Efficient Parallelization Strategy of STREAM for Three-dimensional Whole-core Neutron Transport Calculation

Sooyoung Choi and Deokjung Lee*

Ulsan National Institute of Science and Technology (UNIST)

CONTACT

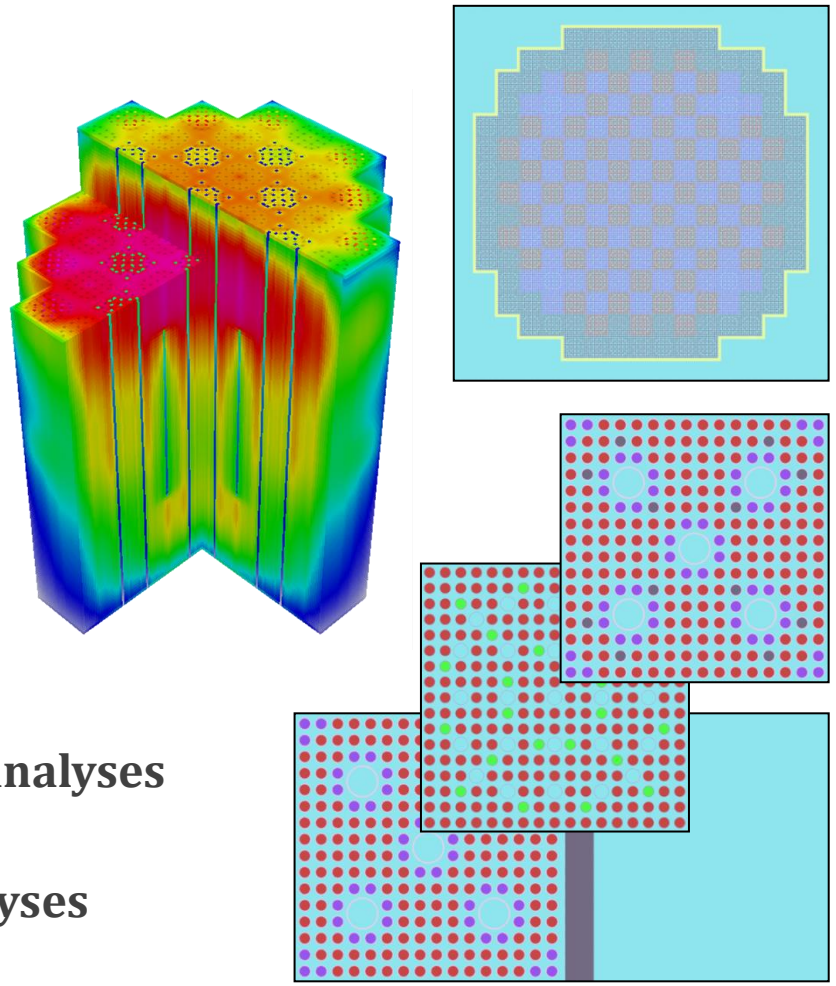
Ulsan National Institute of Science and Technology
Address 50 UNIST-gil, Ulju-gun, Ulsan, 44919, Korea
Tel. +82 52 217 0114 Web. www.unist.ac.kr

Computational Reactor physics & Experiment lab
Tel. +82 52 217 2940 Web. reactorcore.unist.ac.kr

Contents

Introduction

- **Neutron Transport Analysis Code STREAM**
  - **Features and methodologies**
    - 2D & 3D transport calculations
    - Hybrid MPI/OPENMP parallelization
    - Resonance self-shielding using PSM
    - Depletion calculation
    - On-the-fly energy release model
    - Sub-channel T/H feedback
    - Automatic thermal expansion
    - Few-group constant generation
    - CBC search, Equilibrium-Xe feedback
    - 6 symmetric modeling options
    - Source term calculation
  - **Applications**
    - PWR fuel pin / assembly / whole core analyses
    - STREAM/RAST-K 2-step procedure
    - UO_2, gadolinia, MOX and R-BA fuel analyses
    - Spent fuel analysis



