Layered CAD/CSG Geometry for Neutronics Modeling of Advanced Reactors

Elliott Biondo[1], **Gregory Davidson**[1], **and Brian Ade**[1]
[1]Oak Ridge National Laboratory
1 Bethel Valley Rd., Oak Ridge, TN 37830, USA

veb@ornl.gov, gqe@ornl.gov, ba7@ornl.gov

ABSTRACT

Constructive solid geometry (CSG) has traditionally been sufficient for Monte Carlo (MC) neutron transport simulations of standard light-water reactors (LWRs). Advanced reactor designs, as well as demand for multiphysics coupling and ex-core analysis, have created opportunities to use computer-aided design (CAD) to model a subset of reactor components. For scenarios where it is necessary to combine CAD and CSG models, a *layered* geometry method has been implemented in the Shift MC code. With this method, multiple CSG and/or CAD models can be arbitrarily clipped, translated, rotated, and placed in overlapping layers with assigned precedence to form transport-ready geometries. This is demonstrated with an MC k-eigenvalue calculation using a layered geometry model of the Transformational Challenge Reactor (TCR) consisting of complex CAD fuel elements placed in a CSG core model.

KEYWORDS: Monte Carlo radiation transport, CAD geometry, advanced reactors

1. INTRODUCTION

Production-level Monte Carlo (MC) radiation transport codes increasingly support particle tracking on surface or volume meshes generated from computer-aided design (CAD) models. Shift [1], OpenMC [2], and modified versions of MCNP6 [3] can track particles on surface meshes via the Direct Accelerated Geometry Monte Carlo (DAGMC) library [4]. Serpent 2 [5] can track particles on stereolithography (STL) surface meshes and unstructured volume meshes, and MCNP6 can track particles on unstructured volume meshes [3]. CAD and traditional constructive solid geometry (CSG) formats differ significantly in their features, limitations, and original application, which creates challenges when models in different formats must be combined.

CSG formats are the primary, native formats of all production-level MC codes, and the vast majority of MC analysis of fission reactors is performed using purpose-built CSG models. With CSG formats, models are constructed by combining geometric primitives using Boolean logic operations. These primitives are typically limited to 3D bodies consisting of planar and/or quadratic surfaces (e.g., cuboids, cylinders, spheres), making it difficult to model complex shapes. In addition, CSG text files are verbose and minimally human-readable. Thus, modifying or combining CSG models often requires significant human effort.

CAD geometry can be used to model arbitrarily complex shapes, which can be constructed and modified in an intuitive 3D interface. However, even for simple models containing planar and quadratic surfaces, the resultant meshes used for transport require significantly more computer memory than CSG formats. Unlike CSG, CAD models are rarely purpose-built for MC transport; one of the primary use cases for CAD transport is using pre-existing models created for thermal hydraulics or structural mechanics analysis. These pre-existing models generally contain non-physical gaps and overlaps: regions that are either not contained in any cell or contained in multiple cells. MC particles may become "lost" when encountering gaps and overlaps, which can prevent the simulation from completing or introduce systematic bias into the simulation. Although there are some automatic methods for fixing these problematic regions [6], additional labor-intensive manual intervention is often required. Any time a CAD model is modified, it must be re-meshed, and this may reintroduce these problematic regions in unpredictable ways.

Traditionally, CSG geometry has been sufficient for modeling light-water reactors (LWR), as most LWR components can be reasonably approximated with geometric primitives. However, several factors have increased geometric complexity requirements, creating opportunities to use CAD for a subset of reactor components. Advances in computing have created a demand for analysis beyond eigenvalue calculations, such as coupled multiphysics or ex-core analysis. It would be convenient to use pre-existing models in these cases. Advances in additive manufacturing have expanded the range of feasible shapes for fuel elements and other components. The Transformational Challenge Reactor (TCR) [7] uses additively manufactured fuel elements, designed via CAD, that are difficult to model in CSG. Additionally, as the fidelity of simulations increases, it may become more important to use CAD models to accurately model LWR components traditionally approximated in CSG, such as spacer grids and rod springs.

