delta z}{D_z}$.

The procedure for identifying the periodic track cycles and finding the start and end points for all tracks using the MRT and 3DGT methods can seem a bit tedious, so others have simplified the MRT method by noticing that the lengths of all 2D tracks are an integer multiple of the shortest 2D track [5]. For example, the length of the first three 2D tracks, t_1^i, t_2^i, and t_3^i, for azimuthal angle ϕ^i are shown in Eqs. 19, 20, and 21.

$$l_1^i = \sqrt{\left[\frac{\delta_x^i}{2}\right]^2 + \left[\frac{\delta_y^i}{2}\right]^2} = \frac{1}{2}\sqrt{(\delta_x^i)^2 + (\delta_y^i)^2} \tag{19}$$

$$l_2^i = \sqrt{\left[\frac{3\delta_x^i}{2}\right]^2 + \left[\frac{3\delta_y^i}{2}\right]^2} = \frac{3}{2}\sqrt{(\delta_x^i)^2 + (\delta_y^i)^2} = 3l_1^i \tag{20}$$

$$l_3^i = \sqrt{\left[\frac{5\delta_x^i}{2}\right]^2 + \left[\frac{5\delta_y^i}{2}\right]^2} = \frac{5}{2}\sqrt{(\delta_x^i)^2 + (\delta_y^i)^2} = 5l_1^i \tag{21}$$

Since the length of all 2D tracks is an integer number of lengths of the first track, l_1^i, any valid track laydown for the first track is also valid for all other tracks. Therefore, we can set the cycle length for an azimuthal angle, L_P^i, to the length of the shortest track, l_1^i. The algorithm for generating tracks for the MRT and simplified MRT (s-MRT) method can be described by Alg. 2.

It is important to note that the track generation procedure described in Alg. 2 favors correcting the polar track spacing over correcting the polar angle. Alternative algorithms could easily be designed to favor correcting the polar angle over the polar track spacing, but it is expected that correcting the polar track spacing is a more conservative procedure and this is in line with previous work on MRT [5]. When the periodic cycle length is small relative to the desired polar track spacing, the correction to the polar track spacing can be very large. This has significant implications for the s-MRT method where the periodic cycle length is always on the order of the azimuthal track spacing. For instance, if we consider the case of a shallow polar angle ($\boldsymbol{\theta}$ near $\frac{\pi}{2}$) where $\boldsymbol{\phi} = \frac{\pi}{4}$, $\boldsymbol{\theta} = 0.9 \cdot \frac{\pi}{2}$, $\boldsymbol{\delta_\phi} = 0.05$, and $\boldsymbol{\delta_\theta} = 0.25$ for an infinitely tall domain, the s-MRT method produces a corrected polar track spacing of $\delta_\theta^{i,j} \approx 0.0079$. This increases the number of tracks for this direction by a factor of $\sim$32 over the ideal case and in Sec. 4 we discuss the memory implications of this characteristic in more detail.

Additionally, others have claimed that the MRT method must store all tracks in the unit sphere while the s-MRT method only requires tracks in half of the unit sphere [5]. Our algorithm for generating tracks using the MRT and s-MRT methods (Alg. 2) satisfies the criteria for cyclic tracking and in practice we have been able to produce cyclic track laydowns for the MRT method that require storage of tracks over only half the unit sphere. It is suspected that the previous work did not consider the track cycles as a unit and instead generated the z-stacked arrays of 3D tracks independent of other tracks in the cycle.

