Transactions of the Korean Nuclear Society Sprint Meeting

Jeju, Korea, May 18-20, 2023

GPU-based Method of Characteristics with CMFD Acceleration in Unstructured Mesh Geometry

Kim Kyung Min, Han Gyu Lee, Han Gyu Joo

Department of Nuclear Engineering, Seoul National University

May 18th, 2023

{kmkim3182, hglee531, joohan}@snu.ac.kr

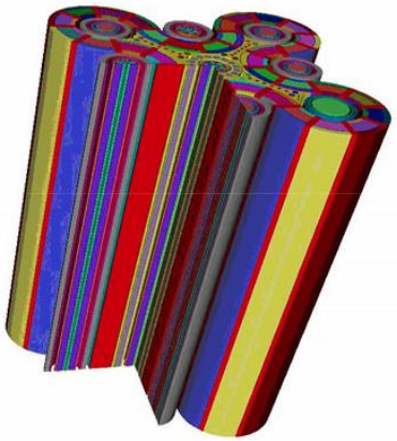
ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY
서 울 대 학 교 공 과 대 학

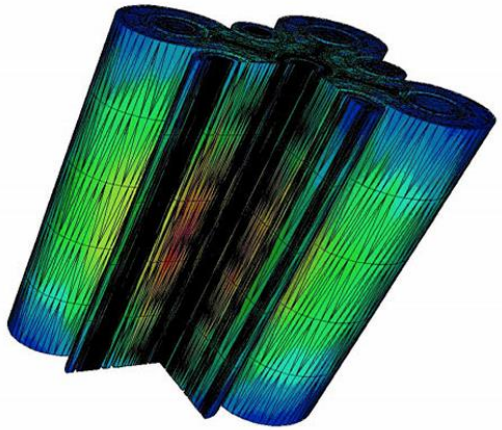
Contents

Statement of Research Purpose

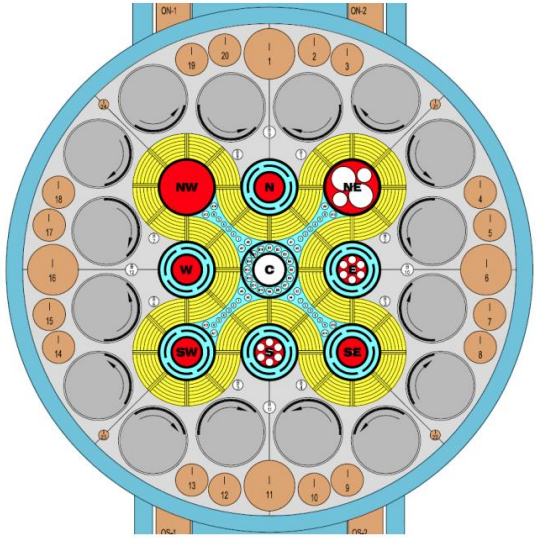
- **Emerging Advanced Reactors**
  - "Advanced Reactor Demonstration Program" led by the US DOE
  - "선진원자로 설계/검증을 위한 고신뢰도 시뮬레이션 기술 개발" led by KAERI
  - "Generation IV International Forum," cooperated worldwide

- **Difficulties in Analyzing Advanced Reactors via Deterministic Methods**
  - Irregular geometry of the advanced reactors
  - Hard to establish solvers more general for irregular geometry
  - ➢ **Preferring unstructured mesh geometry over constructive solid geometry**





INL's Advanced Test Reactor (ATR)



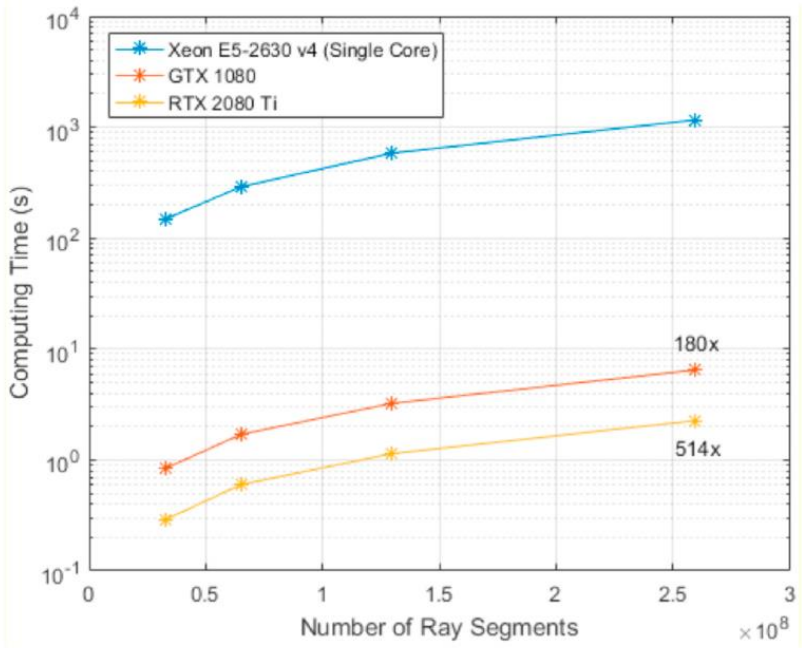
- **Deterministic Codes Operating over Unstructured Mesh Geometry**
  - PROTEUS-MOC by Argonne National Laboratory
  - Rattlesnake (S_N) by Idaho National Laboratory
  - ➢ **Griffin based on the MOOSE framework**

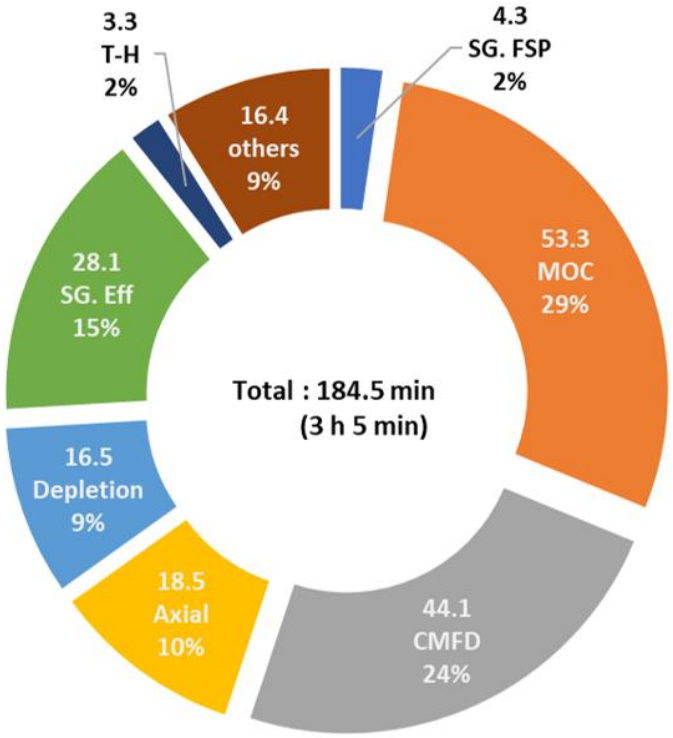
MOOSE

Argonne NATIONAL LABORATORY

INL Idaho National Laboratory

- **nTRACER Overcoming Runtime Overhead via GPUs as Accelerators**
  - A transport code based on the planar method of characteristics (MOC)
  - Employing the coarse mesh finite difference (CMFD) as an acceleration method
  - ➢ Consumed **a few minutes** to solve a 3D whole-core problem.
  - ➢ **Takes full advantage of the parallel computing power of GPUs.**