Methodology

- **2D/3D Method in STREAM**
  - **Method of Characteristics (MOC) + discontinuous Galerkin (DG) method**
  - **Flux & source approximations**
    - **Combination of radial and axial components**

$$\hat{\Phi}^{g}_{i,j,k}(s,z) \approx \sum_{n} \psi^{g}_{i,j,k,n}(s) b_n(z) \qquad \phi^{g}_{m} \approx \sum_{n} \phi^{g}_{m,n} b_n(z) \qquad \hat{Q}^{g}_{i,j,m} \approx \sum_{n} Q^{g}_{i,j,m,n} b_n(z)$$

    - **Linear polynomial function for axial flux representation**
  - **3D transport equation with approximated variables**

$$\Omega \cdot \nabla \hat{\Phi}^{g}_{i,j,k}(s,z) + \Sigma^{g}_{tr,m} \hat{\Phi}^{g}_{i,j,k}(s,z) = \hat{Q}^{g}_{i,j,m}$$

    - **By using orthogonal properties and integrating over discretized axial domain,**

$$\cos\bar{\theta}_j \frac{d\psi^{g}_{i,j,k,n}(s)}{ds} + \tilde{\Sigma}^{g}_{tr,m,n} \psi^{g}_{i,j,k,n}(s) = S^{g}_{i,j,m,n}$$

      - **where**

$$\begin{cases} \tilde{\Sigma}^{g}_{tr,m,1} = \Sigma^{g}_{tr,m} + \dfrac{2\sin\bar{\theta}_j}{\Delta z} \\ \tilde{\Sigma}^{g}_{tr,m,2} = \Sigma^{g}_{tr,m} \end{cases} \qquad \begin{cases} S^{g}_{i,j,m,1} = Q^{g}_{i,j,m,1} + \dfrac{2\sin\bar{\theta}_j}{\Delta z} \hat{\Phi}^{g,\mp}_{i,j,m} \\ S^{g}_{i,j,m,2} = Q^{g}_{i,j,m,2} \end{cases}$$

Methodology

2D/3D Method in STREAM

- **Pros**
  1. **Similar formula as a conventional 2D MOC** $\cos\bar{\theta}_j \frac{d\psi^g_{i,j,k,n}(s)}{ds} + \tilde{\Sigma}^g_{tr,m,n}\psi^g_{i,j,k,n}(s) = S^g_{i,j,m,n}$
     - Easy to implement 3D solver
  2. **Implicit evaluation of second order flux**
     - Reduced # of operations
     - PROTEUS-MOC solves 1st and 2nd order fluxes by using matrix form
  3. **Stable convergence behavior**
     - 2D/1D method has stability issue for problems with large axial leakage
  4. **Angle dependent axial source**
     - Detailed representation of axial neutron streaming
- **Cons**
  1. **Much expensive compared to 2D/1D method**
     - Difficult to use sub-plane method
     - Need to implement higher order basis function (future work)
  2. **Limited to extruded geometry modeling**

Methodology

Algorithm Transport sweep for STREAM3D

```
do Outer iterations
  CMFD acceleration
  Update fission source
  do Inner iterations
    do Assemblies                                   ! MPI parallelization & Domain decomposition
      Collect boundary angular flux                 ! MPI communication with adjacent assemblies
      Update scattering source
      do Upward/downward sweepings                  ! Upward (θp = 0~π/2) / Downward (θp= -π/2~0)
        do Planes
          Get surface source                        ! Surface sources from adjacent plane or boundary
          do Azimuthal angles                       ! OPENMP parallelization
            do Parallel rays                        ! Parallel MOC rays in an assembly
              do Segments                           ! Sequential segments in a MOC ray
                do Energy groups
                  do Polar angles
                    Compute and collect angular flux change
                    Update segment outgoing flux
                  end do
                end do
              end do                                ! Forward/backward sweeping is omitted
            end do
          end do
          Update axial surface sources              ! Surface source for adjacent plane
          Compute scalar fluxes                     ! Scalar flux of current plane
        end do
      end do
    end do
  end do
  Check convergence
end do
```