Algorithm 2 3D track generation using the Modular Ray Tracing Method

```
User specifies $n_\phi$, $\boldsymbol{\delta_\phi}$, $n_\theta$, $\boldsymbol{\delta_\theta}$, $D_x$, $D_y$, and $D_z$.
for all $i \in I$ do {Loop over azimuthal angles}
  Compute the # of tracks in x and y for $\boldsymbol{\phi^i}$, $n_x^i$ and $n_y^i$, and distance between tracks in x and y, $\delta_x^i$ and $\delta_y^i$ (Eq. 16 and 17).
  Correct the azimuthal angle and azimuthal track spacing, $\phi^i$ and $\delta_\phi^i$ (Eqs. 3 and 4).
  if MRT then
    Compute the length of the 2D periodic track cycles, $L_P^i$ (Eq. 18).
  else
    Compute the length of the shortest track, $l_1^i$ (Eq. 19).
  end if
  for all $j \in J$ do {Loop over polar angles}
    Compute the # of track spacings after one complete 2D cycle, $n_l^{i,j}$.
      $$n_l^i = \left\lceil L_P^i \cdot \cot(\boldsymbol{\theta^{i,j}}) \cdot \frac{\sin(\boldsymbol{\theta^{i,j}})}{\boldsymbol{\delta_\theta}} \right\rceil$$
    Compute the # of tracks on the z axis, $n_z^{i,j}$.
      $$n_z^{i,j} = \left\lceil \frac{\Delta z \cdot n_l^i \cdot \tan(\boldsymbol{\theta^{i,j}})}{D_z \cdot L_P^i} \right\rceil$$
    Compute the z-distance between 3D tracks, $\delta_z^{i,j}$.
      $$\delta_z^{i,j} = \frac{\Delta z}{D_z \cdot n_z^{i,j}}$$
    Correct the polar angle, $\theta^{i,j}$.
      $$\theta^{i,j} = \tan^{-1}\left(\frac{L_P^i}{n_l^i \cdot \delta_z^{i,j}}\right)$$
    Correct the polar track spacing, $\delta_\theta^{i,j}$ (Eq. 15).
  end for
end for
```

4 IMPLICATIONS FOR 3D MOC

As a rough approximation, the required parameters to converge a full-core problem might be similar to those presented in Table I. With the 3DGT method, these parameters lead to the generation of 5.66×10^9 tracks when no memory decomposition scheme is used. With this many tracks, the storage of the two 100-group angular flux vectors for each track with single precision would require 4.53 TB. This is much larger than the memory limit of current machines. For instance, the on-node memory of BlueGene/Q is constrained to 16 GB/node [12].

For any MOC implementation, iterations are performed in which the Boltzmann neutron transport equation is solved for every characteristic track. Therefore, the computational time

Table I. Anticipated parameters for a converged 3D MOC solution of a PWR core.

Parameter	Dimension
Height (z)	455.444 cm
Width (x and y)	365.56188 cm*
Radial ray spacing	0.05 cm
Axial ray spacing	0.25 cm
Number of azimuthal angles	64
Number of polar angles	10

*Width assumed to be 17 assemblies in each direction.

associated with solving any problem directly scales with the number of tracks [5]. In addition, the on-node memory constraints are often dominated by the storage of boundary angular fluxes for each track [13]. The number of tracks generated using the s-MRT, MRT, and 3DGT methods using the parameters in Table I is presented in Table II.

Table II. Number of tracks generated using the three track generation methods applied to the parameters in Table I.

Method	Number of tracks generated
s-MRT	4.74×10^{10}
MRT	5.70×10^{9}
3DGT	5.66×10^{9}

Previous work has often used only one track spacing for both the desired azimuthal and polar track spacing [5, 9]. However, due to the success of 2D/1D methods [3], we anticipate that the required axial spacing will be far more coarse than the radial spacing. Fig. 7 shows the number of tracks versus the ratio of desired polar to azimuthal track spacing for the parameters in Table I. Both the MRT and 3DGT methods do not require large corrections on the polar track spacing due to the relatively large length of the cycle compared with the azimuthal spacing. With equal azimuthal and polar track spacing, the s-MRT method only requires $\sim$2.0 times more tracks than the 3DGT and MRT methods while for the expected parameters, the s-MRT method requires $\sim$8.4 times more tracks. This illustrates the inflexibility of the s-MRT method when applied to common reactor problems in which the radial complexity is much greater than the axial complexity and suggests that the s-MRT method should be avoided.

