Resolution of Self-Intersection Issue in Monte Carlo Simulations Employing Graphics Ray Tracing Technology

Sung Joon Kwon, Jaeuk Im, Han Gyu Joo

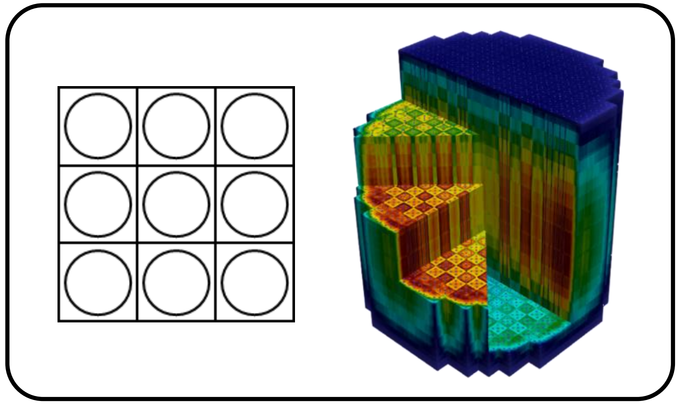
Department of Nuclear Engineering

Seoul National University

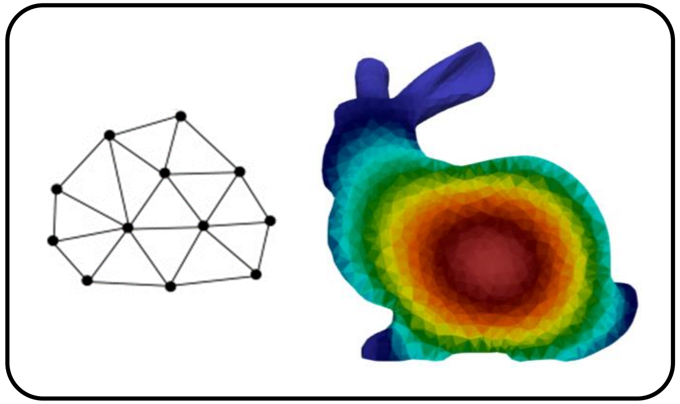
May 18th, 2023

- **Power Reactor Analysis using GPU-based Monte Carlo Algorithm (PRAGMA)**
  - **Funded by KHNP through K-CLOUD project.**
  - **Language: CUDA C++**
  - **Objectives of the PRAGMA code**
    - Apply dedicated optimizations for PWR analysis.
    - Enable efficient simulation in feasible time scale on small cluster equipped with consumer-grade GPUs.
    - Supports general unstructured mesh geometry treatment powered by graphics ray tracing technology.

