Nuclear Engineering and Technology 49 (2017) 1189–1198

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

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

Original Article

Particle tracking acceleration via signed distance fields in direct-accelerated geometry Monte Carlo

Patrick C. Shriwise*, Andrew Davis, Lucas J. Jacobson, Paul P.H. Wilson

CNERG Research Group, University of Wisconsin - Madison, Madison, WI, USA

ARTICLE INFO

Article history:
Received 3 June 2017
Accepted 7 August 2017
Available online 26 August 2017

Keywords:
Monte Carlo
Radiation Transport
CAD
DAGMC

ABSTRACT

Computer-aided design (CAD)-based Monte Carlo radiation transport is of value to the nuclear engineering community for its ability to conduct transport on high-fidelity models of nuclear systems, but it is more computationally expensive than native geometry representations. This work describes the adaptation of a rendering data structure, the signed distance field, as a geometric query tool for accelerating CAD-based transport in the direct-accelerated geometry Monte Carlo toolkit. Demonstrations of its effectiveness are shown for several problems. The beginnings of a predictive model for the data structure's utilization based on various problem parameters is also introduced.

1. Introduction

The direct-accelerated geometry Monte Carlo (DAGMC) [1] toolkit provides the capability for robust radiation transport on CAD geometries. This allows nuclear analysis to be performed on the same models that are used in other engineering domains. The toolkit also has the capability to convert native Monte Carlo models to computer-aided design (CAD) for further development, modification, and analysis.

DAGMC tracks particles on discretized representations of analytic CAD surfaces as triangle meshes, such that all points on any given triangle are within a specified tolerance of the analytic representation. This mesh representation of the model is then stored in the mesh oriented database (MOAB) [2] in which geometric queries such as next surface intersection, point containment, and nearest to boundary are performed.

It has been shown that DAGMC can achieve robust transport equal to that of the native codes with which it communicates [3], but that it currently takes much longer [4]. This additional time occurs despite the acceleration techniques that DAGMC uses to avoid searching over the large number of triangles ($\sim 10^6$–10^7). Those acceleration techniques, implemented within MOAB, will be briefly discussed to motivate this work, which aims to further accelerate geometry queries within DAGMC.

To begin, this article gives a brief overview of the employed acceleration techniques in DAGMC for particle tracking. Next, the signed distance field data structure intended for rendering dynamic surfaces is introduced. The remaining content describes this data structure's adaptation for use in accelerated particle tracking for CAD-based Monte Carlo simulations. The application of this data structure and associated tracking methods are then demonstrated as a new method for particle tracking acceleration in DAGMC. Included demonstrations of this method are shown to greatly reduce run times in a few test problems for commonly encountered conditions during transport. Next, some studies involving the data structure's effectiveness for various problem parameters are investigated and discussed. Finally, conclusions, current limitations, and future extensions of this work are discussed.

2. Background

2.1. Acceleration techniques in DAGMC

Particle tracking in DAGMC relies on the capability to robustly perform geometric queries on the triangle surface meshes which represent the problem geometry. The most common form of geometric query in particle tracking is a next surface crossing query. This is called to determine if a particle within the current cell will reach its next event location or cross a surface of the cell, potentially entering a new medium with different physical characteristics, based on the particle's current position and trajectory. Geometry kernels in native Monte Carlo codes perform analytic calculations for the nearest intersection along the particle's trajectory from its current location. In DAGMC, a similar operation is performed, but on the triangle surfaces that compose the current cell (or volume).

* Corresponding author.
E-mail address: shriwise@wisc.edu (P.C. Shriwise).

http://dx.doi.org/10.1016/j.net.2017.08.008

The determination of the nearest triangle intersection with a trajectory from a starting position is a well-researched problem in the field of ray tracing. As such, a ray tracing data structure, the bounding volume hierarchy (BVH) [5], is applied within MOAB to accelerate these queries. More specifically, MOAB utilizes an oriented bounding box (OBB) BVH [6,7] to bound sets of triangles so that they may be rapidly excluded from the query in large groups via intersection checks with these OBBs.

This process begins by bounding all triangles with a single OBB. The triangles are then split into two sets based on which side of the splitting plane they are on. The splitting plane is chosen as the current OBBs median plane which divides the number of triangles into equal parts. OBBs are then generated for these subsets of triangles. Concurrently, the new OBBs are linked to the original OBB as children. This process recurs until OBBs are bounding small sets of triangles. A hierarchical structure is created along the way which can then be traversed to rapidly isolate sets of triangles in space. A two-dimensional (2D) visualization of this process using curve segments rather than triangles is provided in Fig. 1. Queries are satisfied using data structure by performing ray-box intersection checks with the root OBB. If an intersection with the root OBB is found, then the children of that OBB will also be checked for an intersection. If the OBB has no children, it is considered a leaf node and the triangles it bounds are then checked for an intersection with the ray. During traversal, each time an OBB is missed by the ray, all triangles bounded by that box are ignored. It is in this way that the search for an intersection is rapidly limited to a small number of triangles with a high chance of intersecting with the ray. This results in an algorithm with $O(log(N))$ complexity where N is the number of entities (triangles) in the data structure. MOAB's OBB tree can be used to satisfy point containment and closest intersection queries as well. Many variations of this data structure exist, and research is in progress to further accelerate particle tracking with these methods [4].

2.2. Signed distance fields and implicit surfaces

Signed distance fields are commonly derived from implicit surface functions, and variations on these functions are known as level-set functions. Both offer a rich and versatile representation of closed manifolds that can be used for modeling, simulation, and rendering. The constructive solid geometry (CSG) representations seen in native Monte Carlo codes are usually formed from Boolean combinations of predefined implicit surfaces at their core. While these predefined surfaces do not give the freedom of model creation and manipulation found in many CAD systems, important geometric information required for visualization and simulation can be readily recovered from these implicit surfaces which may be of value in CAD-based radiation transport simulations.

Implicit surface functions are multivariate functions defined over the R^3 domain as:

$$\Omega\left(R^3\right) \rightarrow R \tag{1}$$

where an isocontour of value, v, of the implicit surface can be described as

$$\Omega(\vec{x}) - v = 0 \tag{2}$$

for all points $\vec{x}$ satisfying that equation. For simplicity, the surface isocontour value is typically defined as 0.

By recognizing that the magnitude of $\Omega(\vec{x})$ is in fact a minimum interface distance function, one can construct a signed distance function, $SDV(\vec{x})$, using the isocontour representation and the magnitude of the function as seen in Eq. 3 [8].

$$SDV(\vec{x}) = |\Omega(\vec{x})| \tag{3}$$

Signed distance function generation from implicit surfaces is a particularly valuable property of implicit surfaces. A signed distance function, $SDV(\vec{x})$, meets the following requirements for any point $\vec{x}$:

- $SDV(\vec{x}) = 0$ for all $\vec{x}$ on the surface boundary,
- $SDV(\vec{x}) < 0$ for all $\vec{x}$ inside the surface boundary, and
- $SDV(\vec{x}) > 0$ for all $\vec{x}$ outside the surface boundary.