Methodology

Algorithm Transport sweep for STREAM3D

```
do Outer iterations
  CMFD acceleration
  Update fission source
  do Inner iterations
    do Assemblies                                          ! MPI parallelization & Domain decomposition
      Collect boundary angular flux                        ! MPI communication with adjacent assemblies
      Update scattering source
      do Upward/downward sweepings                         ! Forward/backward sweeping is omitted
        do Planes
          Get surface source                               ! Surface source for adjacent plane
          do Azimuthal angles
            do Parallel rays                               ! OPENMP parallelization
              do Segments                                  ! Parallel computing possible
                do Energy groups                           ! Sequential segments in MOC ray
                  do Polar angles
                    Compute and collect angular flux change
                    Update segment outgoing flux
                  end do
                end do
              end do
            end do
          end do
          Update axial surface sources                     ! Surface sources from adjacent plane or boundary
          Compute scalar fluxes                            ! Scalar flux of current plane
        end do
      end do
    end do
  end do
  Check convergence
end do
```

- Domain decomposition to reduce a huge amount of memory required in whole core simulation
- CMFD can eliminate slower convergence behavior arising from discontinuous ray tracking
- Plane-wise decomposition requires a lot of memory for STREAM 3D method

Methodology

▪ Estimation of memory requirement for whole core simulation

- **Plane decomposition**

Category	Values
# of 2D FSRs/assembly	10,000
# of assemblies	300
# of energy groups	72
# of azimuthal angles	48
# of polar angles	6
Real type (byte)	4
Memory / domain (GB)	249
# of axial planes	200
Total memory (GB)	49,766

- **Assembly decomposition**

Category	Values
# of boundary points/assembly	1,000
# of axial planes	200
# of energy groups	72
# of azimuthal angles	48
# of polar angles	6
Real type (byte)	4
Memory / domain (GB)	17
# of assemblies	300
Total memory (GB)	4,977

› **Assuming 0.05 cm MOC ray spacing / ~20 cm assembly pitch / 2 cm plane height**

– **Plane-wise domain decomposition requires a lot of memory to link surfaces**

› **Angle dependent angular sources for each flat source region need to be stored**

Methodology

Algorithm Transport sweep for STREAM3D

```
do Outer iterations
  CMFD acceleration
  Update fission source
  do Inner iterations
    do Assemblies                                   ! MPI parallelization & Domain decomposition
      Collect boundary angular flux                 ! MPI communication with adjacent assemblies
      Update scattering source
      do Upward/downward sweepings                  ! Upward (θp = 0~π/2) / Downward (θp= -π/2~0)
        do Planes
          Get surface source                        ! Surface sources from adjacent plane or boundary
          do Azimuthal angles                       ! OPENMP parallelization
            do Parallel rays
              do Segments
                do Energy groups
                  do Polar angles
                    Compute and collect angular flux change
                    Update segment outgoing flux
                  end do
                end do
              end do
            end do
          end do
          Update axial surface sources              ! Surface source for adjacent plane
          Compute scalar fluxes                     ! Scalar flux of current plane
        end do
      end do
    end do
  end do
  Check convergence
end do
```

- Each MPI process calls OPENMP threads
- OPENMP threads distribute work with shared memory allocated for each MPI process
- No need to synchronize angular flux

Methodology

Algorithm Transport sweep for STREAM3D

```
do Outer iterations
  CMFD acceleration
  Update fission source
  do Inner iterations
    do Assemblies                                   ! MPI parallelization & Domain decomposition
      Collect boundary angular flux                 ! MPI communication with adjacent assemblies
      Update scattering source
      do Upward/downward sweepings                  ! Upward (θp = 0~π/2) / Downward (θp= -π/2~0)
        do Planes
          Get surface source                        ! Surface sources from adjacent plane or boundary
          do Azimuthal angles                       ! OPENMP parallelization
            do Parallel rays                        ! Parallel MOC rays in an assembly
              do Segments                           ! Sequential segments in a MOC ray
                do Energy groups
                  do Polar angles
                    Compute and collect angular flux change
                    Update segment outgoing flux
                  end do
                end do
              end do
            end do
          end do
          Update axial surface sources              ! Surface source for adjacent plane
          Compute scalar fluxes                     ! Scalar flux of current plane
        end do
      end do
    end do
  end do
  Check convergence
end do
```