Incorporating a domain decomposition scheme will only increase the memory requirements further. For the 3D MOC problem presented in Table I, we have assessed the memory requirements for two domain decomposition schemes with a variety of input parameters and presented the results in Table III. The increase in storage requirements for the s-MRT method is due almost entirely to the increase in the $\delta_\theta/\delta_\phi$ ratio and not to the domain decomposition scheme. Reducing the number of polar angles did not have a large effect on the ratio of storage required by the s-MRT method to the storage required by the MRT and 3DGT methods either. At this point, it is unclear what the track spacing and number of azimuthal and polar angles will be to converge a full-core problem. The storage requirements for many cases in Table III are close to or above the on-node storage of

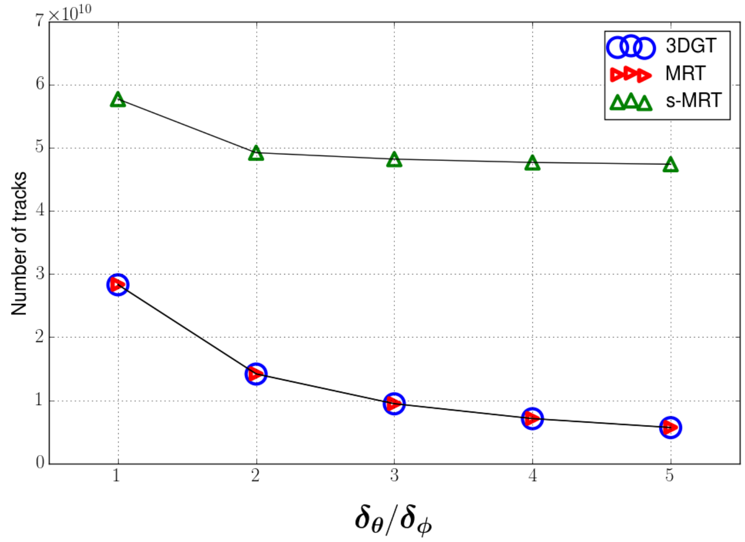

Figure 7. Plot of the number of tracks versus the track polar spacing, δ_θ, divided by the track azimuthal spacing, δ_ϕ, for the specifications described in I. In order to change the ratio of desired polar to azimuthal spacing, the polar track spacing was modified while the desired azimuthal spacing was held constant at 0.05 cm.

BlueGene/Q and the storage for the scalar fluxes and material cross sections has not even been taken into consideration. New machines will likely have different characteristics and it will be important to have a method this both minimizes the memory requirements and is flexible to different domain sizes.

Another important aspect to consider is the generality of the track generation scheme. While benchmark geometries can be quite simple, real geometries are often far more complex and can be composed of many different repeating subdomains of different sizes. The implementation of the MRT method must be able to identify these subdomains and have a robust track generation scheme or rely on the user to identify the subdomains in the input. The 3DGT method is completely general and works for arbitrary geometries, whether they have repeating subdomains (e.g. C5G7) or complex structures (e.g. pebble bed reactors). This characteristic makes the 3DGT method well suited as a general track generation scheme to implement in a 3D MOC code.

5 CONCLUSIONS

In this paper, a detailed description of track generation for 3D MOC was provided. A new track generation method for 3D MOC called 3D Global Tracking that is applicable for arbitrary geometries while having slightly lower memory requirements than the current state-of-the-art MRT method was also presented. A brief analysis of the number of tracks required for an anticipated track laydown for a full-core PWR problem was conducted. This revealed the inherent disadvantage of the s-MRT method in that it required ∼8.4 times more tracks than the MRT and 3DGT methods. As we move towards solving larger, more complex problems, the track generation method will become

Table III. Number of tracks and memory requirements for full core PWR problem using quarter assembly geometry, assembly-wise domain decomposition, 64 azimuthal angles, and 0.05 cm azimuthal ray spacing.