Several options exist for combining CAD and CSG models. One option is to translate models into the same format. CAD models can be automatically translated to MCNP CSG using tools such as McCAD [8], but this often requires significant manual intervention [9,10]. MCNP CSG models can be translated to CAD using the MCNP2CAD tool [11], but not all MCNP cell types are supported, and using CAD as the common format might significantly increase memory requirements. Geometry format translations can also be done by hand, but this is extremely labor intensive. Another method of combining CSG and CAD models is the "universe" method. MCNP6 [12] and OpenMC [13] allow CAD universes to be embedded in CSG cells. Serpent 2 allows CAD and CSG universes to be embedded in CAD and CSG cells [14]. Serpent 2 also allows for all of the undefined space in or surrounding a CSG or CAD model to be filled in with a "background universe" [15,16]. The method described herein is most closely related to the background universe concept and retains all of these features.

In this work, a *layered* geometry capability was implemented in the Shift MC code, which allows multiple CSG and/or CAD models to be arbitrarily clipped, translated, rotated, and placed in overlapping layers with assigned precedence to form transport-ready geometries. Section 2 describes how layered geometries are constructed, and how MC tracking is performed on layered geometries. This implementation is demonstrated in Section 3 with an eigenvalue simulation on a layered geometry model of TCR consisting of complex CAD fuel elements in a CSG core. Section 4 provides concluding remarks and planned future work.

2. Methodology

Layered geometries are created by combining pre-existing models in CSG and/or CAD formats. At this time, layered geometries can be constructed using models in Shift's native general geometry (GG) CSG format, the SCALE KENO CSG format [17], and DAGMC CAD, but this could be expanded to any other geometry format supported by Shift, including MCNP CSG. These pre-existing models are referred to as *constituent* models. The manner in which constituent models are combined to create a layered geometry is illustrated in Fig. 1.

Fig. 1 shows that the first step is creating *objects* by specifying axis-aligned bounding boxes enclosing any portion of any constituent model. Any materials within the constituent model may be declared *transparent*. Multiple objects can be created from the same constituent model. For example, objects 1 and 2 come from the same constituent model but have different material transparencies. Objects 3 and 4 also come from the same constituent model but are generated using different bounding boxes. The bounding boxes for objects 3 and 4 both truncate portions of the orange cell.

The next step is creating *layers* from the collection of objects. The layer with the lowest precedence is known as the *base layer*, and must consist of a single object whose bounding box defines the spatial domain of the problem. This ensures that the geometric scene contains no undefined regions. In transport problems where no constituent model is a convenient base layer, an additional constituent model can be created, consisting of a single cell encompassing the full geometry. After the base layer, an arbitrary number of additional layers can be specified, labeled with consecutive integers representing precedence. Each non-base layer may contain an arbitrary number of objects, which may be translated and/or rotated in any position—provided that the bounding boxes of objects on the same layer do not overlap. A single object is permitted to appear multiple times on a single layer, such as object 4 on layer 2 in Fig. 1, or multiple times on different layers, such as object 3 on layers 1 and 2. In all cases, each object is stored only once in memory.

The final step is creating the layered geometry itself by simply superimposing all of the layers in order of precedence, with the base layer on the bottom of the stack, and each subsequent layer stacked on top. Though objects on the same layer cannot have overlapping bounding boxes, objects on different layers are allowed to overlap arbitrarily. Overlapping regions respect the transparency of the specified materials within objects. For example, in Fig. 1 the base layer and object 3 can be seen through the center of object 1.

To run Shift transport on a layered geometry, a user first creates a layered geometry input file. This input file denotes the file paths of the constituent models, corresponding material files, the bounding boxes and transparent materials for each object, and the translations and rotations of the objects on each layer. Transport is then run via Shift's Omnibus interface [18].

2.1. Layered Geometry Tracking

MC tracking on a layered geometry requires special considerations, which are described here briefly. The principal strategy is that tracking is done independently on each layer and is updated on each layer for every particle event. As a result, distance-to-boundary queries must always return the minimum of the distance-to-boundary across all layers. Even though a particle is logically tracked on each layer simultaneously, the particle is always in exactly one cell within the physical system. The layer that contains this cell is the *active* layer. The active layer is simply the highest precedence layer for which the particle is within a geometry cell containing a non-transparent material. This can be found by querying the layers in order of decreasing precedence. Cross section lookups and tallies are always evaluated on the active layer.