- Loop for parallel ray is also one of possible choices to use OPENMP

Methodology

Algorithm Transport sweep for STREAM3D

```
do Outer iterations
  CMFD acceleration
  Update fission source
  do Inner iterations
    do Assemblies
      Collect boundary angular flux
      Update scattering source
      do Upward/downward sweepings
        do Planes
          Get surface source
          do Azimuthal angles
            do Parallel rays
              do Segments
                do Energy groups
                  do Polar angles
                    Compute and collect angular flux change
                    Update segment outgoing flux
                  end do
                end do
              end do
            end do
          end do
          Update axial surface sources
          Compute scalar fluxes
        end do
      end do
    end do
  end do
  Check convergence
end do
```

! **MPI parallelization & Domain decomposition**
! **MPI communication** with adjacent assemblies

! Upward (θ_p = 0~π/2) / Downward (θ_p= -π/2~0)

- Length of MOC ray segment projected on the *x-y* plane is the same for energy groups and polar angles
- No need to access track length data repeatedly
- Possible to reduce the number of calculations
- Possible to maximize cache usage

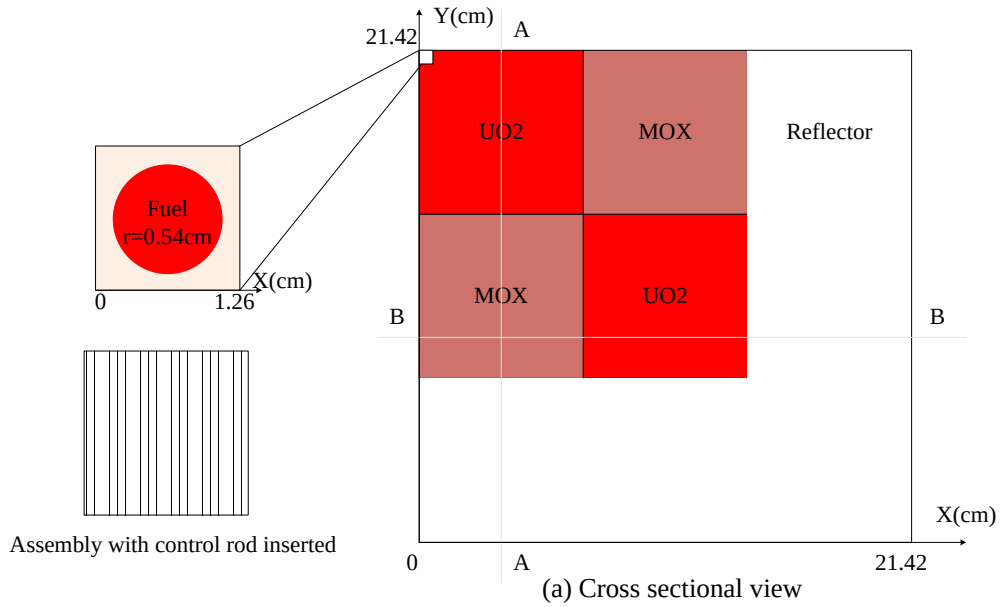
Methodology

Algorithm Transport sweep for STREAM3D

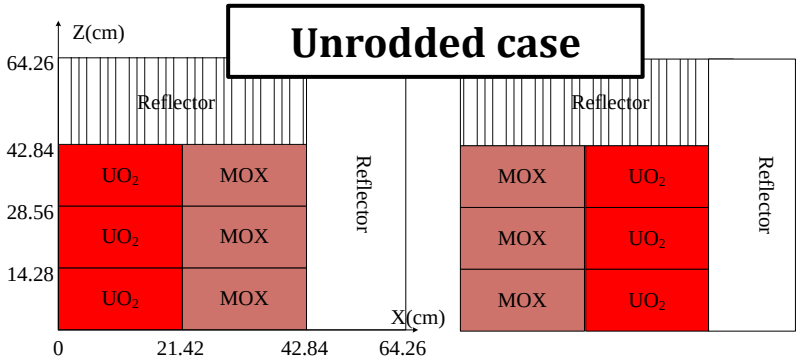
```
do Outer iterations
  CMFD acceleration
  Update fission source