Implicit surfaces and level-set methods are easily extended to represent dynamic geometries by including a time dependence in the function, making them powerful tools for populating signed distance fields in simulation and rendering of fluids, smoke, fire, etc. In these applications the data structure is populated with signed distance values for a given time in the rendering. The signed distance field can then be used to determine point containment queries and trace rays at any time via a method in which the ray length is repeatedly clipped using signed distance values to approach a surface in a process called ray marching [9]. This work establishes the potential use of signed distance fields in static CAD-based Monte Carlo radiation transport, with possible future extension to dynamic geometries.

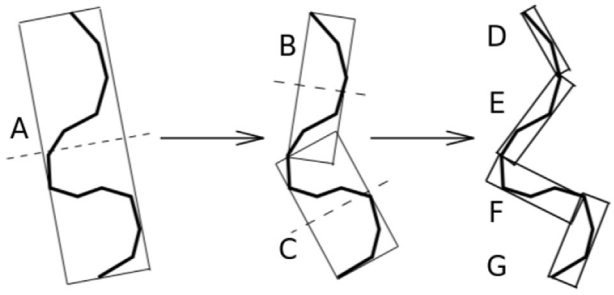

Fig. 1. Two-dimensional example of a bounding volume hierarchy using oriented bounding boxes on a discretized curve.

3. Signed distance field implementation in DAGMC

As an initial implementation, one signed distance field is generated for each volume in DAGMC with extents matching the axis-aligned bounding box of the volume. The signed distance field is represented as a uniform structured mesh with a signed distance value at each vertex in the mesh as indicated in Fig. 2. (Altering the sign convention when populating the data structure rather than incurring the additional computational cost of altering the sign of values for each operation is preferable.) It is mentioned above that signed distance fields are typically generated using an implicit, analytic representation, but a suitable data structure for populating the structured mesh with signed distance values is already in place in the form of DAGMC's bounding volume hierarchy. It is a more straightforward process to simply use DAGMC's current closest to location algorithm to generate signed distance values than to create an implicit surface approximation of the triangle mesh. Though the latter may be faster, performance improvements didn't seem to be impeded greatly by using this method to populate the data structure. This method also maintains a consistency between the intersections found by the ray tracing kernel and the signed distance field values.

DAGMC's closest to boundary algorithm returns, among other pieces of information, the nearest intersection location and the

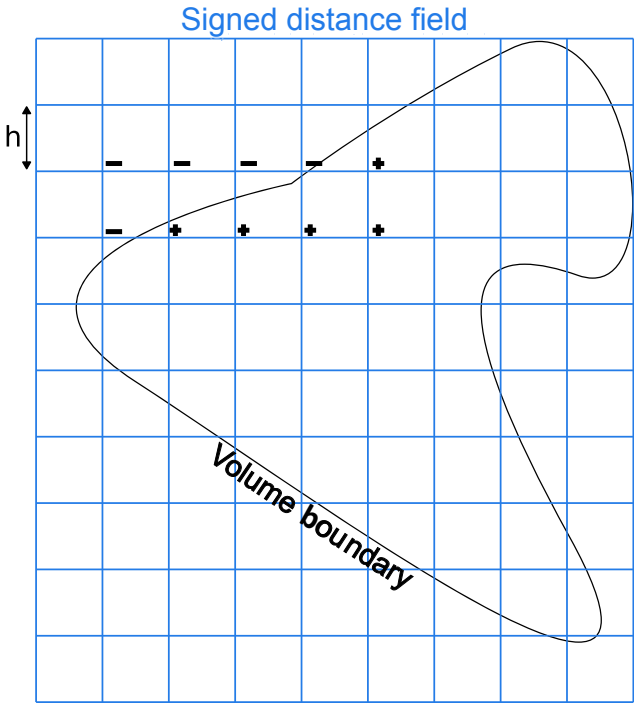

Fig. 2. 2D visualization of the signed distance field with sign conventions reversed for use in radiation transport.

triangle on which this intersection exists. For each point in the signed distance field mesh, this algorithm is used to determine the magnitude of the distance value. To determine the sign of the distance value, a ray is constructed from query location and the intersection location. The dot product of this ray vector with the triangle's outward normal vector is used to determine the sign of the distance value. DAGMC maintains enough information to consistently orient triangle normals such that they point outward from the volume they represent. In the rare cases for which the dot product of these vectors is ambiguous, or zero, DAGMC's point containment algorithm is used to disambiguate the value's sign.

The initial implementation of the signed distance field employs MOAB's structured mesh interface and data tagging for storage. This interface maintains a representation of the structured mesh with vertex coordinates and handles at each point along with hex elements for each mesh voxel. These vertex coordinates and hex elements can be accessed using $<i,j,k>$ indexing. While this format is somewhat memory intensive, it provides a fast path for verification, visualization, and proof of concept in the transport test cases for demonstration. One can imagine a much less memory intensive implementation in which only a box corner, grid step size, dimensions, and the signed distance value data are stored. A transition to an implementation like this will likely occur before application to production work.

Signed distance values can be retrieved from the structured mesh by determining which mesh voxel the point lies within. The point's element is accessed by determining an $<i,j,k>$ index using the point's x, y, and z values divided by the structured mesh step size. A trilinear interpolation of the 8-element vertex coordinates and their signed distance values is then used to provide the signed distance value for the location of interest. As a result, the complexity of a signed distance value lookup from the signed distance field is $O(1)$. Performance improvements for geometric queries are based upon using this process in place of the $O(\log(N))$ ray tracing process.

3.1. Signed distance field application in DAGMC

The signed distance field can be used to accelerate common Monte Carlo geometry queries by providing information about the nearest surface crossing in any direction in a very similar manner to the way this is accomplished in CSG representations.

Three types of geometric queries are common among the various Monte Carlo codes that DAGMC supports. These include next surface intersection, point containment, and closest boundary intersection queries. Typically, a ray is fired to satisfy any of these queries in DAGMC with $O(\log(N))$ complexity, but it is hypothesized that these queries can be accelerated in many cases by first performing an $O(1)$ signed distance value lookup to precondition ray fire calls and make sure they are necessary.

Point containment queries can be performed by examining the signed distance value for the point. If the point's value is negative (or outside of the signed distance field data structure), then the point is not contained within the volume. If the point's value is positive, then the point is determined to be inside the volume. Given that there is error associated with each of these interpolated values, the point containment using a signed distance field should only be trusted if the absolute value of the signed distance is greater than the expected error associated with the value. If this is not the case, then a ray must be fired to determine the particle's containment with respect to the volume in question.

Closest to boundary queries can be performed in a similar manner to the point containment queries, but they are more dependent on the native code's intent for their use. Some Monte Carlo codes query for the nearest volume surface intersection to determine whether the particle will exit the volume before reaching its next physics event location. Signed distance fields are designed for exactly this operation. In similar fashion to the point containment case, the signed distance value should only be trusted if it is greater than the error associated with the value. Additionally, the error should be subtracted from the value, returning to the code a conservative value for the nearest intersection. If the signed distance value's magnitude is not greater than its error evaluation, or if the value is negative, then a ray should be fired to determine the exact location of the nearest boundary crossing for the particle's location.