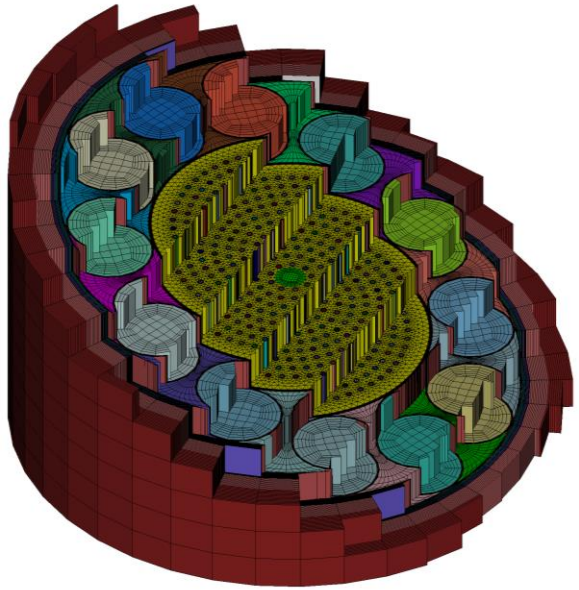


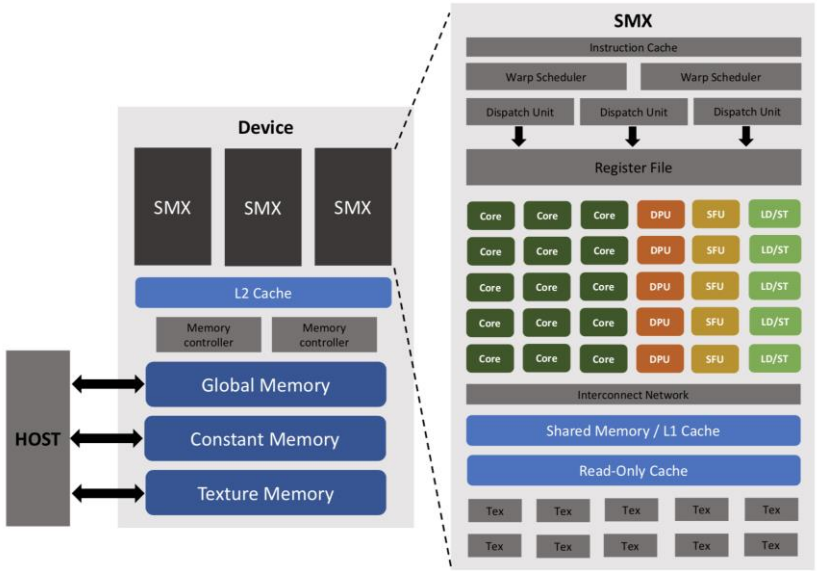
N. Choi *et al.*, "Practical Acceleration of Direct Whole-core Calculation Employing Graphics Processing Units"

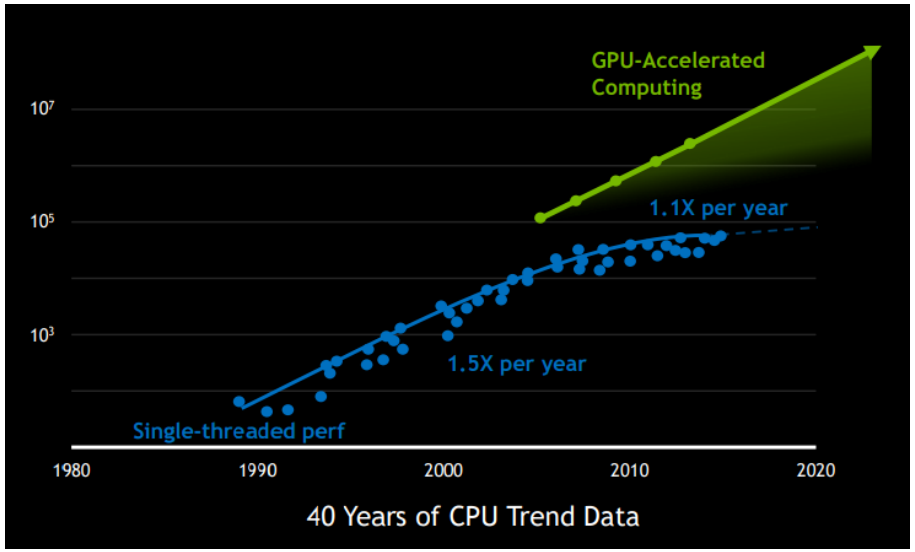
Statement of Research Purpose

- Development of a New Deterministic Code Capable of Flexible Geometry Treatment
- Employing GPUs as Accelerators to Achieve High Performance on Small Clusters
  - Establishment of an efficient data structure for navigating unstructured mesh
  - GPU-based MOC operating over unstructured mesh geometry
  - CMFD formulation compatible with the higher-order solver

➢ NuDEAL Neutronics using Deterministic Finite Element Algorithm





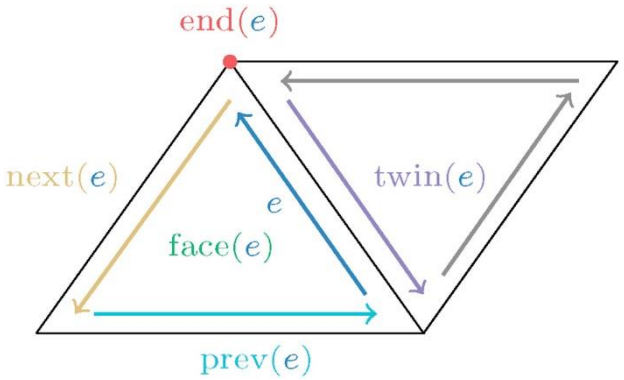


Methodologies and Algorithms

- Construction for MOC Calculations
- GPU-parallelized Ray Tracing
- CMFD Development

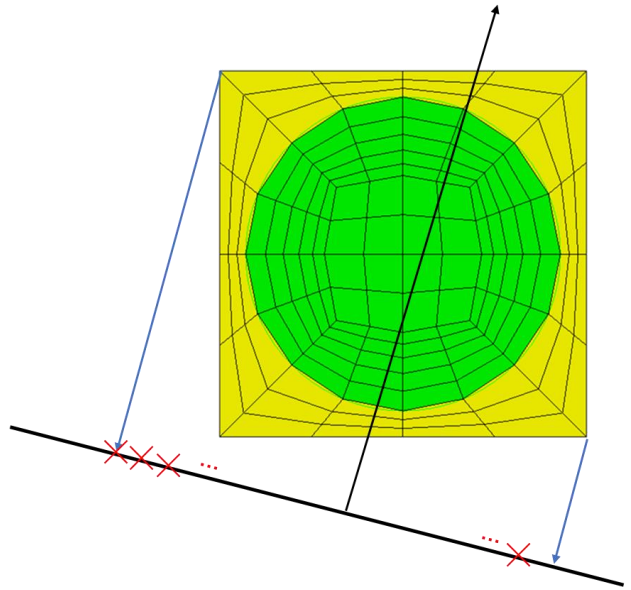
- **Array-based Half-edge Data Structure***
  - Every topological element is related to each corresponding half-edge.
  - A half-edge has most of the geometrical information.
  - ➢ **Efficient manipulation of a 2D manifold**

- **Back Projection Method for Configuring Ray Distribution**
  1. The vertices in a mesh are projected onto a line perpendicular to the ray direction.
  2. The rays are generated such that they start at the midpoint of the projected vertices.
  - ➢ Only boundary vertices are projected in general.