Within each layer, particle tracking is simplified by the requirement that all the bounding boxes of the objects on the layer do not overlap. If a particle is within an object bounding box, it must be tracked within the object itself. However, if a particle is not within an object bounding box, distance-to-boundary queries are inexpensive point-to-plane calculations. Since tracking is only expensive within objects, the performance of a layered geometry is expected to be closely related to the track density in regions where objects on different layers overlap, rather than simply scaling with the number of layers present in a geometry.

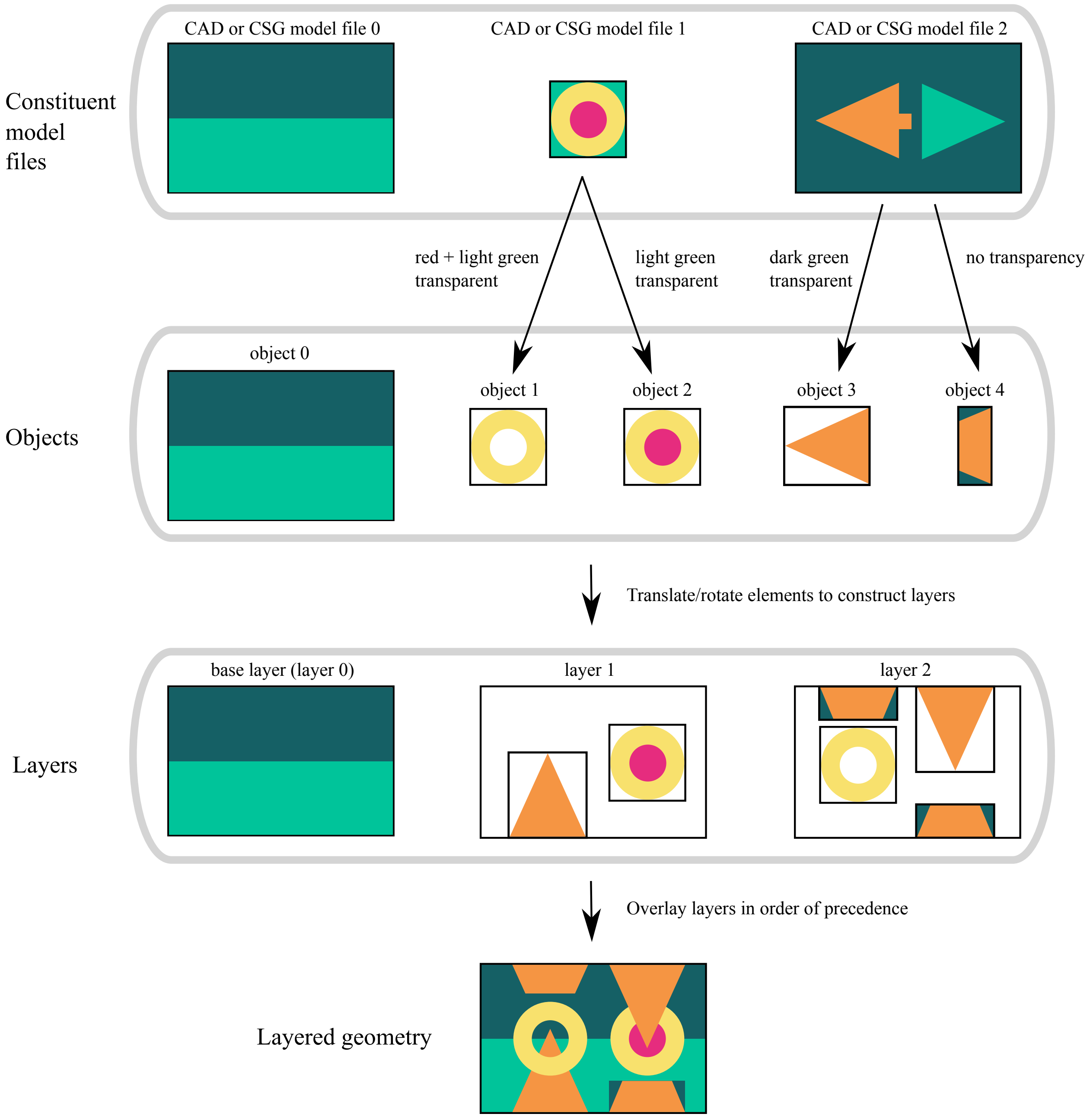


Figure 1: Illustration of the layered geometry construction process.