Next surface intersections are called by native Monte Carlo codes to determine if a particle will cross a surface before reaching its next physics event location. This is the most common geometry query in an average Monte Carlo simulation. Normally in DAGMC a ray is fired each time this query is called. This can be avoided by using the signed distance field to exclude the possibility of a surface crossing without explicitly determining the next surface intersection. If the sum of the signed distance values for both the current particle position and the next physics event location is greater than the distance between the two, then no surface crossing will occur and the particle can safely advance to the next physics event location. For robustness, the error for each interpolation should be subtracted from the sum of the signed distance values as a conservative measure. If the expanse between the particle's current location and its next physics event interaction cannot be accounted for by the signed distance values of the two points, then a ray will be fired to determine the particle's next surface intersection along that trajectory. Fig. 4 shows a graphic representation of this process. Not all Monte Carlo codes provide the next physics event location along with the particle's current location to their geometry kernels. In this case, preconditioning of these queries will not be possible.

Using these methods, the storage of signed distances field for volumes in DAGMC could provide a way to accelerate the Monte Carlo queries listed above by using a $O(1)$ process to establish that the conditions of a geometric query are such that a more robust and computationally expensive ray is necessary before performing the

ray tracing operation. It is hypothesized that in many cases during radiation transport, this process can be used to subvert many ray tracing calls and a large number of these $O(\log(N))$ searches can be avoided. The application of the signed distance field data structure in several test problems indicates that this is true.

3.2. Signed distance value error estimation

A critical component of using a sampled signed distance function as a ray firing preconditioner is the evaluation of an error limit for the interpolated signed distance values that allow DAGMC to maintain its robust particle tracking. The failure mode of concern is the scenario in which an inaccurate signed distance value results in a particle being numerically moved outside of the current volume without logically leaving the volume. This will result in a lost particle or incorrect tallying of the particle's history caused by the decoupling of the particle's logical position within the model with the numerical position in space. Such a scenario would arise from situations such as that shown in Fig. 4C, in which the interpolation error is underestimated. However, overly conservative values of the error estimation will result in lower utilization of the preconditioner, so it is important not to overestimate the error by too large a margin.

The error of linear interpolation is well understood when derived from continuous functions with nonzero second-order derivatives. The formula for the error of a bilinear interpolation is provided in Eq. 4. Extending this representation to 3D and examining the dominant terms of this formulation, the error of the trilinear interpolation is nominally $O(h^3)$ where h is the step size of the structured mesh. Considering only these terms, the signed distance value can be found to a high degree of accuracy for a relatively coarse mesh, but it would be unwise to ignore the remaining terms representing the principal curvature of the surface, particularly given the geometric complexity of the models commonly analyzed using DAGMC. Unfortunately, triangulated surfaces do not provide a continuous function from which this principle curvature can be evaluated, nor is there a simple way to determine the second-order derivative of the mesh at a given point without encountering a secondary error estimate for that value as well.

$$\varepsilon = \frac{1}{2}\Delta x(h-\Delta x)\frac{\partial^2 u}{\partial x^2} + \frac{1}{2}\Delta y(h-\Delta y)\frac{\partial^2 u}{\partial y^2} \tag{4}$$

h – *mesh interval size* (cm)
Δx – *x distance to interpolation point from data point* (cm)
Δy – *y distance to interpolation point from data point* (cm)
$u(x,y)$ – *sampled function on mesh* (cm)
ε – *error* (cm)

Until a better solution is settled upon to account for the second-order terms, a more aggressive error estimate has been employed. For the work presented here, the error is evaluated as the length of a mesh element's longest diagonal $(3\sqrt{h})$. This represents an upper bound on the error of the signed distance value, as this is the farthest a location can possibly be from any of the data points while still being contained by that mesh element. This evaluation of the error has the effect of removing any data in mesh elements which intersect with a volume's surface. This provides a robust manner of using the signed distance field even if utilization could be higher for a more optimistic approach to calculating the interpolation error.

4. Initial performance results

The signed distance field (SDF) was first applied to DAGMC coupled with MCNP5 v1.6 [10] known as DAG-MCNP5. This data structure was implemented for a simple single-volume sphere model to test its utilization and demonstrate performance improvements. The model consists of a single 10-cm-radius sphere filled with hydrogen of density 1 $\frac{g}{cm^3}$ and a 5-MeV neutron point source at the sphere's origin. Flux and energy deposition tallies were applied to this volume for result verification between MCNP5, DAG-MCNP5, and DAG-MCNP5 with the signed distance field preconditioner applied. Fig. 3 shows the signed distance field (w/SDF) for a sphere of radius 10 cm using MOAB's structured mesh interface with a step size of 0.5 cm. The design intent in this test was for the particles to have an average mean free path significantly smaller than the size of the sphere. This highly scattering medium is expected to provide a good problem scenario for the preconditioner as it is most effective when particle interactions are frequently occurring far from surface interfaces. Simulations of 100 million histories were run for each implementation listed above. Signed distance field preconditioning was only implemented for next surface intersection queries.

A simple sphere presents a worst case for DAG-MCNP5 in comparison to a CSG implementation such as the one that exists in MCNP5. Analytic geometry queries are very simple even in comparison to other CSG objects, let alone in comparison to a sphere representation composed of hundreds of thousands of triangles, even with DAGMC's existing accelerations.

The results of this initial test can be seen in Table 1. This initial test case indicates that signed distance field preconditioning was quite effective—accelerating the simulation in comparison to a normal DAGMC run by a factor of 3.5. It should be noted that no difference was found in either the flux or energy tallies between the two DAG-MCNP5 runs.

5. Signed distance field preconditioner utilization

In effect, the preconditioner is attempting to check whether the particle will cross a surface before explicitly searching for the particle's intersection with a surface along its current trajectory. If the result of this preconditioning check is always false and a ray is always fired, then these checks are only adding to the computational cost of the problem. This will always occur in volumes filled with void, for example, as particles immediately travel from one surface to another. As a result, the signed distance field may need to be applied selectively depending on each volume's geometric and material properties for optimal performance and high utilization of the preconditioning methods. Ideally, this method will only be applied to volumes in which the data structure can precondition ray fire calls often or with high utilization of the data structure. The signed distance field is expected to have the biggest impact in performance when preconditioning next surface intersection queries, as they are most frequently called in Monte Carlo codes when tracking particles through the geometry. As such, this type of query is the focus of utilization measurement for the remainder of this section.

$$U = \frac{\text{Rays Avoided w/ SDF}}{\text{Number of Geometry Queries}} \tag{5}$$

The utilization, U, of the signed distance field as a ray fire preconditioner can be described as the number of ray fire calls related to the next surface intersection queries that are avoided, divided by the total number of next surface intersection queries made by the Monte Carlo code. This value is described in Eq. 5 and can be quantified using this definition using debugging tools, such as Valgrind, during DAGMC simulations. It is expected that in most cases, as the utilization of preconditioning methods goes up, the performance of the simulation will also improve.