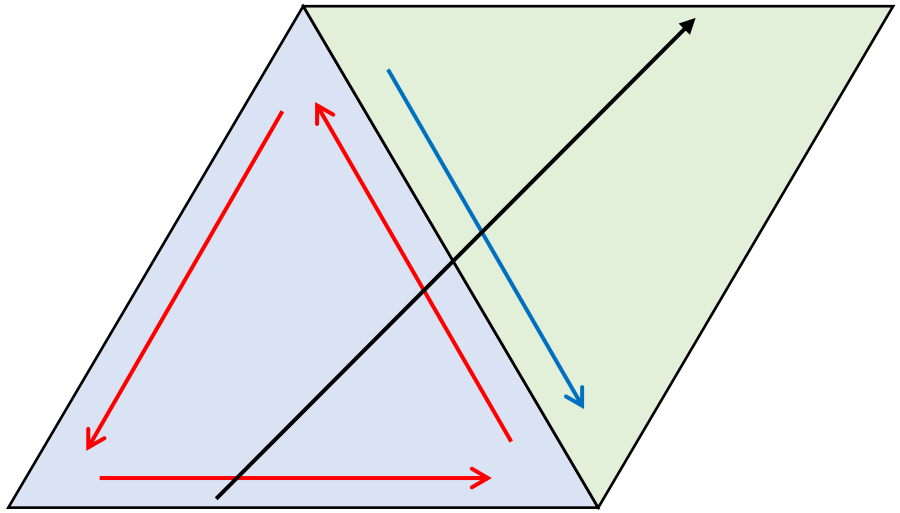
Half-edge Data Structure

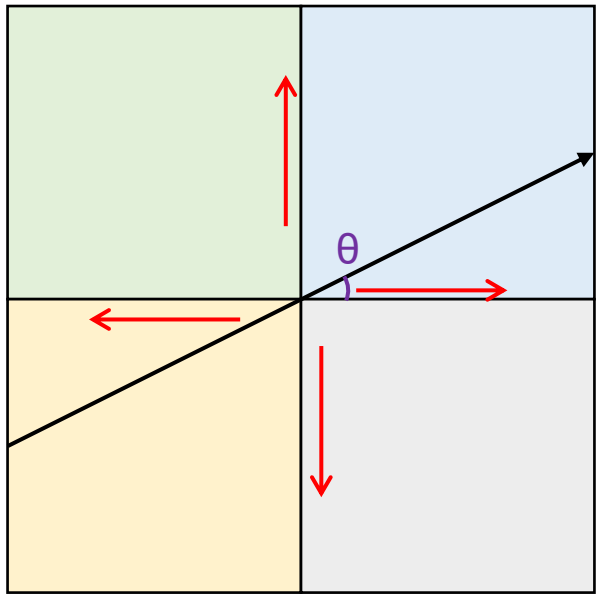
Element	Pointing to
Vertex	A half-edge
Edge	
Face	
Half-edge	Twin, next, vertex, edge, face



B. S. Bischoff *et al.*, “OpenMesh – A Generic and Efficient Polygon Mesh Data Structure”

- **Ray-mesh Intersection**
  - A corresponding face (flat source region) is attached to all the half-edges.
  - Traversal around a face yields an appropriate half-edge whose twin is the next face.
  - Traversal around a vertex handles exceptions due to the truncation error.





- **Ray Tracing Algorithm and Data Layout**

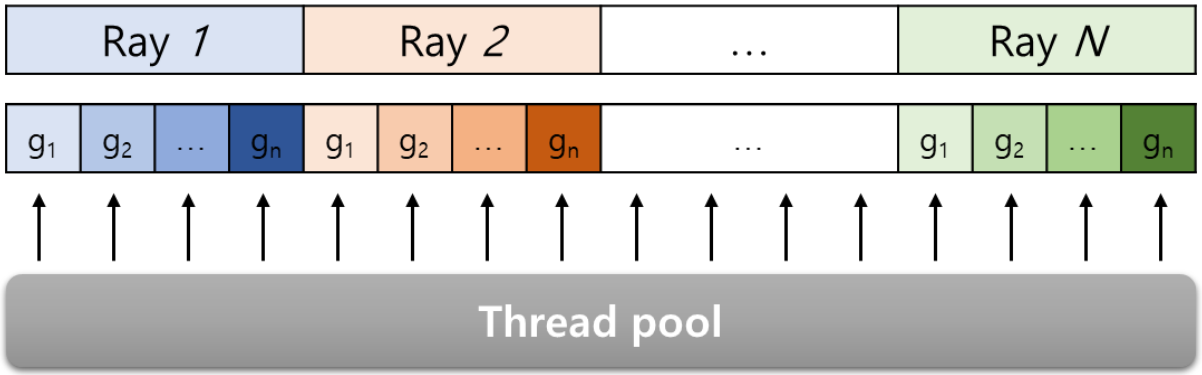
```
for each ray r parallel do
    for each group parallel do
        for each segment s ∈ r do
            for each polar angle do
                Save outgoing angular flux on cache
                Accumulate angular flux change on register
            end for
            Atomically accumulate FSR pseudo scalar flux
        end for
    end parallel for
end parallel for
```

$$\Delta\varphi \equiv \varphi_{in} - \varphi_{out} = \left(\varphi_{in} - \frac{q}{\Sigma_t}\right)\left(1 - e^{-\Sigma_t l}\right)$$

$$\tilde{\phi}_e = \sum_m \sum_{k \in e} \omega\left(\hat{\Omega}_m\right) \delta_k \Delta\varphi_k \sin\theta_m$$

e: element index

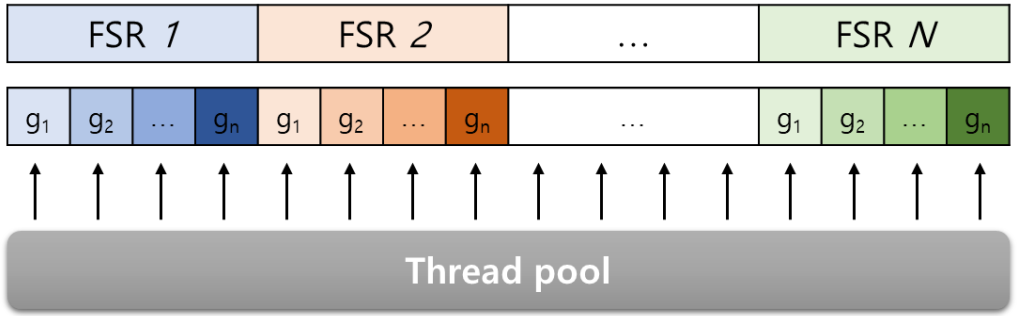
***k*: ray segment index**



Scalar Flux Reconstruction

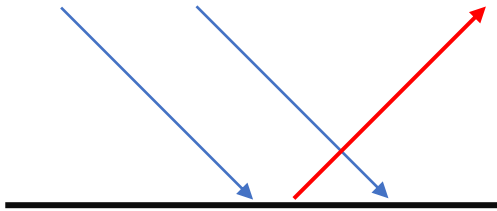
for each flat source region **parallel do**
  for each group **parallel do**
    Reconstruct scalar flux
  end parallel for
end parallel for