One disadvantage of layered geometry is robustness against lost particles. In both CAD and CSG cells, adjacent surfaces are typically merged together. As a result, when a particle exits a cell, the next cell is known automatically, without a point-in-cell call. This feature is currently absent in the layered geometry implementation within Shift. As a result, when a particle crosses a surface in such a way that the active layer changes, floating point issues can cause lost particles. Methods to address this failure mode are being investigated and will be reported on in future work.

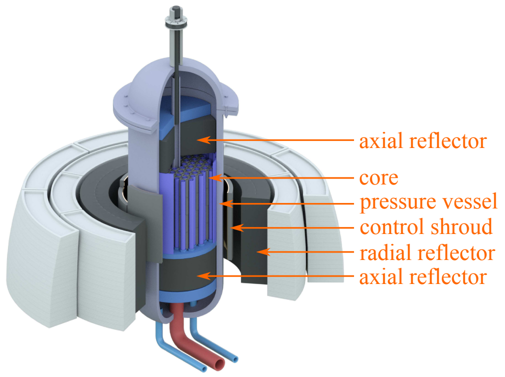


Figure 2: Rendering of the Transformational Challenge Reactor.

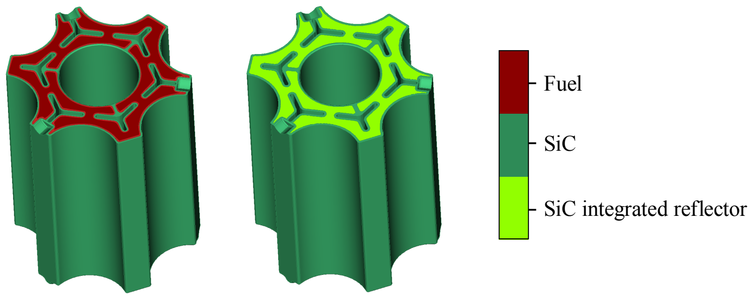


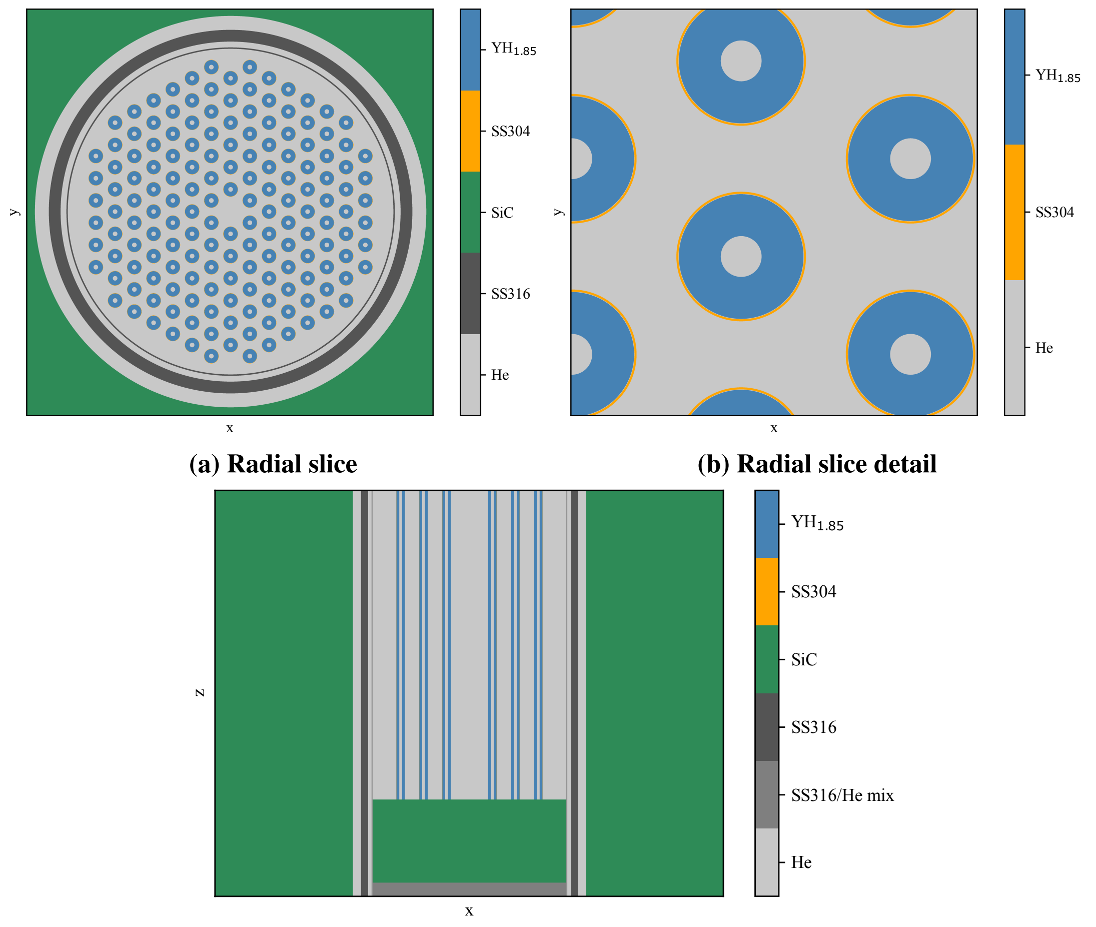
Figure 3: CAD fuel and reflector elements, which differ only by material assignments.