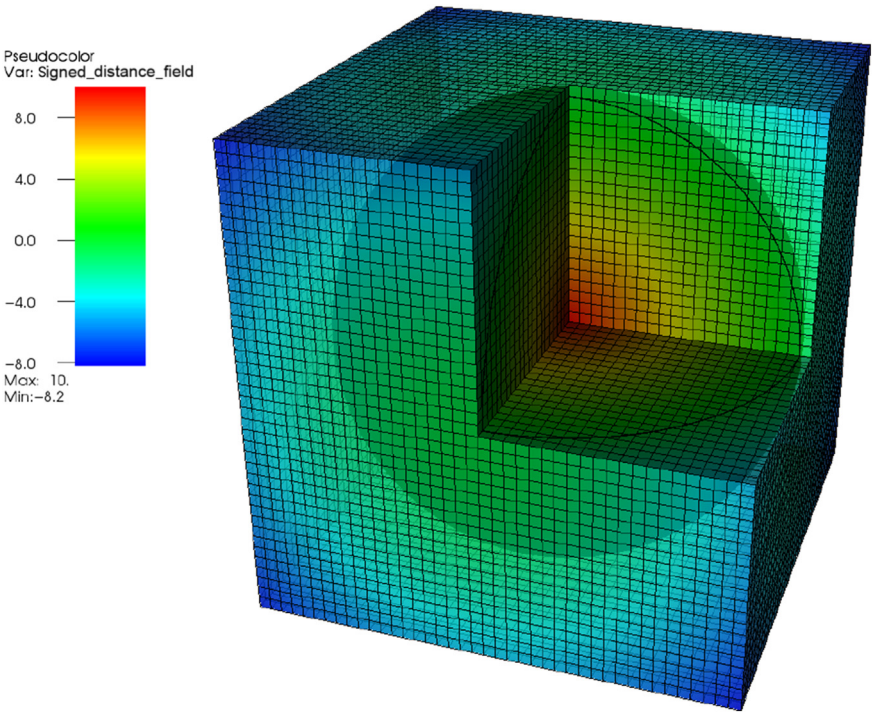

Fig. 3. A visual of a signed distance field with step size 0.5 cm surrounding the spherical volume of test case with a radius of 10 cm.

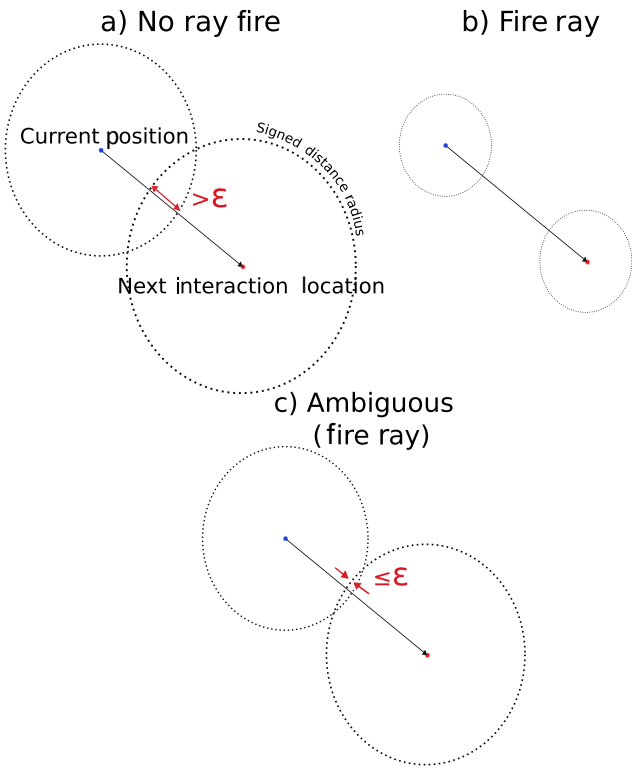

Fig. 4. Visualization of ray preconditioning scenarios. The ε here represents the associated error of the signed distance value interpolation.

To understand this utilization more deeply with respect to material parameters, the hydrogen density was varied from 0 to 1 $\frac{g}{cm^3}$ in the single-volume sphere test problem with a 5-MeV neutron point source. For each density, one simulation was performed without the signed distance field and another with the signed distance field and preconditioning enabled. Fig. 6 shows the results of this study.

Table 1
Initial results of a signed distance field query preconditioner in DAGMC for a 5-MeV point source at the center of a single 10-cm-radius hydrogen-filled sphere ($1 \frac{g}{cm^3}$) test case. One hundred million histories were run in each implementation.

Implementation	ctme (min)	wall time (min)	time ratio	precond. utilization
MCNP5	2.75	2.75	1	N/A
DAG-MCNP5	25.04	25.13	9.13	N/A
DAG-MCNP5 w/SDF	7.15	7.3	2.65	0.97

Utilization of the data structure in this study remains high until the hydrogen density falls to 0.1 $\frac{g}{cm^3}$ at which point a distinct knee appears and the utilization falls off quickly. Even at the lowest density reached in the study of 0.01 $\frac{g}{cm^3}$, the utilization of the signed distance field to avoid ray fire calls is 0.54. It is difficult to judge the impact on the performance of this simulation for these low-density values due to the limited size of the geometry and the short-lived histories. As the material density decreases, particles quickly leave the geometry after very few collisions, but Fig. 5 provides an impression of the performance of these three implementations converge as the density of the hydrogen is varied. The application of the signed distance field allows for significantly improved performance until the density drops below 0.1 $\frac{g}{cm^3}$ in agreement with utilization plot. To have more control over a simulation's physical parameters, subsequent experiments were performed using a simple simulation tool.

6. Signed distance field utilization modeling

To characterize utilization of a signed distance field as a next surface intersection query preconditioner for ray fire calls in DAGMC, a pseudo Monte Carlo simulation tool was developed using

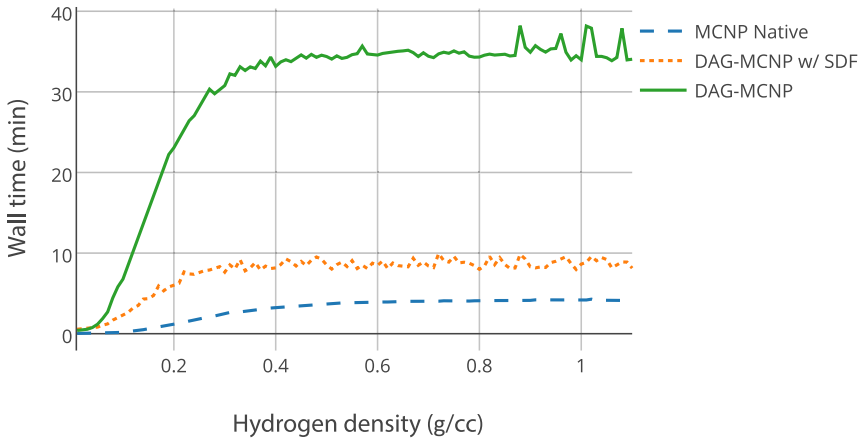

Fig. 5. Performance results for a 5-MeV neutron source at the origin of a 10-cm-radius sphere. Hydrogen density was varied from 0 to 1 $\frac{g}{cm^3}$. Simulations of 100 million histories at each density were performed using native MCNP5, DAG-MCNP5 without the signed distance field, and DAG-MCNP5 with the signed distance field.