  do Inner iterations
    do Assemblies                                   ! MPI parallelization & Domain decomposition
      Collect boundary angular flux                 ! MPI communication with adjacent assemblies
      Update scattering source
      do Upward/downward sweepings                  ! Upward (θp = 0~π/2) / Downward (θp= -π/2~0)
        do Planes
          Get surface source                        ! Surface sources from adjacent plane or boundary
          do Azimuthal angles                       ! OPENMP parallelization
            do Parallel rays                        ! Parallel MOC rays in an assembly
              do Segments                           ! Sequential segments in a MOC ray
                do Energy groups
                  do Polar angles
                    Compute and collect angular flux change
                    Update segment outgoing flux
                  end do
                end do
              end do                                ! Forward/backward sweeping is omitted
            end do
          end do
          Update axial surface sources              ! Surface source for adjacent plane
          Compute scalar fluxes                     ! Scalar flux of current plane
        end do
      end do
    end do
  end do
  Check convergence
end do
```

Numerical Results

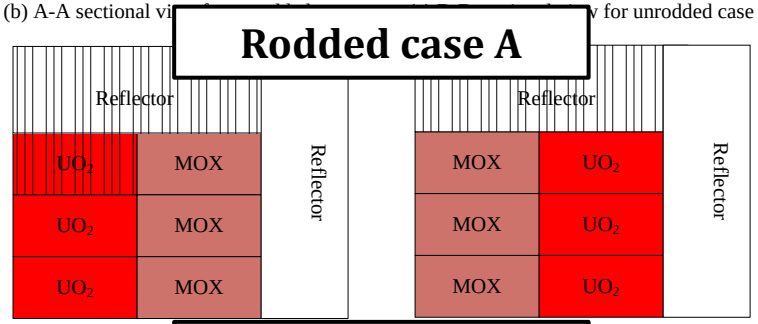
- **C5G7 benchmark**
  - **OECD/NEA C5G7 benchmark**
  - **7-group cross-sections**