3. Transformational Challenge Reactor Demonstration Problem

The TCR program [7] explored how advancements in additive manufacturing enable complex fuel element designs. A core design was proposed [19], as shown in Fig. 2, consisting of cog-shaped fuel elements as shown in Fig. 3. These fuel elements consist of additively-manufactured SiC shells loaded with 19.5% enriched UN tristructural isotropic (TRISO) particles in an SiC matrix. Additional integrated reflector elements consist of the same SiC shell packed only with SiC. These fuel and reflector elements have wishbone-shaped cooling channels that cannot be analytically represented by quadratic surfaces and can therefore only be approximated in CSG. An assembly consists of eight fuel elements and three integrated reflector elements stacked axially with a $YH_{1.85}$ moderator rod through the center. The full core consists of 54 assemblies arranged in a hexagonal grid, for a total of 594 fuel/reflector elements. Additional "free" moderator rods occupy the spaces formed between adjacent assemblies.

3.1. Layered Geometry Construction

To construct a layered geometry model of TCR, a Shift CSG base layer was first created as shown in Fig. 4. This model does not contain a control shroud and is truncated below the upper axial reflector, since neutrons that make it past the integrated reflector elements above fuel elements are not expected to return to the core. Next, a layered geometry object was created from each of the CAD models in Fig. 3. The void material around the models was assigned to be transparent. This allows the base layer moderator rods to appear in the center of each element, and within the spaces between adjacent elements.



(a) Radial slice **(b) Radial slice detail**

(c) Axial slice

Figure 4: Shift CSG model of the TCR moderator rods and ex-core.

As mentioned in Section 2, only axis-aligned bounding boxes are currently supported when creating layered geometry objects, and these bounding boxes must not overlap for all objects within a single layer. As a result, multiple layers were created in order to place fuel/reflector elements into a hexagonal grid. The layering scheme is shown in Fig. 5. This figure shows that each row of assemblies is assigned to its own layer (denoted by color), since the bounding boxes of assemblies in each row do not overlap. TCR has nine rows of assemblies, thus nine layers are placed atop the base layer to form the full layered geometry, shown in Fig. 6. These images, as well as those in Fig. 4, were created with the Omnibus ray tracer [18], which uses Shift's tracking routines directly.

The bottom of each fuel element CAD model contains three hemispherical alignment features that fit into three arch-shaped dimples on the top of each element (as seen in Fig. 3). For the purpose of this demonstration, these alignment features were truncated via the object bounding box to simplify the layering scheme.