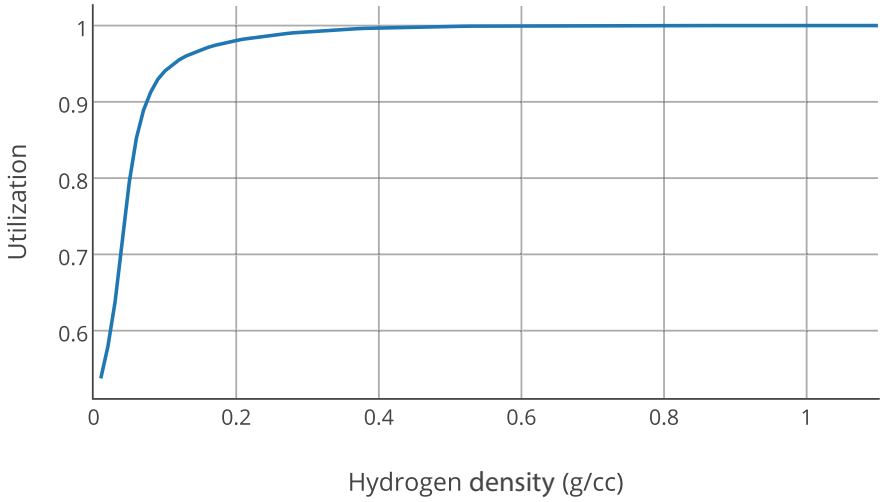

Fig. 6. Utilization results for a 5 MeV neutron source at the origin of a 10 cm radius sphere. Hydrogen density was varied from 0 to 1 $\frac{g}{cm^3}$.

DAGMC. This tool was used to simulate different transport scenarios within a spherical geometry using an isotropic volumetric source and isotropic scattering. Particle histories are terminated based on a maximum number of collisions or departure from the problem geometry. Particle distance traveled, d, can be represented by either a fixed distance or by sampling for the standard probability of interaction in a medium with mean free path, λ. The tool allows the mean free path to be set directly, enabling a relation between the signed distance field and this value to be developed with the intent to use this relationship as a means for characterizing appropriate conditions for application of the signed distance field.

To begin with, simulations were performed for particles with a fixed distance traveled, λ, varying that distance and the signed distance field step sizes. Run times of the simulation are not shown here as the data structure's utilization is the focus of this study. The results of this study are shown in Fig. 7. As the signed distance field mesh step size increases, utilization of the data structure decreases due to the increasing error associated with the interpolation of signed distance values. Additionally, utilization is expected to decrease with increasing distance traveled. This decreased utilization is caused not only by the increased distance between the two particles, but also by the increased probability that both locations will be closer to surfaces of the sphere and have smaller signed distance values. A theoretical limit for the utilization is also shown in Fig. 7. The development of the analytic form for this limit will now be discussed.

The utilization of the signed distance field as a preconditioner for ray tracing operations can be modelled as an evaluation of the combined probability space for particles with a current position, $\vec{p}$, and a next physics event location, $\vec{n}$, after travelling a distance, d. The fraction of this probability space in which signed distance values can be used to rule out surface crossings for next surface intersections is then considered to be the theoretical utilization of the signed distance field. An initial form for this probability space can found in Eq. 6.

$$\int_{V_{sphere}} \int_{V_{track}} p_p(r) p_n(d)\, \mathrm{d}V_{sphere} \mathrm{d}V_{track} \tag{6}$$

In this model, the starting location of particles, $\vec{p}(r, \phi, \theta)$, is uniformly distributed, $p_p(r) = 1$, throughout a sphere of radius, R. The location of the next event, $n(d, \alpha, \beta)$, where d is the distance traveled by the particle, α is the interior angle between the particle's *position* vector and the particle's sampled direction vector, and β represents an azimuthal angle for directions traveled with angle of departure α. Fig. 8 depicts these variables, r, d, and α more clearly.

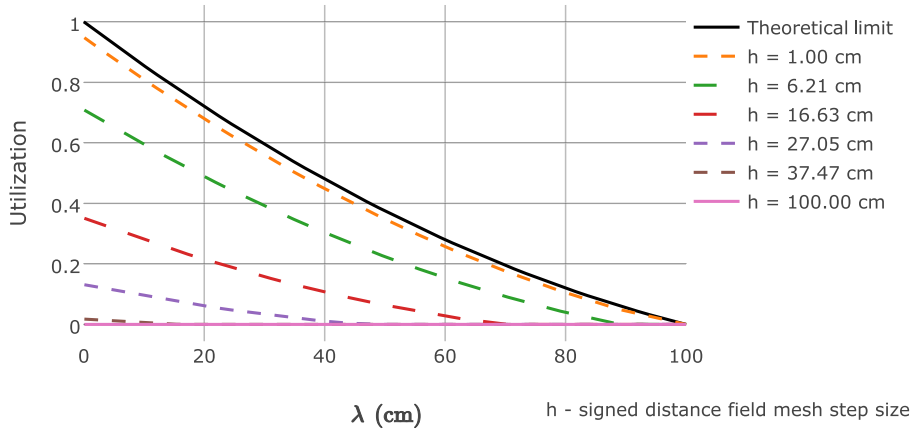

Fig. 7. Results of the model for the theoretical utilization limit with the results of the simulation for a fixed distance traveled case.

These points are distributed uniformly on a sphere at some distance d, and that distance is distributed according to some probability density function, $p_n(d)$.

In Eq. 6 the outer integral represents all possible particle positions within the geometric sphere and expands to

$$\int_0^R \int_0^{2\pi} \int_0^{\pi} \int_{V_{track}} r^2 \sin\phi \, d\phi d\theta dr \, p_n(d) dV_{track} \tag{7}$$

The second integral then expands to

$$\int_0^R \int_0^{2\pi} \int_0^{\pi} \int_0^{\infty} \int_0^{2\pi} \int_0^{\pi} r^2 \sin\phi \, p_n(d) d^2 \sin\alpha \, d\alpha d\beta dd \, d\phi d\theta dr \tag{8}$$

Integration of ϕ, θ, and β can now be performed with the knowledge that they are symmetric with respect to the problem and integration of $p_n(d)$ does not rely on them.

$$8\pi^2 \int_0^R \int_0^{\infty} \int_0^{\pi} p_n(d) \, r^2 \, d^2 \sin\alpha \, d\alpha dd \, dr \tag{9}$$

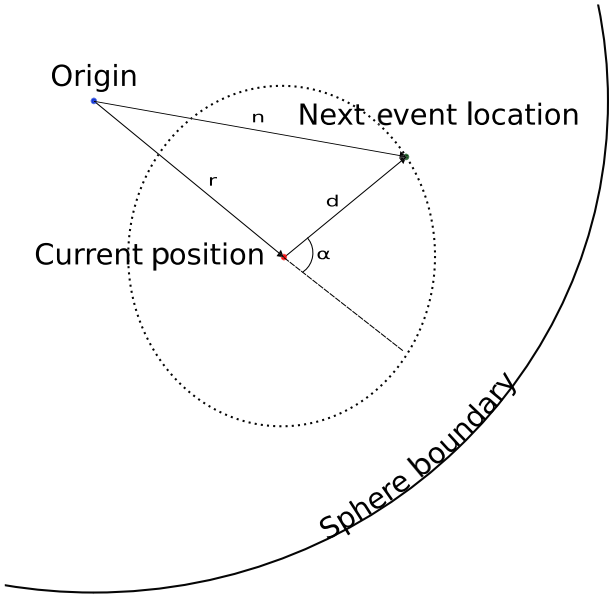

Fig. 8. Depiction of model variables.