$$\phi_e = \frac{\tilde{\phi}_e}{\Sigma_t A_e} + \frac{q}{\Sigma_t}$$

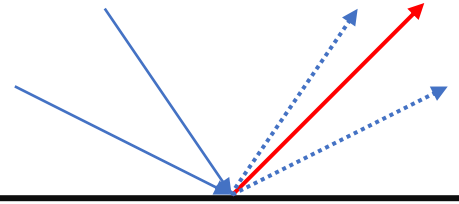


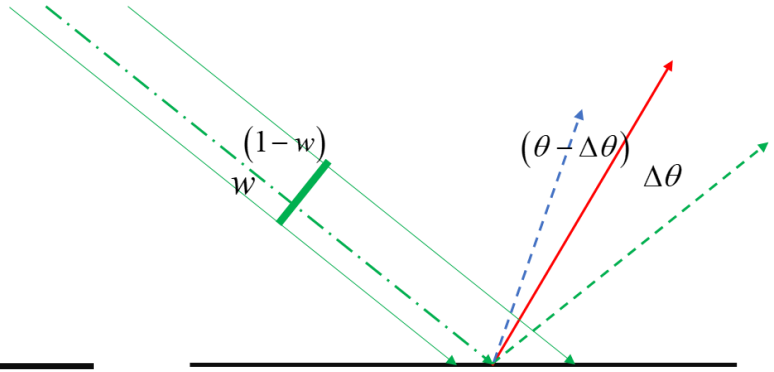
Reflective Boundary Condition Treatment

① Intersection mismatch



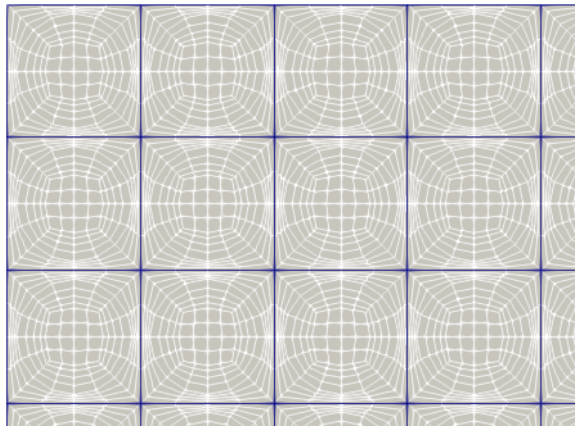
② Angle mismatch



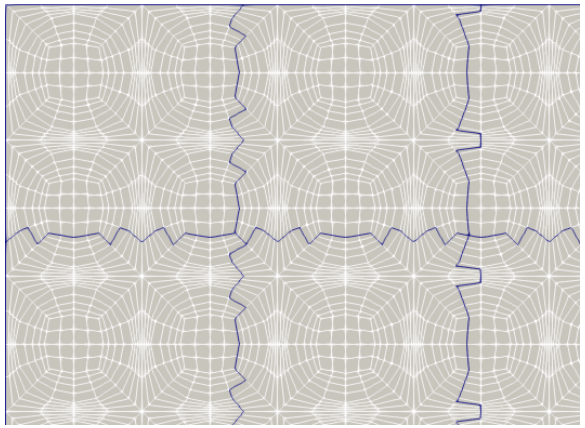


Incident angle 1
Incident angle 2
Reflected ray

- **Tailored Partitioning Algorithm***
  - An individual coarse mesh is required.
  - The algorithm partitions the fine mesh.
  - Compatible with:
    - A stencil grid, e.g., 2×2
    - A separate mesh file
  - Regular or irregular coarse meshes can be generated.



Regular grid



Irregular grid

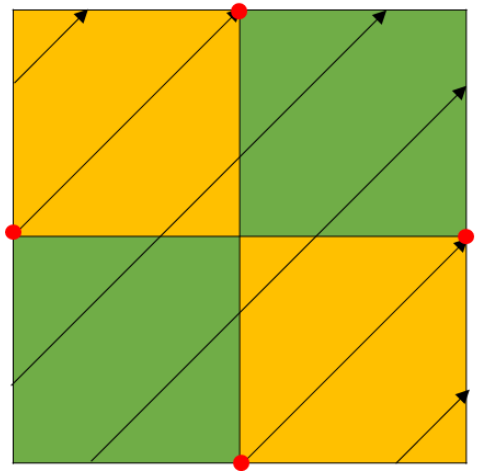
- **Pseudo Absorption Cross Section (PAXS)****

$$J_S^{\pm} = \sum_m \sum_{r \in S} \omega\left(\hat{\Omega}_m\right) \sin\theta_m \varphi_r$$

$$\sum_{S \in I} \left(J_S^+ - J_S^-\right) \Delta A_S + \Sigma_t^I \bar{\phi}_I V_I \neq \bar{q}_I V_I$$

$$\sum_{S \in I} \left(J_S^+ - J_S^-\right) \Delta A_S + \left(\Sigma_t^I + \Sigma_p^I\right) \bar{\phi}_I V_I = \bar{q}_I V_I$$

$$\Sigma_p^I \equiv \frac{\bar{q}_I V_I - \sum_{S \in I} J_S^{net} \Delta A_S}{\bar{\phi}_I V_I} - \Sigma_t^I$$



*A. Hsieh, “Development of Transient Analysis Capability of PROTEUS-MOC for Micro-reactor Applications”

**Y. S. Jung, W. S. Yang, “A Consistent CMFD Formulation for the Acceleration of Neutron Transport Calculations Based on the Finite Element Method”