PWR Lattice Geometry

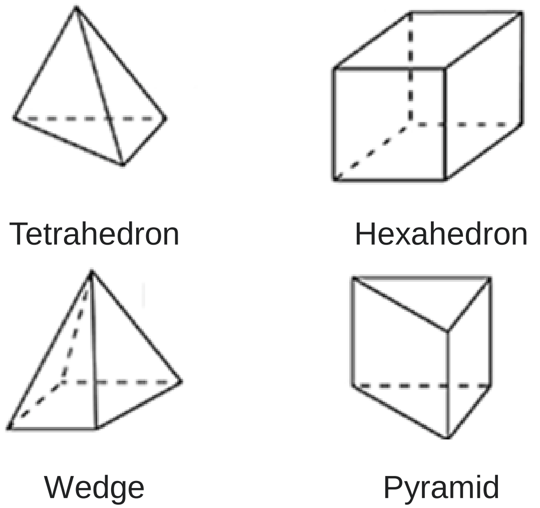


Unstructured Mesh Geometry

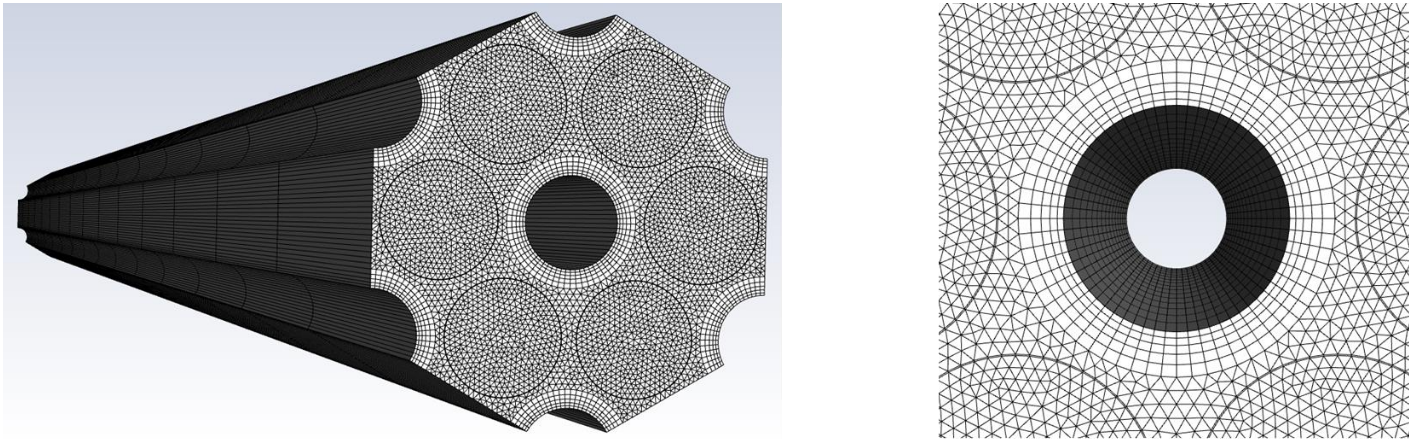


- **Unstructured Mesh Geometry Representation in PRAGMA**
  - **An irregular geometry is represented based on CAD mesh-based geometry model in PRAGMA.**
    - A structure is modeled using only four types of basic mesh cells.
    - PRAGMA reconstructs an unstructured geometry based on a mesh file generated by ANSYS or Cubit.
  - **For efficient modeling of a curve with meshes, a volume correction method was employed to preserve calculation accuracy using a few meshes.**