To represent particles travelling a fixed distance, the relationship in Eq. 10 is applied.

$$p_n(d) = \frac{\delta(d-\lambda)}{d^2} \tag{10}$$

The evaluation of this integral then gives a representation of all the query space available to the problem

$$A = 8\pi^2 \int_0^R \int_0^{\infty} \int_0^{\pi} \delta(d-\lambda) \, r^2 \sin\alpha \, d\alpha dd \, dr \tag{11}$$

and represents all geometric query space, A, for a sphere of radius, R and a fixed distance traveled, λ.

To understand what fraction of this query space can be preconditioned, the condition for avoiding an explicit nearest intersection search along a particle direction in Eq. 12 must now be applied. Because this is intended to be an idealized upper limit for the utilization, error will be ignored.

$$SDV(\vec{p}) + SDV(\vec{n}) > |\vec{p} - \vec{n}| + 2\varepsilon(h) \tag{12}$$

SDV – signed distance value function
$\vec{p}$ *– particle's current position*
$\vec{n}$ *– particle's next event location*
h *– mesh step size*
$\varepsilon(h)$ *– error evaluation for signed distance values*

This condition establishes that the nearest location to intersection for both points must be greater than the distance between the two points plus any error associated with their signed distance values as previously discussed. This condition is true for some fraction of the next surface queries in a Monte Carlo simulation, but not all. Our mathematical model has no error, allowing for the largest possible volume of locations in which preconditioning will apply and the signed distance value for any point, $\vec{x}$, is

$$SDV(\vec{x}) = R - |\vec{x}| \tag{13}$$

Making these substitutions into the inequality gives

$$R - |\vec{p}| + R - |\vec{n}| > |\vec{p} - \vec{n}| \tag{14}$$

The right-hand side of this inequality can simply be described as the distance traveled, d, and the magnitude of $\vec{p}$ can be represented by the variable r.

$$R - r + R - |n(d, \alpha, \beta)| > d \tag{15}$$

Reducing the next event location, $\vec{n}(d, \alpha, \beta)$, into an expression in terms of r, d, and α requires further examination of the problem. Because the coordinates of n depend on the current particle position, the magnitude of n with respect to the geometry origin must be obtained to get a correct form for the signed distance value. Again, Fig. 8 depicts the value of n graphically for reference. The magnitude of n can then be described using the law of cosines as

$$\left|n(d, \alpha, \beta)\right| = \sqrt{r^2 + d^2 - 2rd\cos\pi - \alpha} \tag{16}$$

inserting this into the inequality gives

$$R - r + R - \sqrt{r^2 + d^2 + 2rd\cos\alpha} > d \tag{17}$$

The inequality has now been reduced to the three variables seen in Eq. 11 (r,d, and α). This inequality can be applied to construct limits of integration representing boundaries of space in which the SDF can be utilized. By rearranging the inequality, a limit on the angle of departure, α, from the particle's position can be derived.

$$\alpha_{min} > \arccos\left(\frac{(2R - r - d)^2 - d^2 - r^2}{2dr}\right) \tag{18}$$

This condition on alpha can be interpreted as a minimum interior angle that the particle's trajectory must take relative to the particle's position vector, $\vec{p}$, for a distance traveled, d, for a ray fire to be avoided and the preconditioner to be utilized. The examination of this condition as a function of the distance traveled for various values of r results in some conclusions about how signed distance values are being utilized.

The inequality is undefined until the distance reaches a value $d = R - r$. This is because the angular limit only needs to be applied to areas of the query space in which the distance traveled is large enough to violate the above condition as depicted in Fig. 9. Fig. 10 demonstrates that a violation of this limit may only occur when a particle travels far enough to reach the geometric sphere boundary along the current position vector as if it were moving directly toward the boundary of the sphere. An additional interesting feature of this plot is the convergence of all the curves as $d \to R$ on π. The convergence on π indicates that as the distance traveled approaches R, the only direction that the particle can move is back toward the origin along the position vector. It also defines a maximum distance a particle can travel in the sphere and still be preconditioned using signed distance values. Intuitively this makes sense, as the maximum chord length of a sphere is $2R$, and once a particle travels a distance R the c in Eq. 12 is violated. Hence all curves go to zero at $\lambda = 100cm$ in Fig. 7.

To account for the fact that the form of α_{min} is undefined until $d = R - r$, a Heaviside function is applied before applying it as a limit on the particle's angle of departure from the position vector. Similarly, because the α_{min} condition is undefined after $d = R$ a Heaviside function is used to limit the condition to π for any distances traveled larger than R.

$$\alpha_{min} = (H(d - (R - r)) - H(d - R))\arccos\left(\frac{(2R - r - d)^2 - d^2 - r^2}{2dr}\right) + \pi H(d - R) \tag{19}$$

By inserting this condition as a lower limit of the $d\alpha$ integration, the following integral will give all utilized space, US, in the query space of the simulation.

$$US = 8\pi^2 \int_0^R \int_0^\infty \int_{\alpha_{min}}^\pi \delta(d - \lambda)\, r^2\, d^2 \sin\alpha \, d\alpha dd\, dr \tag{20}$$

Evaluating this integral and dividing by all query space gives the following form for the theoretical limit of signed distance field utilization as a preconditioner for ray firing.

$$U_{theoretical} = \frac{US}{A} = \frac{(1 - H(\lambda - R))(2R - \lambda)(R - \lambda)}{2R^2} \tag{21}$$

It can be seen in Fig. 7 that this utilization limit works well as an upper limit for the simulation results using various signed distance field mesh resolutions. As the step size of the mesh approaches zero, so does the evaluation of the error, resulting in the same utilization curve with varying distance traveled, λ, as in the analytic form developed here. Future work will include the comparison of this utilization limit to other single-volume geometries using dimensionless parameters to determine if the model above can be used to predict signed distance field utilization in other geometries as well.