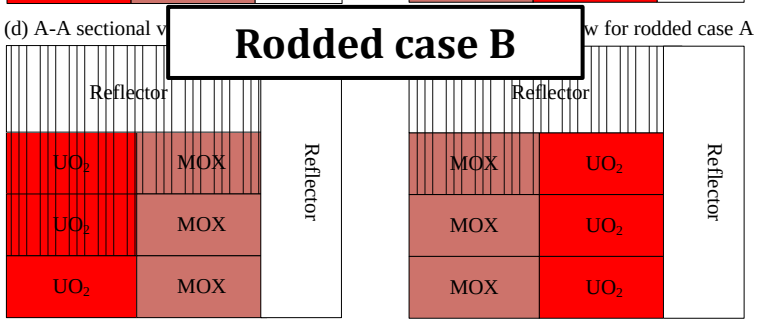
(a) Cross sectional view



(b) A-A sectional vi… for unrodded case



(d) A-A sectional v… for rodded case A

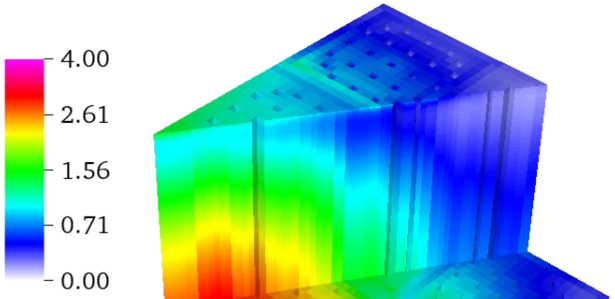


(f) A-A sectional view for rodded case B

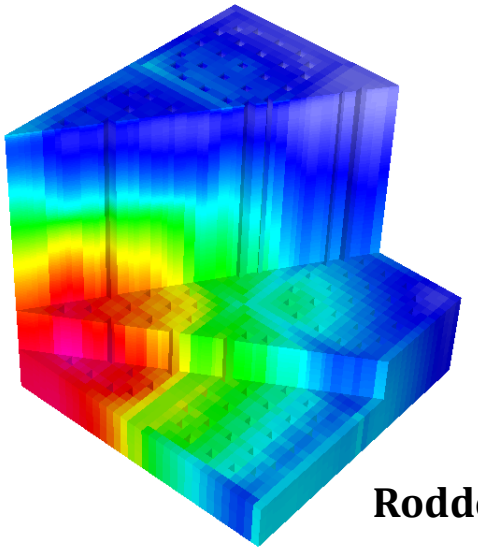
(g) B-B sectional view for rodded case B

Numerical Results

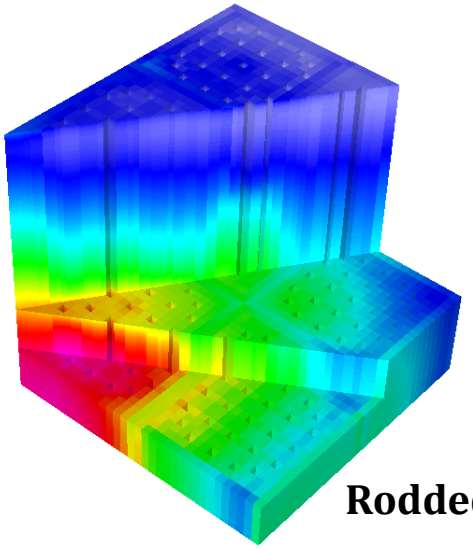
- **Pin power distribution (C5G7 benchmark + void case)**