ENGINEERING

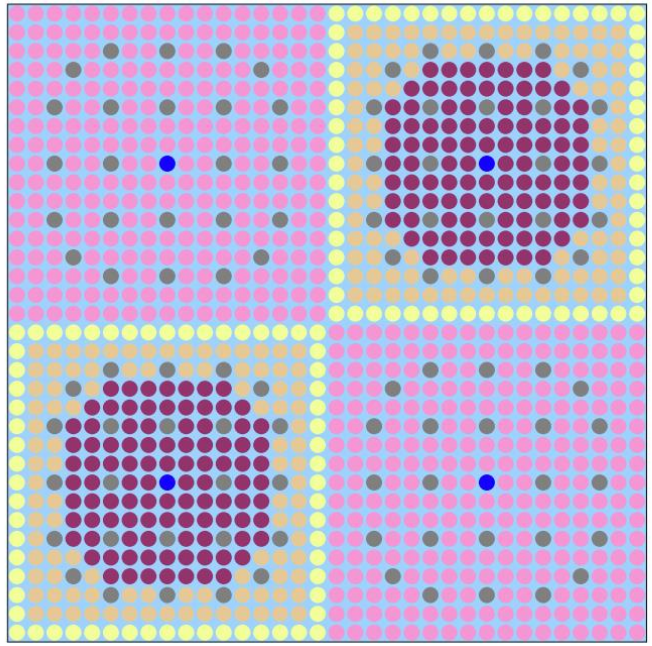
Results and Conclusion

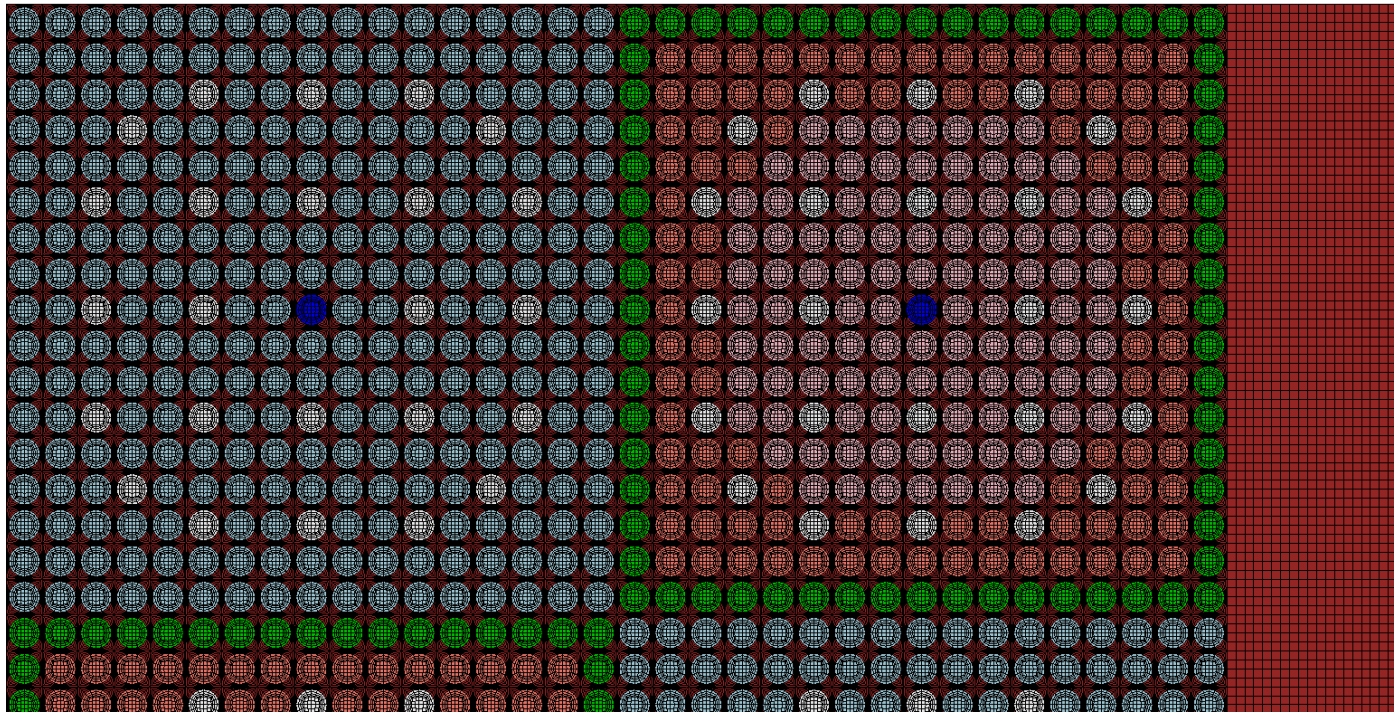
- Accuracy Examination
- Flexibility Demonstration
- Performance and Scalability
- Conclusion and Future Works

Accuracy Examination

- **C5G7 2D Problems**
  - 189,584 FSRs / 2,601 CMFD cells
- **Calculation Conditions**
  - 0.05 cm of ray spacing / 16 azimuthal angles / 4 polar angles
  - ➢ **23,891,009 ray segments**
  - Convergence criteria: 10^{-6} for fission source / 10^{-7} for eigenvalue

GeForce GTX 1080
8,228 GFLOPS FP32
257 GFLOPS FP64
8GB (352GB/s GDDR5X)



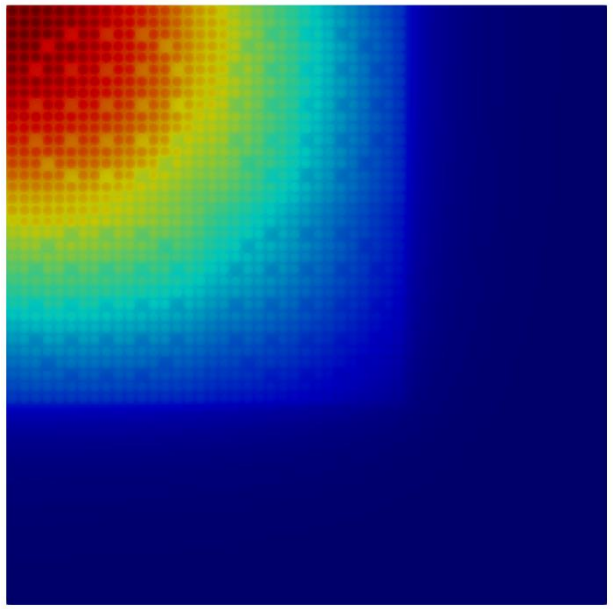


- **Eigenvalue Comparison**

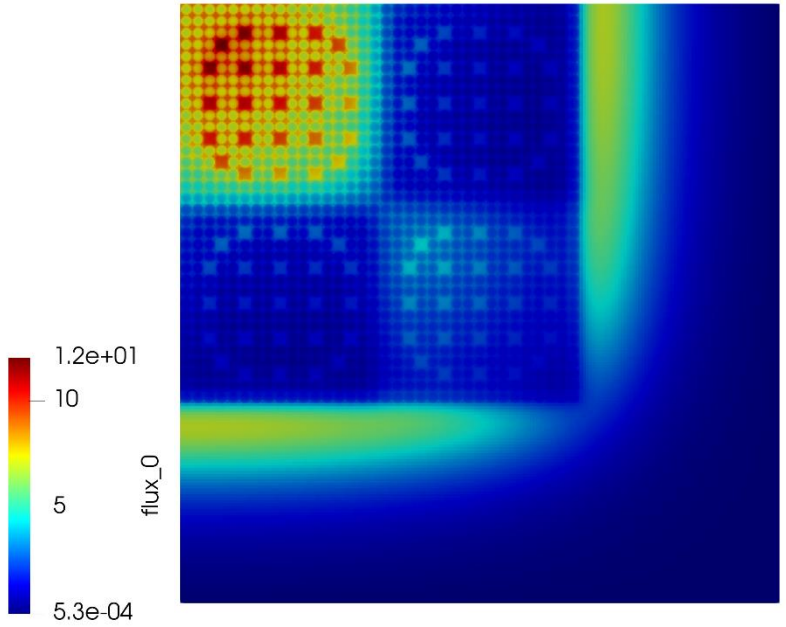
Reference (MCNP)	NuDEAL	
	w/o CMFD	w/ CMFD
1.18655 (±0.00003)	1.18652	