With the agreement of the simulation results and analytic model for signed distance field utilization for the fixed distance traveled case, the simulation has been used to produce a similar set of results in which the distance is sampled based on the standard probability for distance to interaction in a medium with a cross section, Σ, or mean free path $\lambda = 1/\Sigma$. This results in the probability distribution function shown in Eq. 22 for the particle distance traveled in this scenario.

$$p_n(d) \propto \frac{e^{-\Sigma d}}{d^2} = \frac{e^{-\frac{d}{\lambda}}}{d^2} \tag{22}$$

Following the same process as in the fixed distance case by plugging Eq. 22 into Eq. 20, the utilization form for the sampled distance case is shown in Eq. 23.

$$U_{theoretical} = \frac{US}{A} = \frac{\frac{1}{2}\lambda(R - 2\lambda)e^{\frac{-R}{\lambda}} + \lambda^2 - \frac{3}{2}R\lambda + R^2}{R^2} \tag{23}$$

The results of this set of simulations can be seen in Fig. 11. In this scenario, it is not expected that the utilization will approach zero when $\lambda = 100\ cm$, as the actual distance sampled may be considerably less than the provided mean free path for the simulation. Overall utilization values in this scenario for λ from 0 to 100 cm remain higher than the corresponding fixed distance simulation cases as is expected in a sampled distance case. Utilization values remain high for relatively large increases in mesh step size, h. This is important to application of the data structure given concerns regarding its potentially high memory footprint for large volumes. For example, if the utilization of the signed distance field drops 20% when going from a step size of 1 cm to 6.21 cm, the memory footprint of the data structure will have decreased by a factor of 6.21^3 or 239.5 as well. The optimization of the mesh step size with respect to its effect on utilization will also need to be included in future models of the utilization.

7. Charged particle transport results

Signed distance field preconditioning is likely to have high utilization in problems involving charged particle transport due to the straggling paths Monte Carlo codes use to model their behavior. As support for charged particle transport becomes more common in various Monte Carlo codes, these problems are encountered more and more frequently. During development of a DAGMC interface to MCNP6 [11], several electron transport problems in the test suite were found to have egregiously long run times in DAGMC—on the order of 5,000–10,000 times longer than native MCNP6.

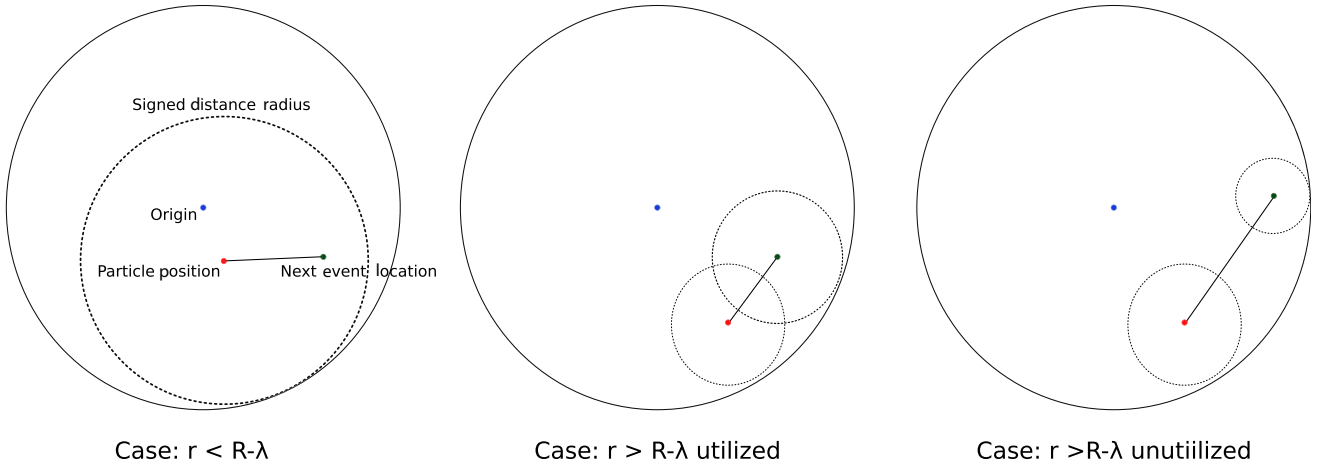

Fig. 9. Depiction of modeling cases.

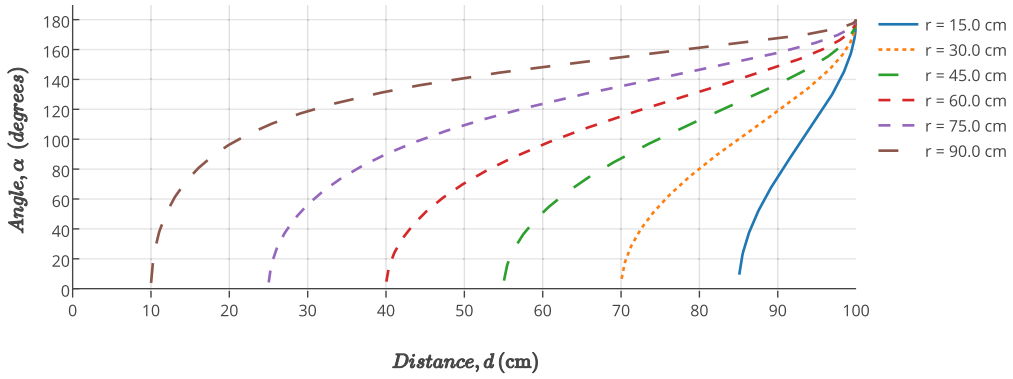

Fig. 10. Plot of minimum angle of departure restriction for particles with various radial positions in a sphere with R = 100 cm and varying distances traveled.

Profiling of the test problems revealed that closest to boundary geometry queries were called more than any other. This is a common occurrence in charged particle transport. Typically, the Monte Carlo code is attempting to retrieve the nearest surface intersection from a given location, a property a signed distance fields or implicit surfaces are well-suited for obtaining. This distance is used to determine if the particle is approaching a surface as it moves along its straggling path through the current medium.

The test problem to which signed distance field preconditioning was applied consisted of a 1- to 100-keV photon source incident on a Fe/W target with the intent of testing secondary generation of electrons and their resulting simulation. This problem's geometry is composed of two halves of a 10-cm-radius sphere divided into two hemispheres in the z-axis. In one void hemisphere exists the photon source and in the other hemisphere the Fe/W target. The unit test default of 5,000 histories were run in each case, and results of the surface current tally in both implementations of DAGMC were the same as those of the MCNP6 output file.

By implementing closest to boundary query preconditioning for this test problem, a substantial improvement in performance was

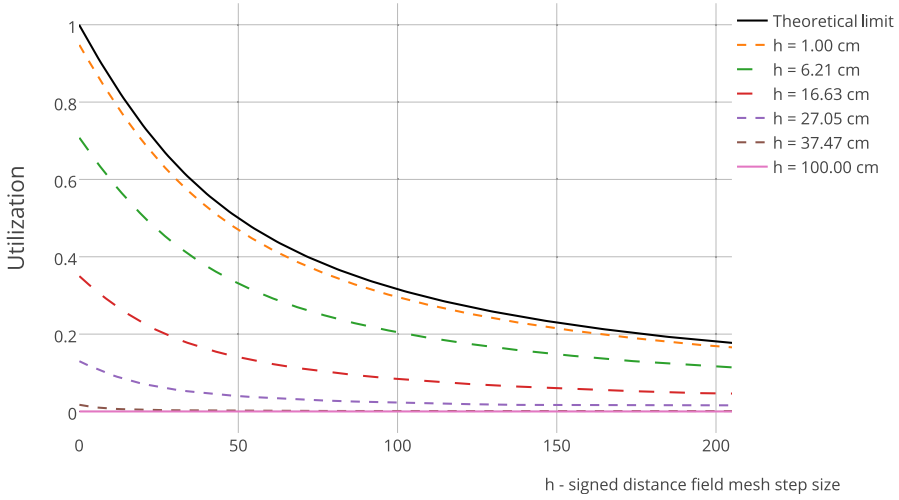

Fig. 11. Results of the model for the theoretical utilization limit with the results of the simulation for a sampled distance traveled case.

gained, as shown in Table 2, without change in the result of the test tallies. It should be noted that in this case, the error evaluation used was less conservative than in other simulations to avoid the high memory consumption of a fine mesh. Because particle transport is occurring so close to a boundary, the signed distance values retrieved from the signed distance field are very small. The condition for an avoided ray fire requires that the error evaluation be on the same order as the signed distance value. With the error being proportional to the resolution of the uniform mesh on which the signed distance field is stored, this means either making the mesh very fine or relaxing the error evaluation for the data structure to be well utilized in this case.