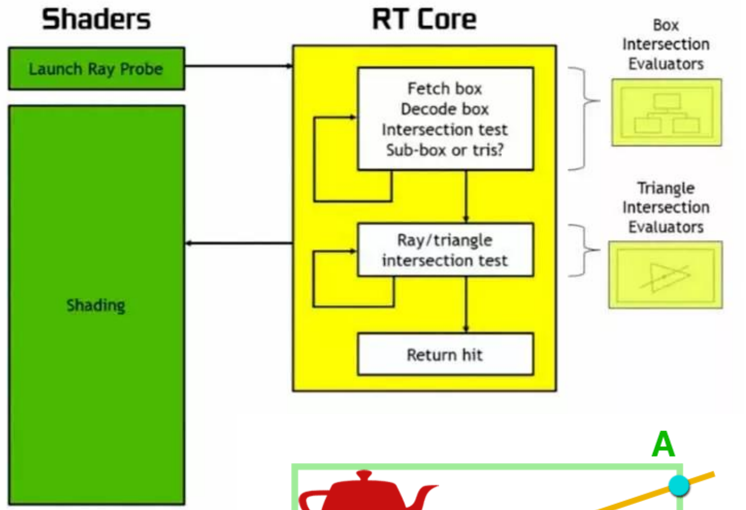
Basic Cells

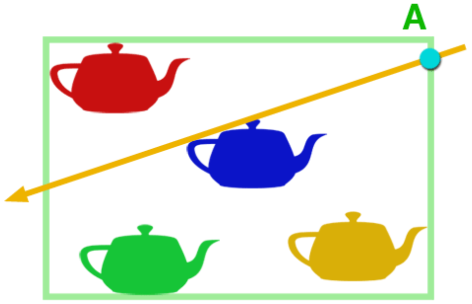


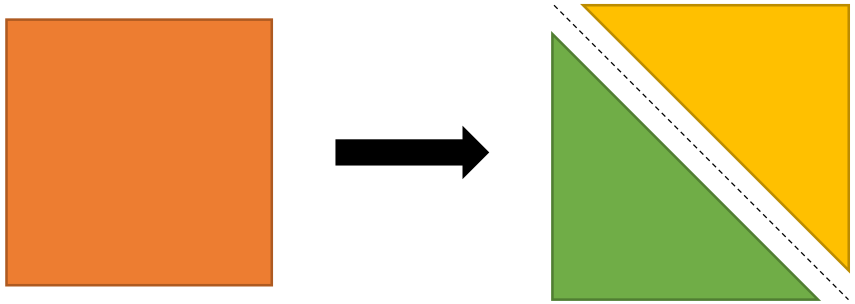
Example of Mesh-based Geometry Modeling



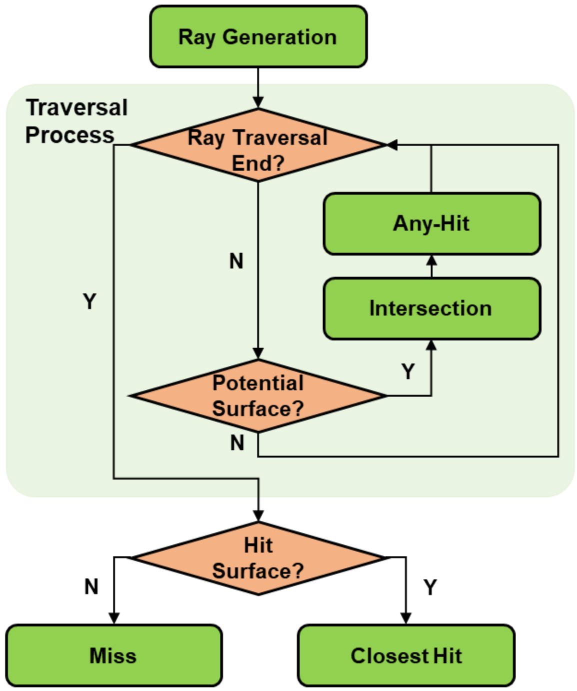
- **Application of OptiX Ray Tracing in Mesh-based Geometry**
  - **PRAGMA adopts OptiX for neutron tracking in a mesh-based geometry.**
    - OptiX is a CUDA-based ray tracing API optimized for NVIDIA GPUs.
  - **It provides a programmable ray tracing pipeline allowing a user to create a custom ray tracing kernel.**
    - Bounding Volume Hierarchy (BVH) traversal is automated by the library, while other programs are user-supplied.
  - **For triangles, an optimized built-in ray tracing algorithm is provided.**
    - The built-in program leverages the hardware acceleration of NVIDIA GPUs (RT cores).
    - Quadrilaterals are split into triangles to exploit the optimized built-in capabilities of the ray tracing engine.

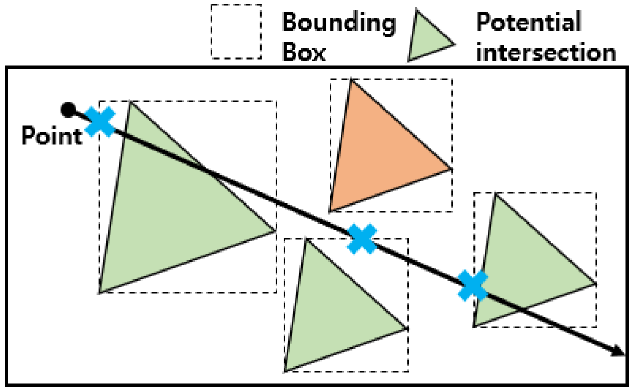






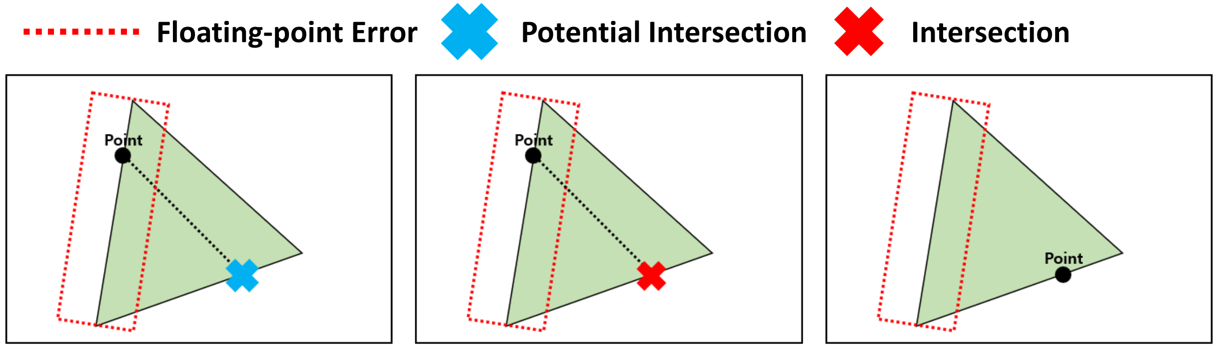
- **OptiX Ray Tracing Pipeline**
  - **The ray tracing pipeline mainly consists of a ray generation, an intersection program, shading, and a miss program.**
    - A virtual ray invoked from the ray generation program traverses scene geometry in the pipeline and finds the intersections with primitives.
  - **When a ray is cast from a point, potential intersections are determined based on bounding boxes of primitives.**
  - **For all potential surfaces, an intersection and Any-Hit program are invoked during traversal.**
    - The Any-Hit program is called when a traced ray finds a potentially-closest intersection point for graphics shadow computation.
    - However, in the MC application, the Any-Hit program is generally redundant since only the distance to the nearest surface from the point is utilized in the simulation process.
  - **Based on the distance calculated by the intersection program, the closest surface is determined among the potential intersections.**