3.2. Transport Results

A Shift eigenvalue simulation was performed using the layered geometry model in Fig. 6. To mimic cold zero power conditions, a temperature of 300 K was used throughout the model. Eigenvalue iteration was performed with 25 inactive cycles and 25 active cycles, with 5×10^6 histories per cycle. Total flux and

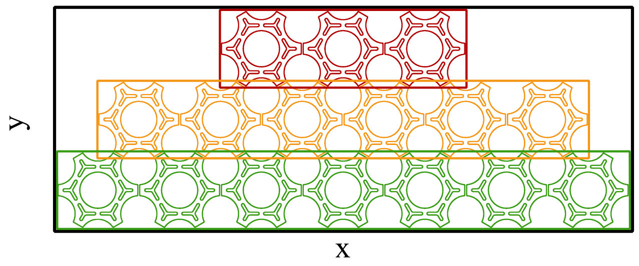
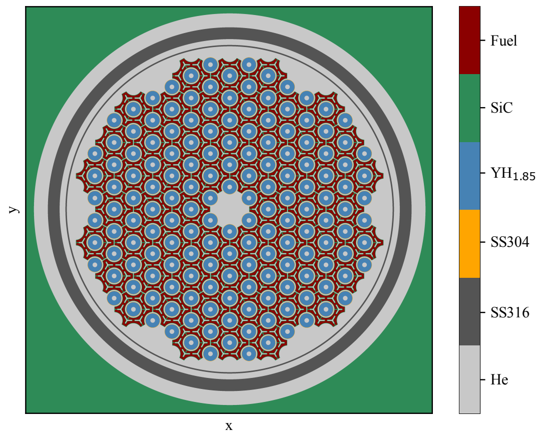
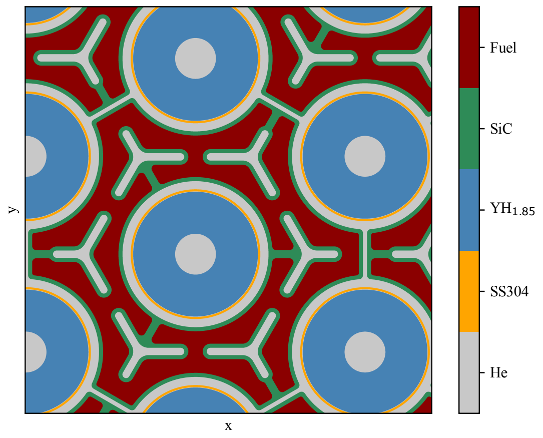


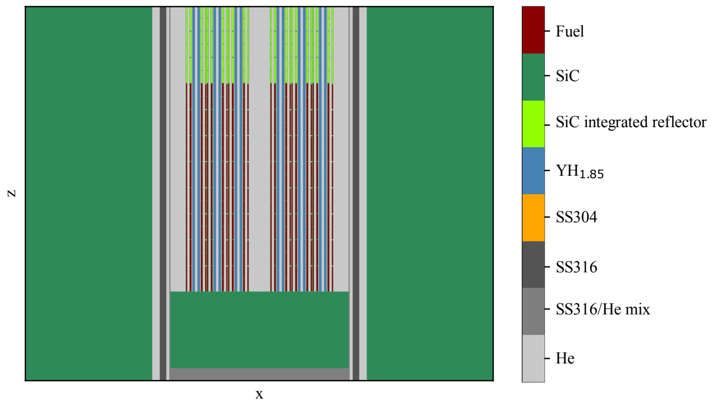
Figure 5: Layering scheme for the first three rows of assemblies. Colors denote layers.