Table 2
Performance results for an MCNP6 test case involving electron transport of a 1- to 100-keV photon source incident on an Fe/W target. Five thousand histories were run in this test problem.

Source Location: <0,0,−1>				
Implementation	ctme (min)	wall time (min)	time ratio	precond. utilization
MCNP6	0.17	0.14	1	N/A
DAG-MCNP6	1,841.33	1,841.33	11,000	N/A
DAG-MCNP6 w/SDF	0.48	0.46	2.82	0.94
Source Location: <0,0,10>				
Implementation	ctme (min)	wall time (min)	time ratio	precond. utilization
MCNP6	0.18	0.18	1	N/A
DAG-MCNP6	11.12	11.16	62	N/A
DAG-MCNP6 w/SDF	0.50	0.52	2.89	0.96

To show that this performance improvement is still seen in a less strenuous case in terms of the error evaluation, the same photon source was placed within the Fe/W target at a location of $<0,0,10>$. Because this new source location results in larger signed distance values during transport, the error evaluation used in this scenario is the same as in the hydrogen-sphere demonstration above. The performance improvement for DAG-MCNP6 when employing the signed distance field for this scenario is considerably smaller in terms of the ratio to the native MCNP6 run, but still provides a large improvement over DAG-MCNP6 alone as seen in the lower section of Table 2. The large ratios between the runtimes of native MCNP6 and DAG-MCNP6 indicate in both cases indicates that some implementation of this data structure may be a necessity going forward for analysis of detailed charged particle transport in DAGMC geometries.

8. Conclusions

This work has shown that signed distance fields can be effectively used to accelerate CAD-based Monte Carlo simulations in DAGMC for several different problem scenarios. It was also shown that this technique can be applied without the use of an analytic implicit surface representation by using MOAB's ray tracing kernel to populate the data structure.

A considerable amount of work related to this data structure has yet to be conducted to make it a viable tool for models with hundreds or thousands of volumes as is commonly seen in analysis work with DAGMC. Of concern is the memory footprint such a data structure might incur on larger models. It is possible that adaptive mesh refinement techniques such as the Octree [12] might be necessary despite an expected reduction in preconditioner utilization. Other interesting methods for sparse uniform mesh representations which can maintain an $O(1)$ look up may be investigated as well [13].

Another area that has yet to be studied is point containment preconditioning using the signed distance field. As each Monte Carlo code has its own way of tracking particles, this query is performed more often in some codes than others and could also provide a significant improvement in performance.

Despite the data structure being used, if the signed distance field is present but not being utilized well, then it is only adding to the computational cost of the simulation. It will be important going forward to have a well-understood set of conditions for which the signed distance field can be applied in an advantageous fashion. One problem property that has already been shown to be a strong factor is the particle mean free path or collision density in a given volume. This is an extremely difficult value to predict a priori when conducting analysis on models with many volumes, each with their own set of important characteristics in the context of the problem. These values are usually tracked by Monte Carlo codes during run time. To avoid manual selection of volumes for application of the signed distance field, it may be possible to apply the data structure by pre-computing geometric characteristics during problem setup and track problem-specific quantities such as the mean free path during simulation. If the proper conditions for utilization are met, then the data structure could be applied on-the-fly, enabling better performance in DAGMC as the simulation continues. It is arguable that in combination with previous work related to an accelerated ray tracing implementation [4], use of this data structure could provide CAD-based radiation transport on par with native Monte Carlo geometry representations under the right conditions.

Conflicts of interest

All authors have no conflicts of interest to declare.

Acknowledgements

This work was funded in part by project NRC-HQ-84-14-G-0030 from the U.S. Nuclear Regulatory Commission Fellowship program as well as projects DE-FG02-99ER54513 and DE-SC0017122 from the U.S. DOE Office of Fusion Energy Sciences.

References

[1] T.J. Tautges, P.P.H. Wilson, J. Kraftcheck, B.M. Smith, D.L. Henderson, Acceleration Techniques for Direct Use of CAD-based Geometries in Monte Carlo Radiation Transport, in International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009), American Nuclear Society, Saratoga Springs, NY, May 2009.
[2] T.J. Tautges, R. Meyers, K. Merkley, C. Stimpson, C. Ernst, MOAB: a Mesh-oriented Database, SAND2004-1592, Sandia National Laboratories, Apr. 2004. report.
[3] B.M. Smith, Robust Tracking and Advanced Geometry for Monte Carlo Radiation Transport, PhD Nuclear Engineering and Engineering Physics, University of Wisconsin-Madison, Madison, WI, United States, 2011.
[4] P. Shriwise, A. Davis, P.P. Wilson, Leveraging Intel's Embree ray tracing in the DAGMC toolkit, in: Am. Nuc. Soc. Winter Meeting 2015, vol. 113, Nov 2015. Washington, DC, USA.
[5] H. Weghorst, G. Hooper, D.P. Greenberg, Improved computational methods for ray tracing, ACM Trans. Graph. 3 (1) (Jan. 1984) 52–69.
[6] S. Gottschalk, M.C. Lin, D. Manocha, OBBTree: a hierarchical structure for rapid interference detection, in: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1996, pp. 171–180.
[7] J. O'rourke, Finding minimal enclosing boxes, Int. J. Comput. Inform. Sci. 14 (3) (1985) 183–199.
[8] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Applied Mathematical Science, Springer, New York, N.Y., 2003.
[9] L.J. Tomczak, GPU Ray Marching of Distance Fields, Technical University of Denmark, 2012.
[10] X-5 MONTE CARLO TEAM, MCNP - a General Monte Carlo N-particle Transport Code, Version 5 - Volume III: Developers Guide, Tech. Rep. LA-CP-03-0284, Los Alamos National Laboratory, Jun. 2004.
[11] J.T. Goorley, M.R. James, et al., Initial MCNP6 Release Overview - MCNP6 Version 1.0, Tech. rep., jun 2013.
[12] A.S. Glassner (Ed.), An Introduction to Ray Tracing, Academic Press Ltd., London, UK, 1989.
[13] R. Setaluri, M. Aanjaneya, S. Bauer, E. Sifakis, SPGrid: a sparse paged grid structure applied to adaptive smoke simulation, ACM Trans. Graph. 33 (6) (Nov. 2014) 205:1–205:12.