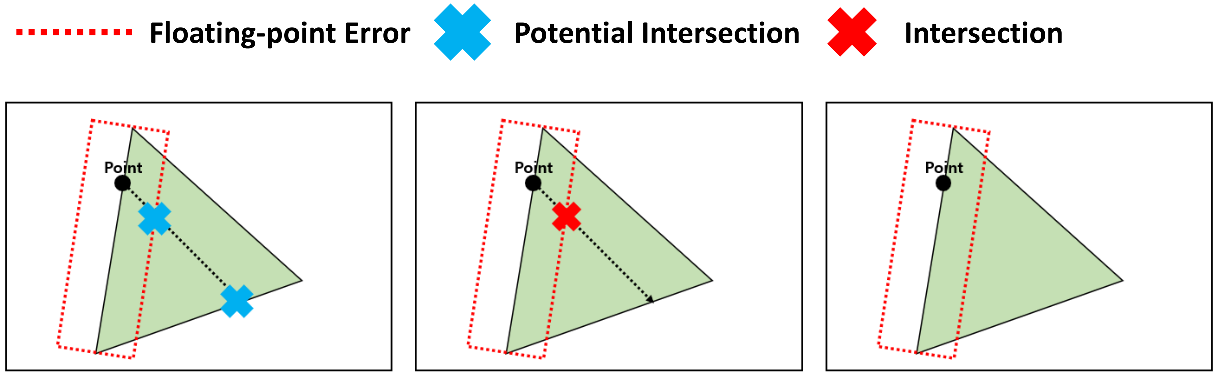


- **Self-Intersection Issue in MC simulation**
  - **A self-intersection is a phenomenon that the intersection of the same surface is repeatedly detected for a new ray due to floating-point arithmetic.**
    - Ray tracing cannot proceed when a self-intersection occurs.
    - In graphics, most self-intersections can be resolved by utilizing a threshold parameter to neglect all intersections with distance parameters smaller than the value.
  - **In an MC simulation, a neutron is trapped on a certain surface when a self-intersection occurs.**
    - It may degrade the performance and accuracy of an MC simulation.
    - Adopting a threshold parameter cannot be a remedy in an MC simulation since the required accuracy in the neutronics is too microscopic.

< Normal Ray Tracing >

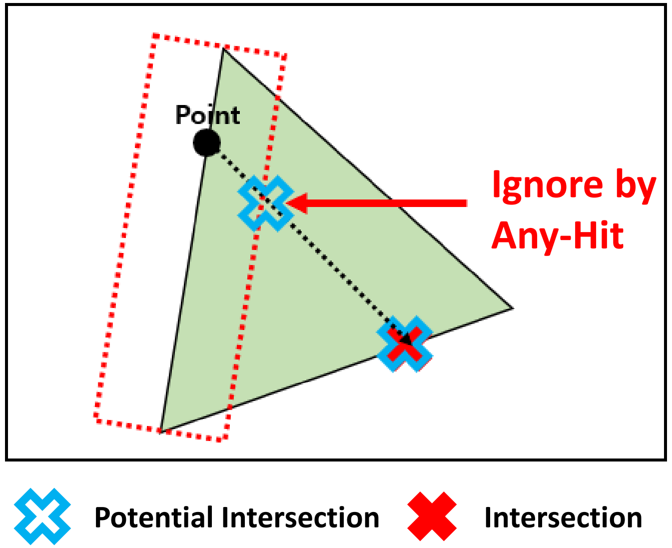


< Ray Tracing with Self-Intersection >



- **Existing Solutions to Prevent Self-Intersections**
  - **A threshold parameter called scene epsilon neglects intersections with a distance smaller than a preset value.**
    - In an MC simulation, the required physical accuracy may decrease by adopting the scene epsilon parameter.
  - **Another solution is applying new dynamic epsilon called track epsilon forced to deviate neutron from the surface.**
    - However, this solution must require an additional ray traversal process increasing computing time.