- **Group Flux and Fission Source Distributions**

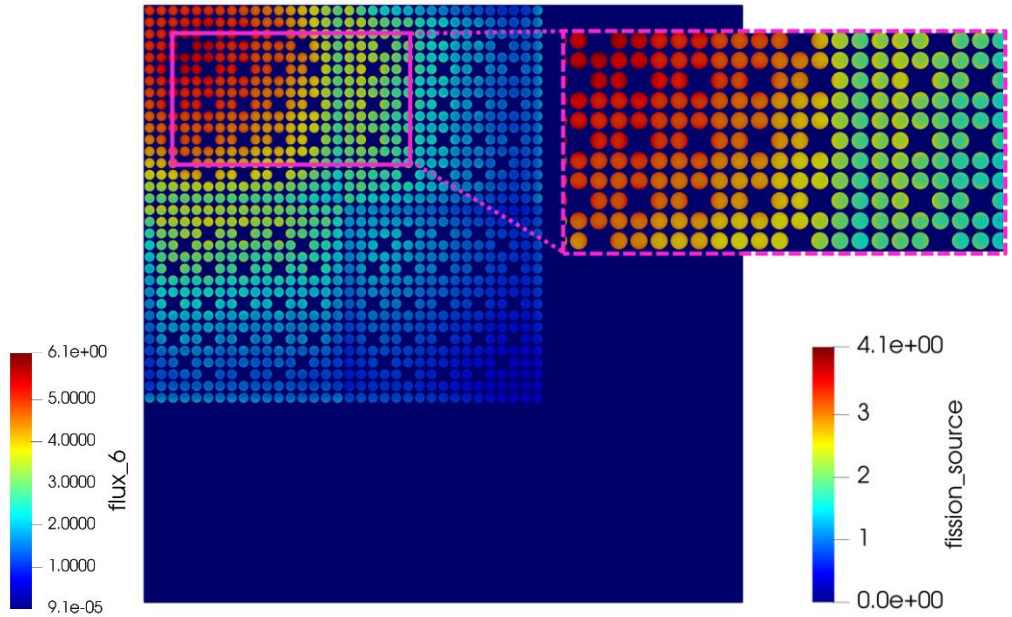
Group 1 Flux



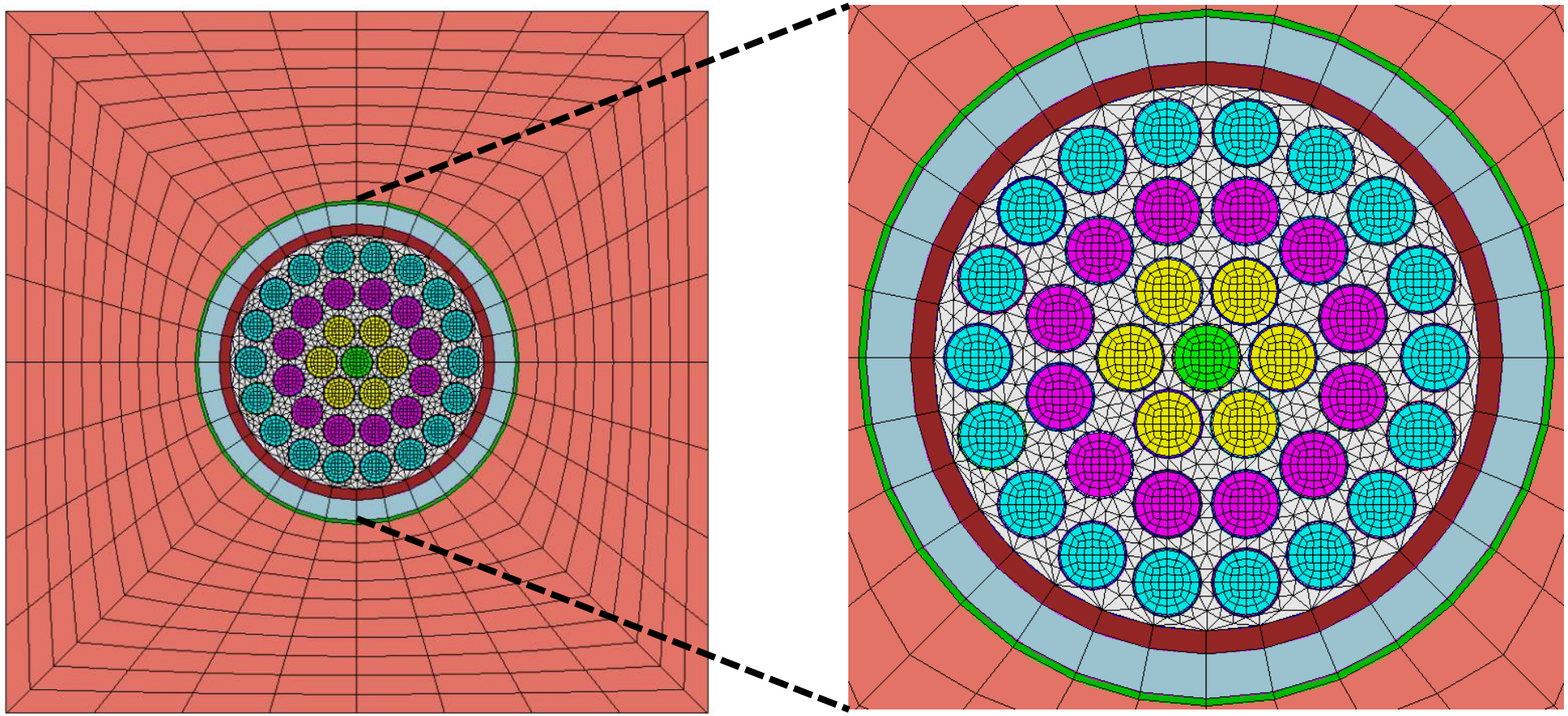
Group 7 Flux



Fission Source Distribution

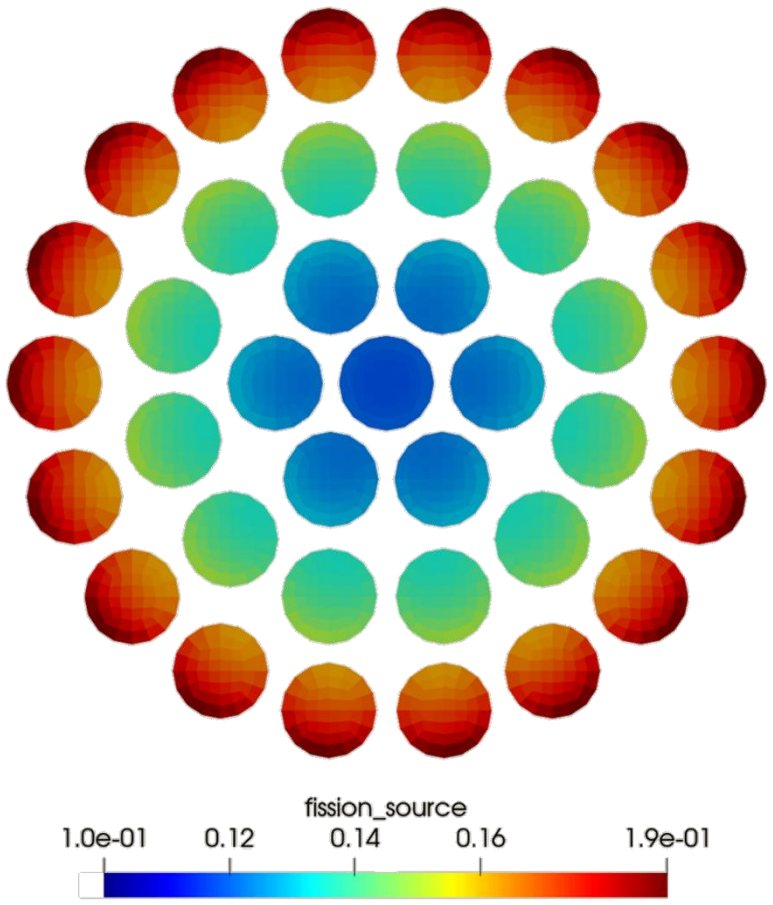


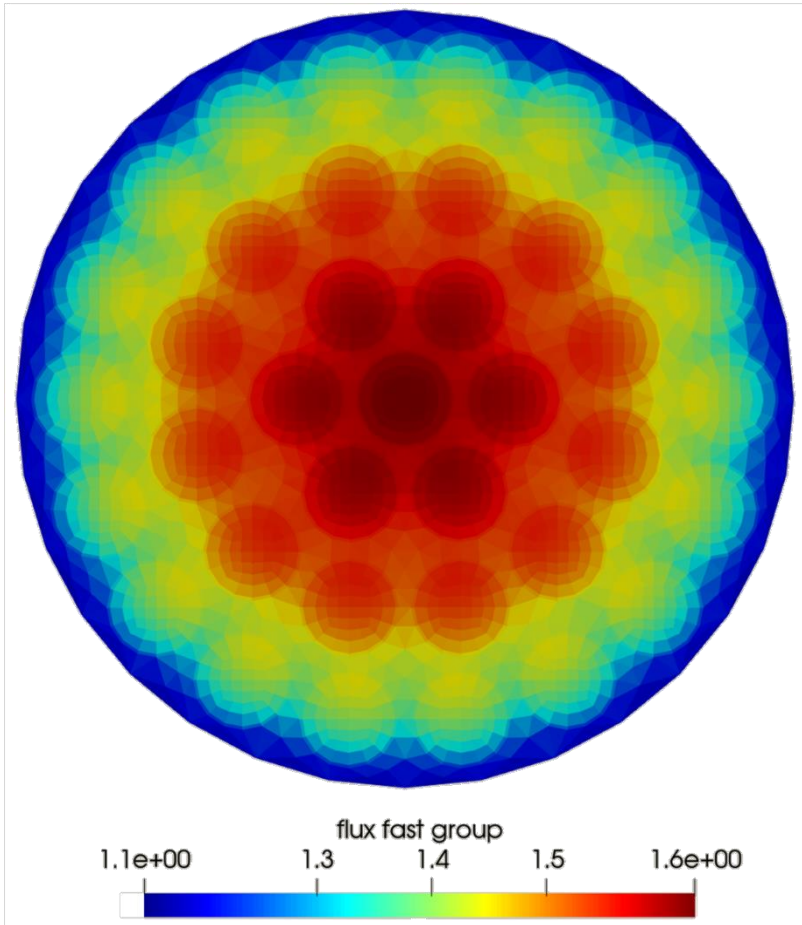
- **2D 2G CANDU Lattice Problem***
  - 3,750 FSRs / 0.01 cm of ray spacing / 16 azimuthal angles / 4 polar angles