(a) Radial slice (radial reflector truncated)

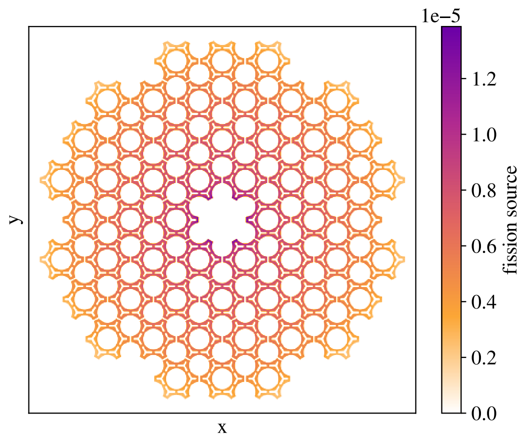


(b) Radial slice detail

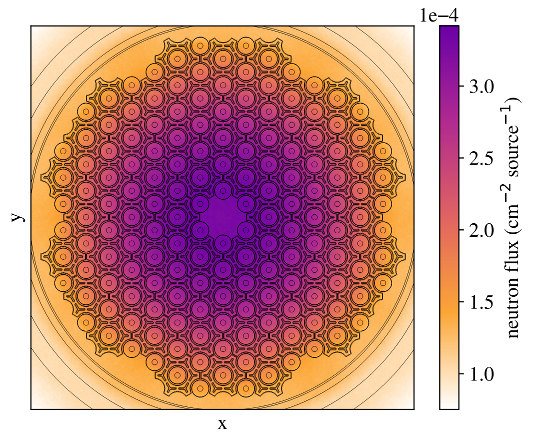


(c) Axial slice

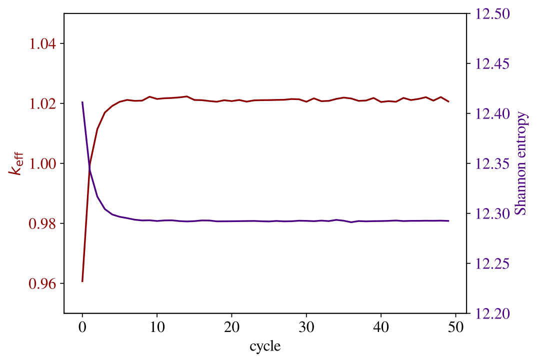
Figure 6: Layered geometry comprised of the CAD models in Fig. 3 and the CSG model in Fig. 4.



(a) Fission source distribution



(b) Flux distribution



(c) $\mathbf{k_{eff}}$ and Shannon entropy

Figure 7: Transport results for the layered geometry shown in Fig. 6.

fission source results were tallied on a $420 \times 420 \times 1$ radial mesh, with a z width extending ± 2 cm about the axial midplane of the fuel elements. An additional $20 \times 20 \times 20$ mesh over the fuel elements was used to assess the Shannon entropy. The simulation was run on 20 nodes with 48 processes per node on the Apollo compute cluster at Oak Ridge National Laboratory, which has two Intel® Xeon® Gold 5118 CPUs per node.

Transport results are shown in Fig. 7. The fission source and flux distributions match expectations, and the k_{eff} and Shannon entropy results show that the simulation achieved a degree of convergence that is acceptable for this demonstration, with a final k_{eff} of 1.0212 ± 0.0005. Transport required 15.7 h of wall-clock time. The CAD models were found to be the major source of lost particles, with a lost particle rate of 6.09×10^{-5} per source particle from DAGMC geometry errors. Particles were also lost via the layered geometry tracking process itself, accounting for an additional lost particle rate of 3.07×10^{-6} per source particle, about $20\times$ less than particles lost from DAGMC errors. In addition to lost particles, erroneous particles were observed to cause streaks of small but non-zero fission source density through regions not containing fissionable material, at an estimated rate of $\sim 4 \times 10^{-6}$ per source particle. It is hypothesized that many of these erroneous particles eventually become lost, and are therefore already accounted for in the reported lost particle rate, but more analysis is needed to confirm this. In Shift, source particles are re-sampled if they are born in non-fissionable material, so these rare erroneous particles only slightly bias the fission source distribution. These tracking errors will be mitigated in future work.

4. CONCLUSIONS

The layered geometry feature implemented within Shift provides a versatile mechanism for combining CAD and CSG models. This feature was demonstrated by creating a high-fidelity model of TCR in which CAD fuel elements with transparent centers are overlaid upon a CSG model of the TCR core. Transport results are consistent with expectations, with a converged k_{eff} of 1.0212 ± 0.0005. Future work will involve increasing the tracking robustness, expanding this capability to other geometry types supported by Shift, and performance optimization.