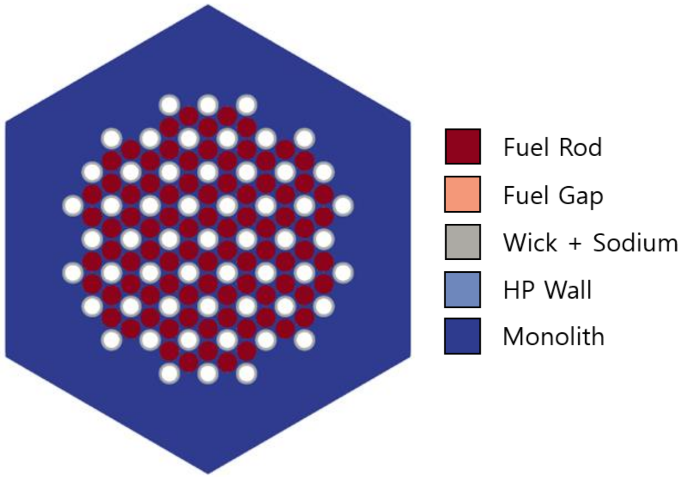
- **Solution to Prevent Self-Intersections Adopting Any-Hit Approach**
  - **The idea is to ignore self-intersections using the Any-Hit program during traversal.**
    - The Any-Hit program can interfere during the traversal process and explicitly ignore several cases of self-intersections.
  - **It is expected that the self-intersections can be eliminated preserving the physical accuracy by adopting the Any-Hit approach.**
    - It should be verified how much the performance decreases by additional calculations during traversal.



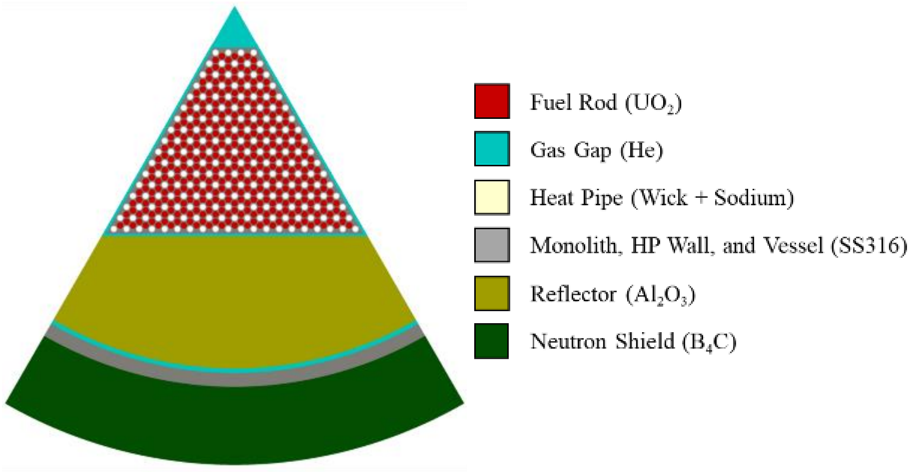
- **Problem Descriptions**
  - **The Minicore designed by ANL and the one-sixth symmetrical MegaPower 3D core was adopted in this research.**
    - The control drums in MegaPower core were not modeled.
    - The vapor region of each heat pipe is replaced by a void pipe with a reflective boundary.
  - **The tracking method without the Any-Hit was adopted as a standard case to confirm the effect of self-intersections.**

Problem	Minicore	MegaPower
# of Cells	904,200	1,897,530
# of Inactive Cycles	25	50
# of Active Cycles	100	
# of Neutrons / Cycle	4,000,000	
Libraries	ENDF/B-VII (900K / 1000K)	ENDF/B-VII (900K / 1000K / 1100K)

< Configuration of Minicore >



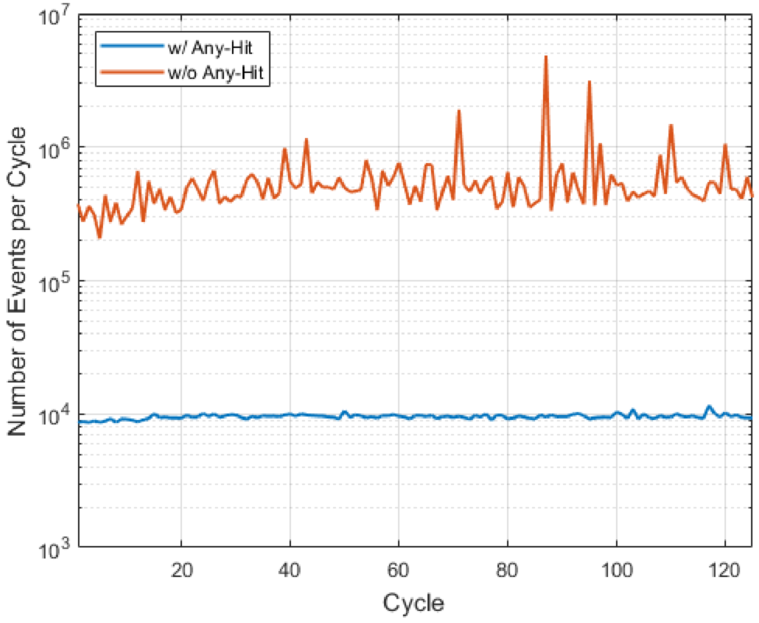
< Configuration of MegaPower >



Effect of Any-Hit Program on Self-Intersections

- **Calculation Results of Minicore Problem**
  - **It is observed that a neutron event behavior of the standard case shows much higher value and fluctuation.**
    - For a standard case, the average number of events appeared to be about 60 times larger than that of the Any-Hit case.
    - Some neutrons stuck on the surfaces by self-intersections lead to an abnormal population tail effect.
  - **It was confirmed that the abnormal population tail effect is resolved by adopting the Any-Hit approach.**