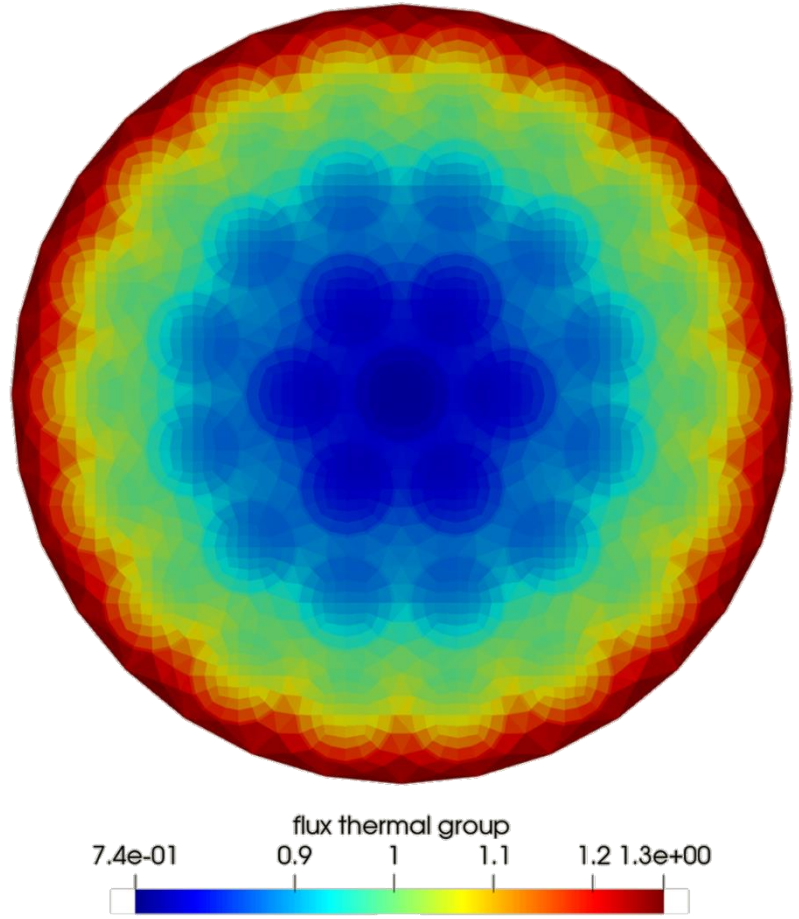


*J. M. Pounders *et al.*, "A 3D Stylized Half-core CANDU Benchmark Problem"

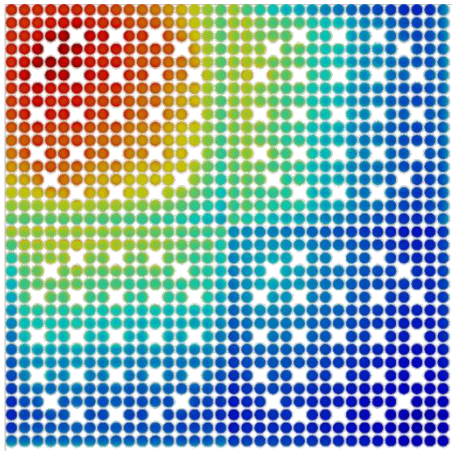
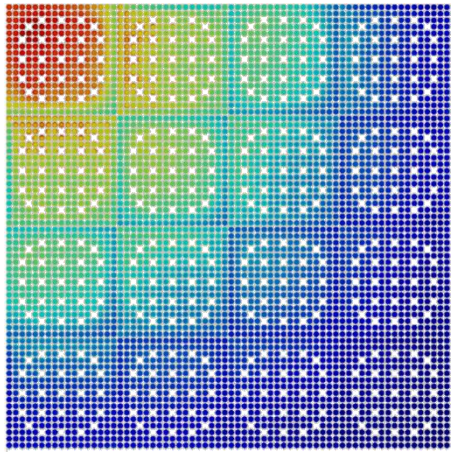
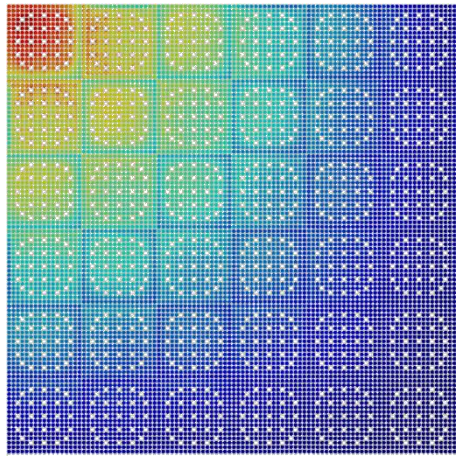
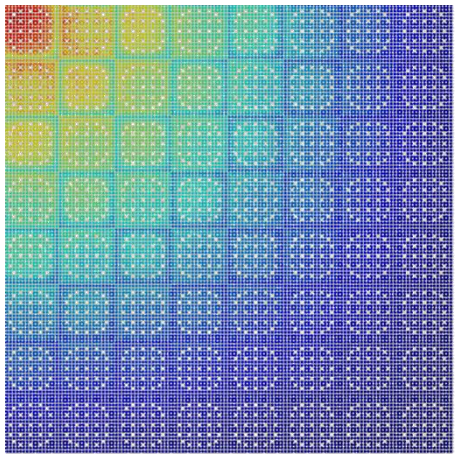
- **2D 2G CANDU Lattice Problem**