ACKNOWLEDGMENTS

This work was supported by the Laboratory Directed Research and Development (LDRD) program of Oak Ridge National Laboratory, which is managed and operated by UT-Battelle LLC for the US Department of Energy (DOE) under contract no. DEAC05-00OR22725.

REFERENCES

[1] T. M. Pandya, S. R. Johnson, T. M. Evans, G. G. Davidson, S. P. Hamilton, and A. T. Godfrey. "Implementation, capabilities, and benchmarking of Shift, a massively parallel Monte Carlo radiation transport code." *Journal of Computational Physics*, **volume 308**, pp. 239–272 (2016).

[2] P. Romano and B. Forget. "The OpenMC Monte Carlo particle transport code." *Annals of Nuclear Energy*, **volume 51**, pp. 274–281 (2013).

[3] D. Pelowitz. "MCNP6 User's Manual Version 1.0." Technical Report LA-CFP-13-00634 Rev 0, Los Alamos National Laboratory (2013).

[4] T. J. Tautges, P. P. H. Wilson, J. Kraftcheck, B. F. Smith, and 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* (2009).

[5] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho. "The Serpent Monte Carlo code: Status, development and applications in 2013." *Annals of Nuclear Energy*, **volume 82**, pp. 142–150 (2015).

[6] B. Smith. *Robust tracking and advanced geometry for Monte Carlo radiation transport*. Ph.D. thesis, University of Wisconsin (2011).

[7] ORNL. "Transformational Challenge Reactor Program." https://tcr.ornl.gov/ (2022).

[8] D. Grosse, U. Fischer, K. Kondo, D. Leichtle, P. Pereslavtsev, and A. Serikov. "Status of the McCad geometry conversion tool and related visualization capabilities for 3D fusion neutronics calculations." *Fusion Engineering and Design*, **volume 88**(9), pp. 2210–2214 (2013).

[9] E. E. Davidson, G. Radulescu, K. Smith, J. Yang, S. Wilson, and B. R. Betzler. "Reactor cell neutron dose for the molten salt breeder reactor conceptual design." *Nuclear Engineering and Design*, **volume 383**, p. 111381 (2021).

[10] J. Yang, S. C. Wilson, S. W. Mosher, and G. Radulescu. "Integration of the Full Tokamak Reference Model with the Complex Model for ITER Neutronic Analysis." *Fusion Science and Technology*, **volume 74**(4), pp. 277–287 (2018).

[11] P. P. Wilson. "MCNP2CAD." https://github.com/svalinn/mcnp2cad (2013).

[12] R. L. Martz. "The MCNP6 Book On Unstructured Mesh Geometry: User's Guide For MCNP 6.2." Technical Report LA-UR-17-22442, Los Alamos National Laboratory (2017).

[13] P. Shriwise. "DAGMC CAD-based geometry as universes." https://github.com/openmc-dev/openmc/pull/1825 (2021).

[14] J. Leppänen. "CAD-based geometry type in Serpent 2–Application in fusion neutronics." *Joint International Conference on Mathematics and Computation, Supercomputing in Nuclear Applications, and the Monte Carlo Method*, pp. 19–23 (2015).

[15] A. Jambrina. https://ttuki.vtt.fi/serpent/viewtopic.php?f=3&t=4015#p13620 (2022).

[16] A. Talamo, Y. Gohar, and J. Leppänen. "SERPENT validation and optimization with mesh adaptive search on stereolithography geometry models." *Annals of Nuclear Energy*, **volume 115**, pp. 619–632 (2018).

[17] B. Rearden and M. A. Jessee, Eds. "SCALE Code System, Version 6.2." Technical Report ORNL/TM-2005/39, Oak Ridge National Laboratory, Oak Ridge, TN (2016).

[18] S. R. Johnson, T. M. Evans, G. G. Davidson, S. P. Hamilton, T. M. Pandya, K. E. Royston, and E. D. Biondo. "Omnibus User Manual." Technical Report ORNL/TM-2018/1073, Oak Ridge National Laboratory, Oak Ridge, TN (2020).

[19] B. Betzler et al. "Transformational Challenge Reactor Preliminary Core Design Report." Technical Report ORNL/TM-2020/1718, Oak Ridge National Laboratory (2020).