< Number of Events at Each Cycle >

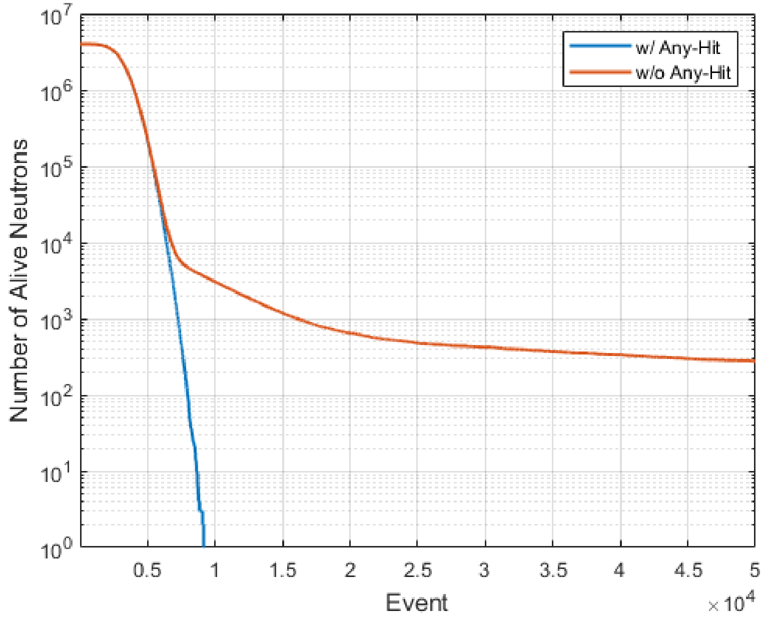


Scheme	w/ Any-Hit	w/o Any-Hit
Multiplication Factor	1.04684(2)	1.04682 (2)
Average Number of Events per Cycle	9,546	572,847

Effect of Any-Hit Program on Self-Intersections

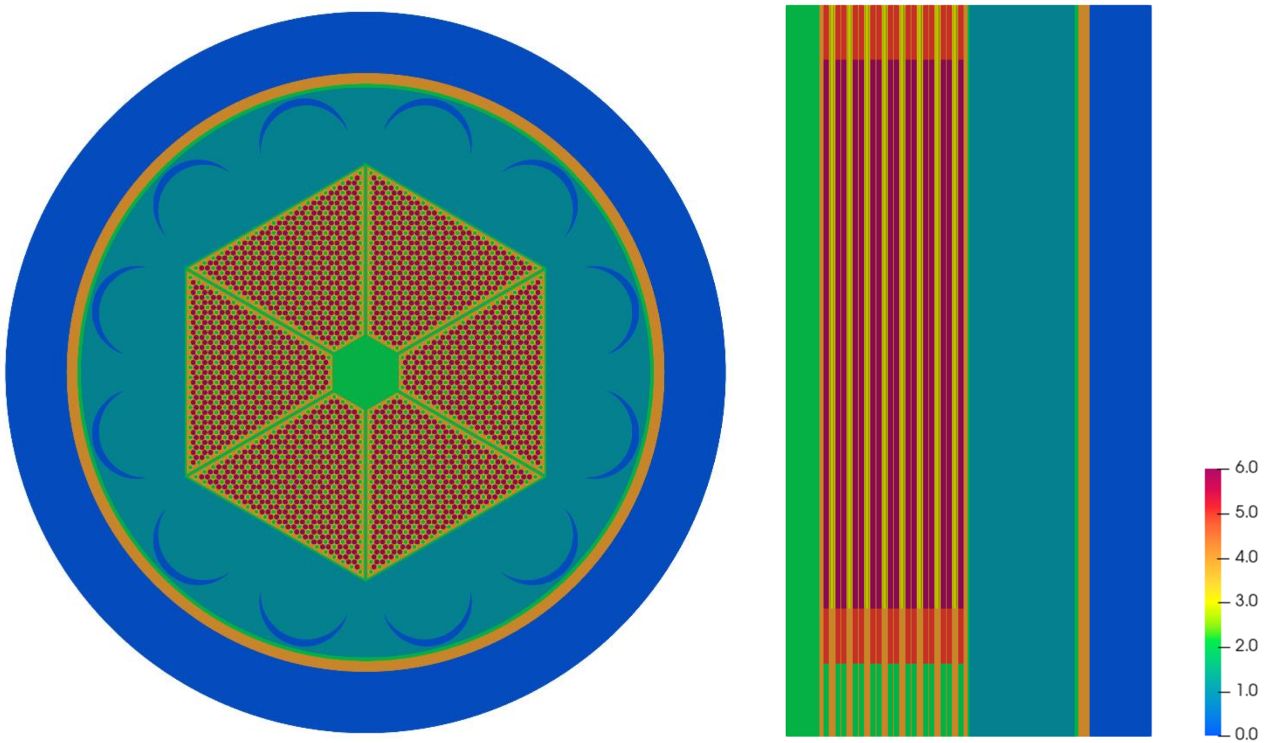
- Calculation Results of MegaPower Problem
  - **The simulation without the Any-Hit program did not proceed due to alive neutrons.**
    - Some particles are stuck on surfaces by self-intersections such that they did not disappear even after several hundred thousand events.
    - The number of alive neutrons rapidly diminished when adopted the Any-Hit program.
  - **It was confirmed that a non-negligible number of particles was trapped on the surface without the Any-Hit approach.**
    - The multiplication factor changes significantly when the alive neutrons are ignored.