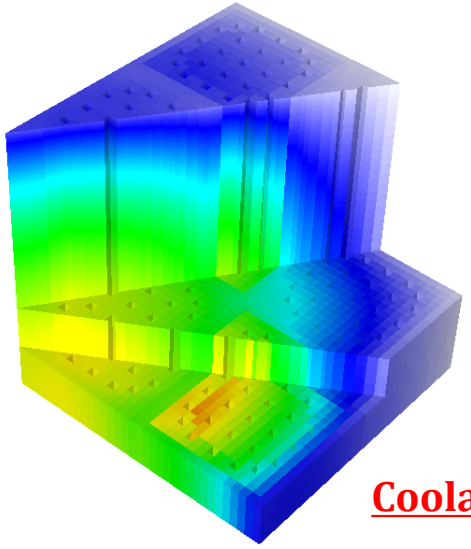
Unrodded case



Rodded case A



Rodded case B



Coolant void case

Numerical Results

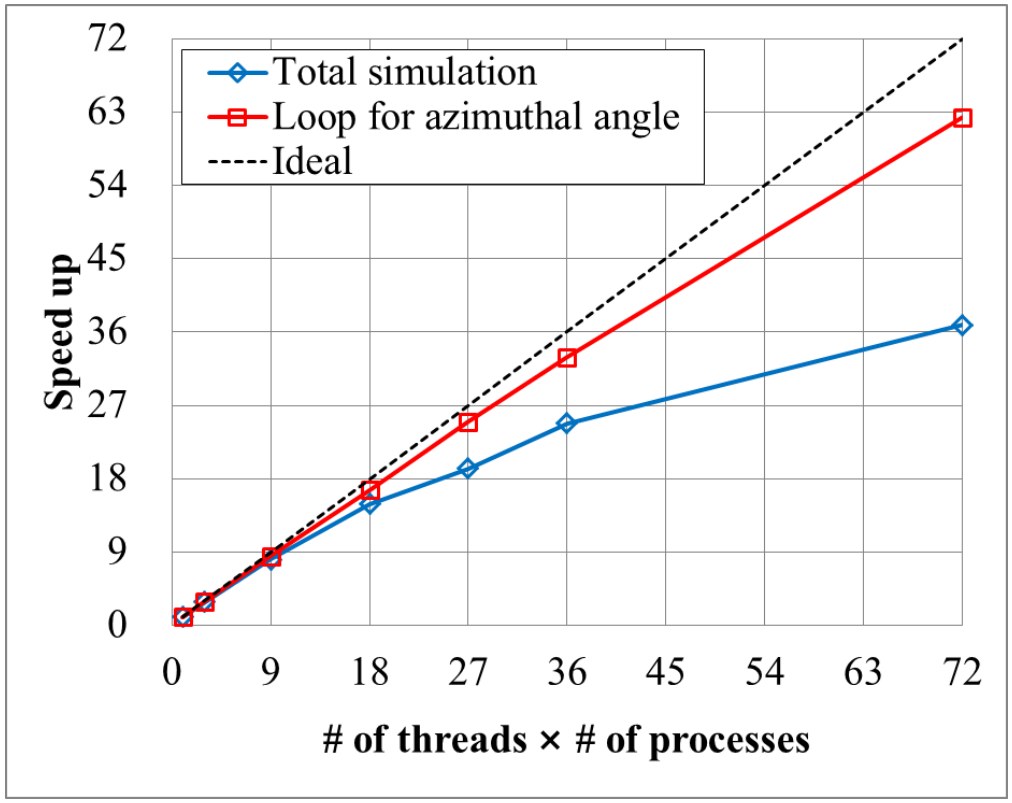
C5G7 benchmark results

- **k-eff & pint power distribution**

	k-eff			Pin power distribution			
				RMS.		Max.	
	MCS	STREAM	Difference (pcm)	Difference (%)	STD (%)	Difference (%)	STD (%)
Unrodded	1.14302 ±0.00001	1.14279	**-23**	**0.33**	±0.02	**1.13**	±0.03
Rodded A	1.12809 ±0.00001	1.12786	**-23**	**0.32**	±0.02	**1.09**	±0.03
Rodded B	1.07772 ±0.00001	1.07744	**-28**	**0.32**	±0.02	**1.10**	±0.03
Void	0.40274 ±0.00001	0.40274	**-1**	**0.10**	±0.01	**0.47**	±0.02

Numerical Results

- **Parallel performance**
  - **41,616 flat source regions in 2D plane**
  - **21 planes with 3.06 cm height**
  - **0.05 cm ray spacing/48 azimuthal angles/10 polar angles**



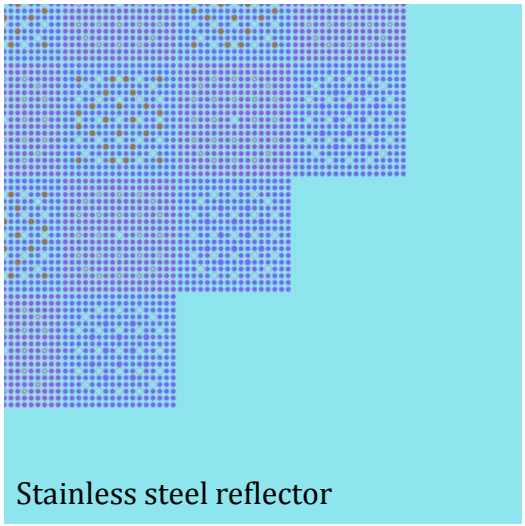
# of threads × # of processes	Simulation time (sec)	
	Total simulation	Loop for azimuthal angle
1	3,298	2,681
9	406	316
72	90	43