Method	Domains (x,y,z)	n_θ	δ_θ (cm)	$\frac{\text{tracks}}{\text{domain}}$	$\frac{\text{memory}}{\text{domain}}$ (GB)
3DGT	(17,17,21)	10	0.25	8.65×10^6	13.8
MRT				8.71×10^6	13.9
s-MRT				7.00×10^7	112.1
3DGT	(17,17,21)	10	0.05	4.32×10^7	69.2
MRT				4.33×10^7	69.3
s-MRT				8.64×10^7	138.2
3DGT	(34,34,42)	10	0.25	2.16×10^6	3.5
MRT				2.18×10^6	3.5
s-MRT				1.76×10^7	28.2
3DGT	(34,34,42)	10	0.05	1.08×10^7	17.3
MRT				1.09×10^7	17.4
s-MRT				2.17×10^7	34.7
3DGT	(17,17,21)	6	0.25	8.65×10^6	8.4
MRT				8.71×10^6	8.4
s-MRT				7.00×10^7	59.4
3DGT	(17,17,21)	6	0.05	4.32×10^7	41.8
MRT				4.33×10^7	41.9
s-MRT				8.64×10^7	75.7
3DGT	(34,34,42)	6	0.25	2.16×10^6	2.1
MRT				2.18×10^6	2.1
s-MRT				1.76×10^7	14.9
3DGT	(34,34,42)	6	0.05	1.08×10^7	10.5
MRT				1.09×10^7	10.5
s-MRT				2.17×10^7	19.0

more relevant due to the increased memory and compute requirements implicit to the track laydown. Additionally, the current state-of-the-art method, Modular Ray Tracing, will face new challenges in track generation for complex geometries. As we tackle these larger problems, additional challenges and insights to the track generation procedure will likely arise.

6 ACKNOWLEDGMENTS

The first and second author are recipients of the DOE Office of Nuclear Energy's Nuclear Energy University Programs Fellowship. This research was supported in part by the Center for Exascale Simulation of Advanced Reactors (CESAR), a co-design center under the U.S. Department of Energy Contract No. DE-AC02-06CH11357.

7 REFERENCES

[1] Y. S. Jung, et al., “Practical numerical reactor employing direct whole core neutron transport and subchannel thermal/hydraulic solvers,” *Annals of Nuclear Energy*, **62**, pp. 357.

[2] J. Y. Cho, et al., “Cell Based CMFD Formulation for Acceleration of Whole-core Method of Characteristics Calculations,” *Journal of the Korean Nuclear Society*, **34**, pp. 250.

[3] B. W. Kelley and E. W. Larsen, “2D/1D approximations to the 3D neutron transport equation. I: Theory,” *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, ID, USA, May.

[4] N. Z. Cho, G. S. Lee, and C. J. Park, “A Fusion Technique of 2-D/1-D Methods for Three-Dimensional Whole-Core Transport Calculations,” *Korean Nuclear Society Spring Meeting*, Kwangju, Korea, May.

[5] B. Kochunas, “A Hybrid Parallel Algorithm for the 3-D Method of Characteristics Solution of the Boltzmann Transport Equation on High Performance Computing Clusters,” Ph.D. Thesis, University of Michigan, Department of Nuclear Engineering and Radiological Sciences (2013).

[6] M. Dahmani, et al., “Development and Parallelization of the Three-Dimensional Characteristics Solver MCI of DRAGON,” *ANS Reactor Physics Topical Meeting (PHYSOR)*, Seoul, Korea.

[7] Z. Liu, et al., “A new three-dimensional method of characteristics for neutron transport calculation,” *Annals of Nuclear Energy*, **38**, pp. 447.

[8] W. L. Filippone, S. Woolf, and R. J. Lavigne, “Particle Transport Calculations with the Method of Streaming Rays,” *Nuclear Science and Engineering*, **77**, pp. 119.

[9] Z. Liu, et al., “The method of modular characteristic direction probabilities in MPACT,” *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, ID, USA, May.

[10] C. Tang and S. Zhang, “Development and verification of an MOC code employing assembly modular ray tracing and efficient acceleration techniques,” *Annals of Nuclear Energy*, **36**, pp. 1013.

[11] N. Horelik, et al., “Benchmark for Evalulation and Validation of Reaction Simulations (BEAVRS),” *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, Sun Valley, ID, USA, 2013.

[12] “Mira/Cetus/Vesta User Guide,” Argonne Leadership Computing Facility, 2011, http://www.alcf.anl.gov/user-guides/mira-cetus-vesta.

[13] G. Gunow, et al., “SimpleMOC - A Performance Abstraction for 3D Method of Characteristics Analysis,” *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, Nashville, TN, USA, 2015.