< Number of Alive Neutrons at Each Event >



Scheme	w/ Any-Hit	w/o Any-Hit
Multiplication Factor	1.13070 (3)	1.12538 (3)
Average Number of Events per Cycle	7,067	10,000
Average Tracking Time of Cycle [s]	29.63	29.70

- **MegaPower 3D Full Core Problem**
  - **Problem configuration (Material-wise)**

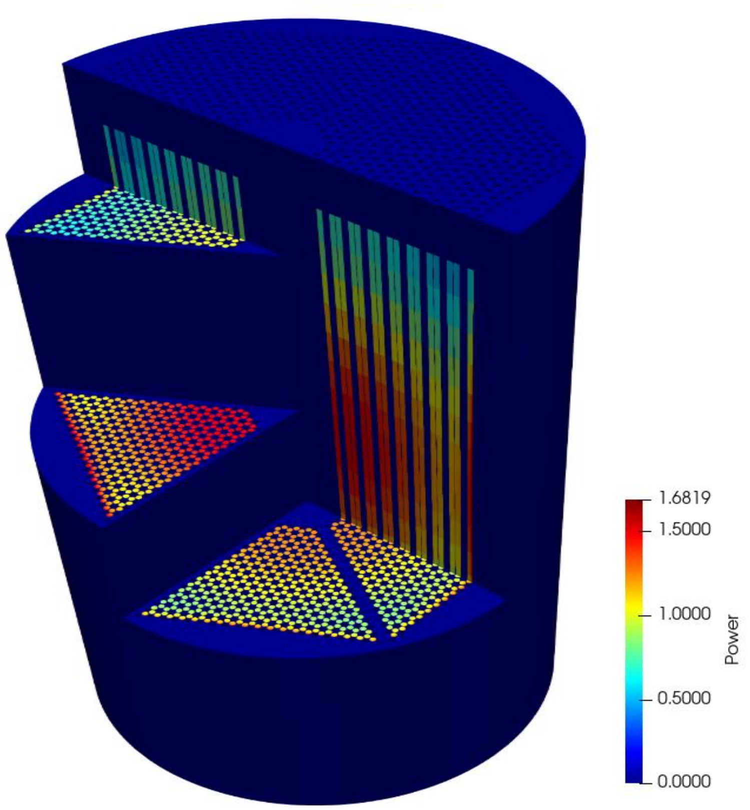


  - **Calculation conditions**

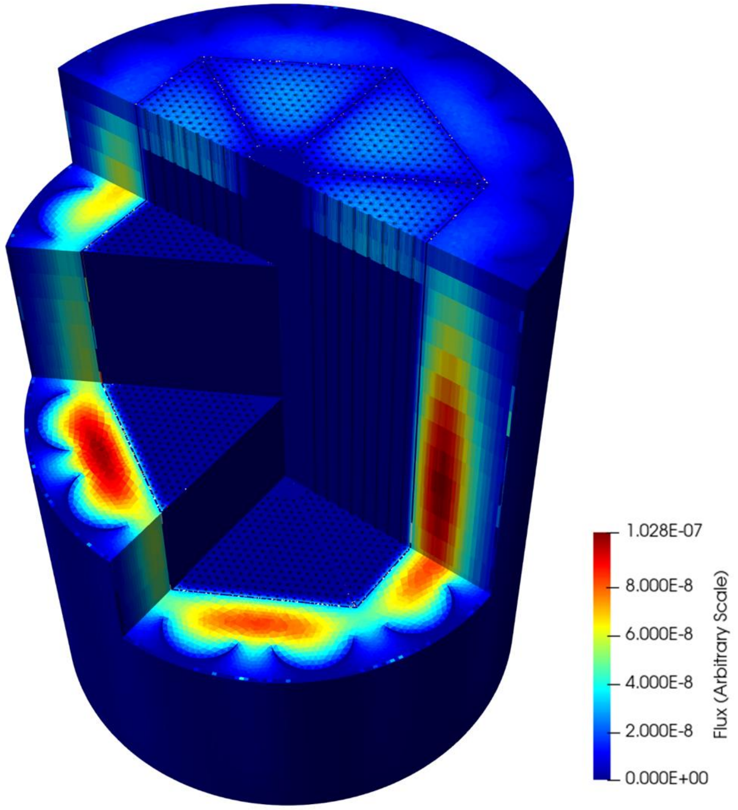
Parameter	Value
Problem	LANL MegaPower 3D Full Core
# of GPUs	24 (NVIDIA RTX A5000)
# of Particles	200,000,000 (Total 11.5 Billion)
# of Cycles	25 (Inactive) / 50 (Active)
Ramp-up	On (Mode: Exponential, Factor: 20)
Core Power	5 MW_{th}
Feedback	Off
Delta Tracking	On (zone-wise)
Grid Hashing	On (Hash size: 50)
Libraries (K)	900 / 1000 / 1100
# of Cells	24,054,408
# of Traingles	133,594,200

- **Calculation Results**
  - **The total computing time: 35 minutes**

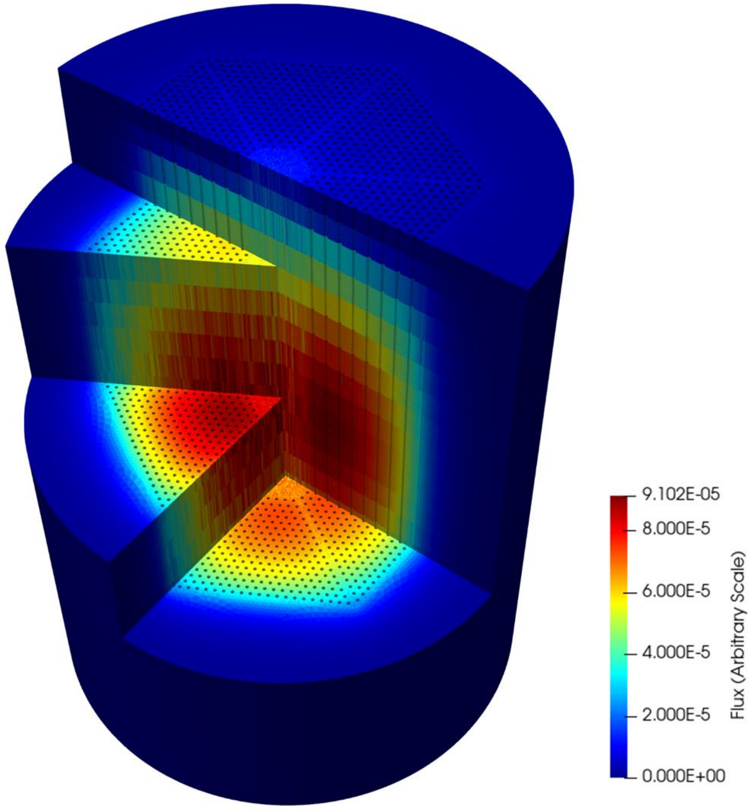
Power Distribution



Thermal Flux (< 0.1 eV)



Fast Flux (> 0.1 MeV)



Conclusion & Ongoing Works

- **Conclusion**
  - **The self-intersections in a MC simulation were resolved by using the Any-Hit approach.**
    - Any-Hit program, which was originally used for graphics shadow computation, was successfully augmented to the existing OptiX pipeline in PRAGMA.
    - It was confirmed that the self-intersections are resolved without degrading the performance, unlike other additional manipulation.
  - **It was demonstrated that the MegaPower 3D full core can be performed efficiently by PRAGMA.**

- **Ongoing Works**
  - **The unstructured geometry treatments of PRAGMA are being verified through various realistic problems.**
    - It should be verified whether the self-intersections are resolved totally through verifications with various problems.
  - **Some multiphysics analysis systems based on PRAGMA are being developed for an accurate advanced nuclear reactor simulation.**
    - Neutronics – thermo-mechanical – heat-pipe coupled multiphysics analysis system for the heat-pipe analysis was already established.