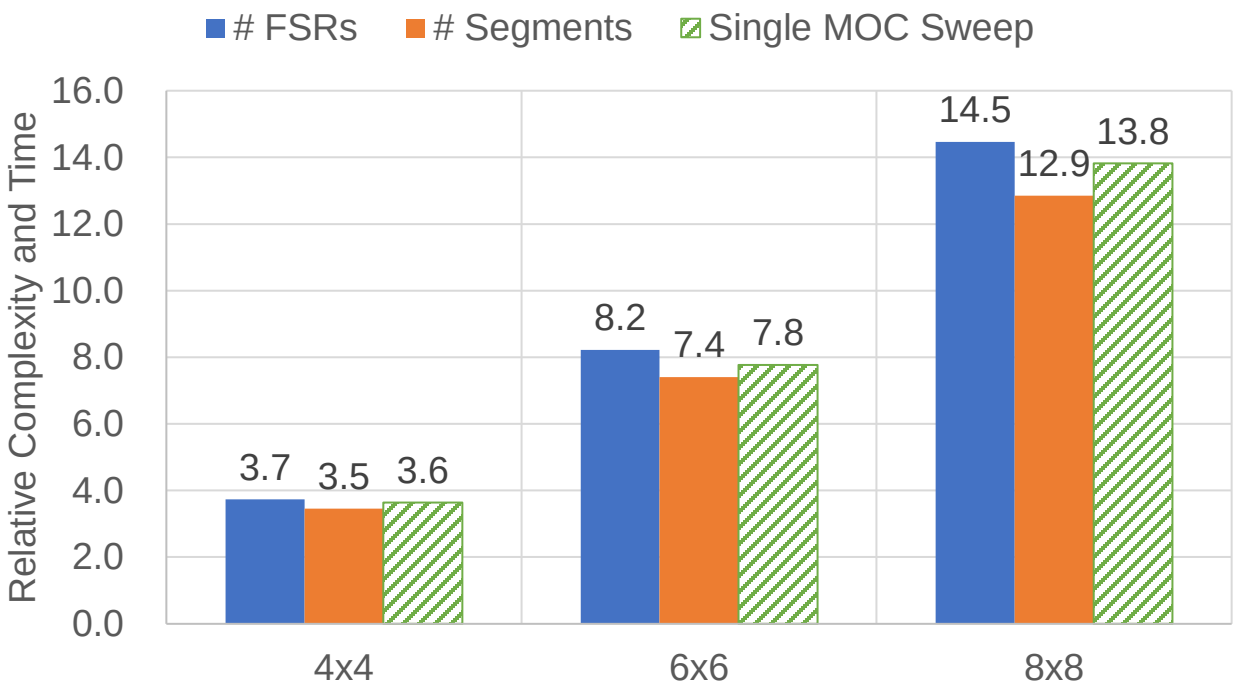


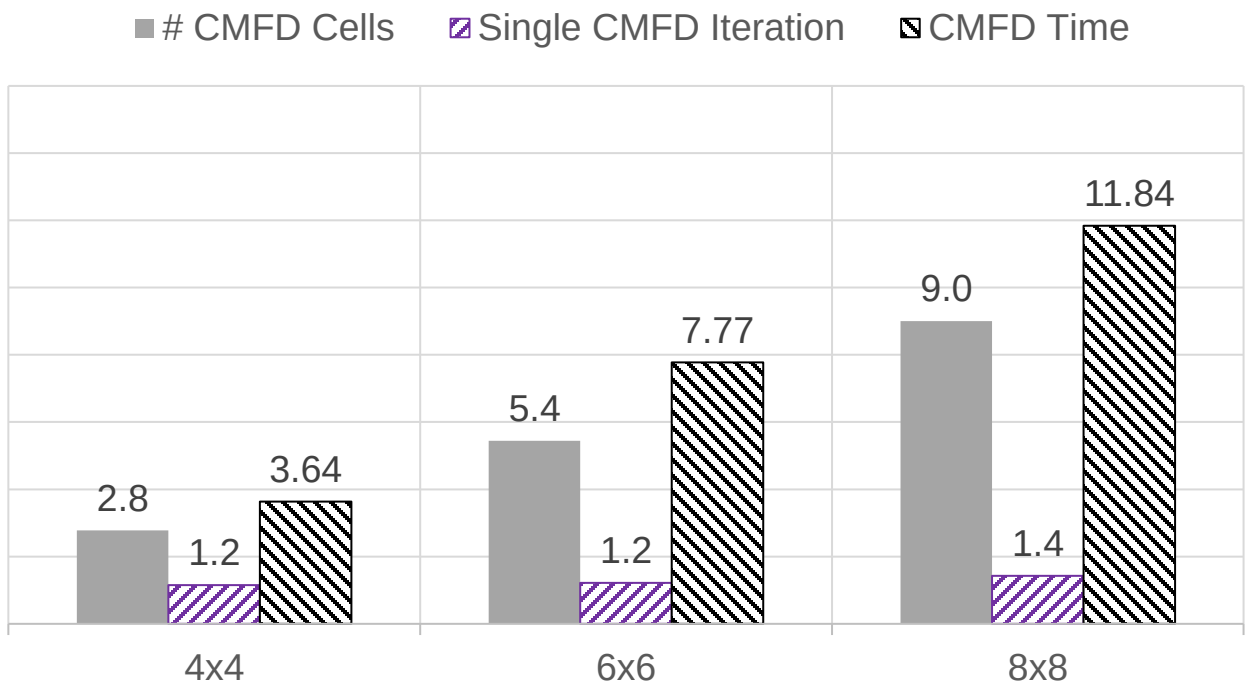
- Extended C5G7 2D Problems

Fuel Assemblies	2x2	4x4	6x6	8x8
# FSRs	189,584	707,472	1,558,288	2,742,032
# Segments	23,910,664	82,506,571	176,982,765	307,351,503
# CMFD Cells	2,601	7,225	14,161	23,409
MOC (# iter.)	3.18s (7)	11.57s (7)	24.72s (7)	37.662s (6)
CMFD (# iter.)	2.5s (97)	9.0s (298)	18.7s (586)	30.2s (812)

- **Extended C5G7 2D Problems**
  - **A single MOC sweep is near the ideal case.**
  - **A single CMFD iteration shows great scalability, but the number of iterations dominates.**
    - Preconditioners will resolve this.





Conclusion

- **Development of GPU-based MOC over Unstructured Mesh Geometry**
  - Array-based half-edge data structure enjoyed when establishing the mesh data
  - Initialization of MOC calculations built without severe effort or computational burden
  - Working well with the unstructured mesh geometry
  - ➢ It is shown that a single MOC calculation has near-ideal performance.

- **Implementation of Consistent CMFD Adapted for Unstructured Mesh**
  - Tailored partitioning algorithm employed as a coarse mesh generation scheme
  - Solution inconsistency resolved via PAXS enforcing the neutron balance
  - ➢ CMFD successfully accelerates the transport calculation.

➢ **GPU acceleration shows outstanding performance even in the unstructured mesh geometry.**

- **High/Low-order 3D Solver Stable and Accurate Enough**
  - **The MOC with the discontinuous Galerkin method was already implemented and investigated.**
    - Excessive memory requirement and poor acceleration gains
  - ➢ **More practical solvers should be devised and developed.**
  - ➢ **Variational nodal method (VNM) is being investigated as an alternative.**

C5G7 3D Core Problem Solution Obtained from MOC with the Discontinuous Galerkin Method