Applications

▪ CUSMR Design

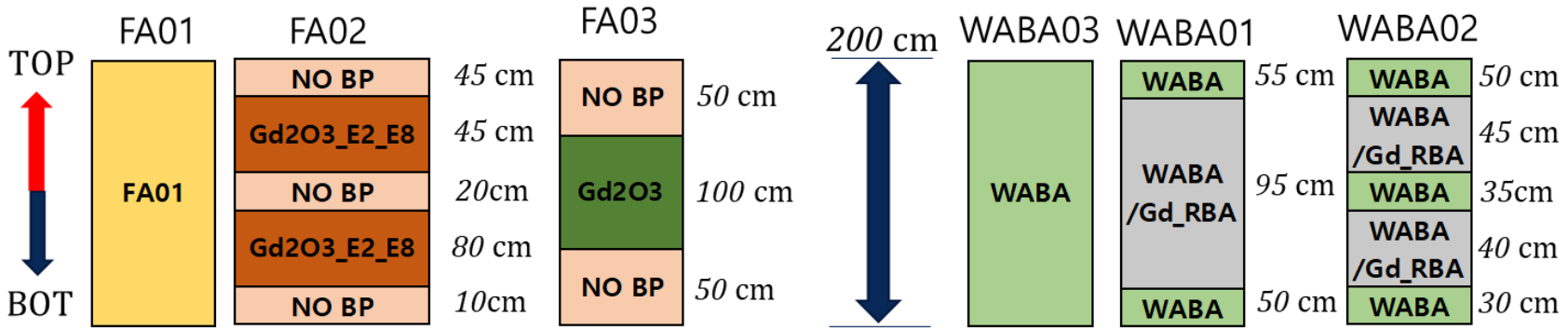
• Configuration of assembly

FA01	WABA01	FA01	WABA03	REF
WABA01	FA01	WABA02	FA02	REF
FA01	WABA02	FA03	REF	REF
WABA03	FA02	REF	REF	
REF	REF	REF		



• Design Parameters

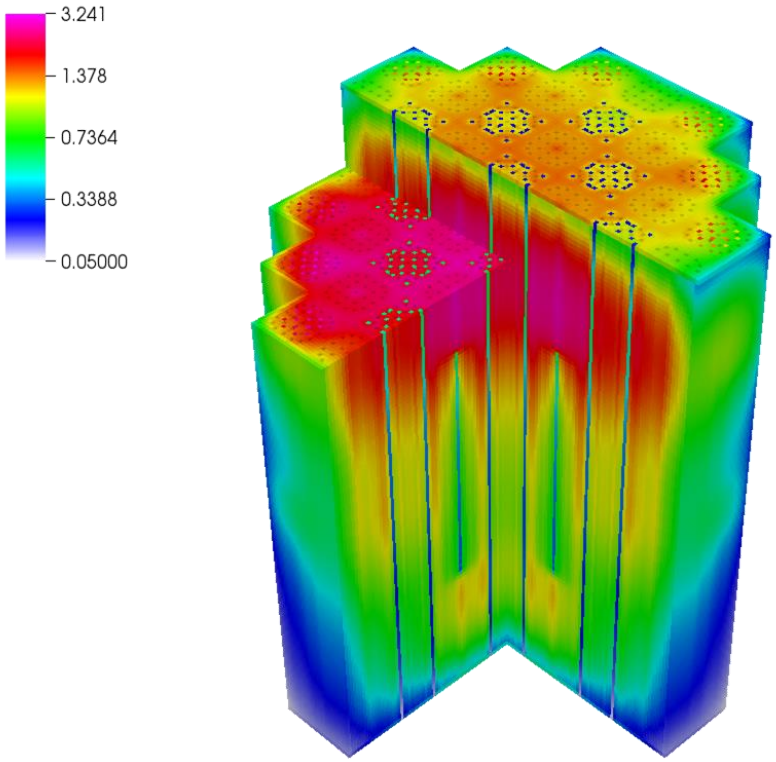
Parameter	Value
Thermal Power	180 MWth
Target cycle length	1,500 days
Fuel lattice	17x17
# of fuel assemblies	37
Active length	200 cm
Inlet Temperature	558.15 K
Assembly Pitch	21.504 cm
Pin pitch	1.260 cm



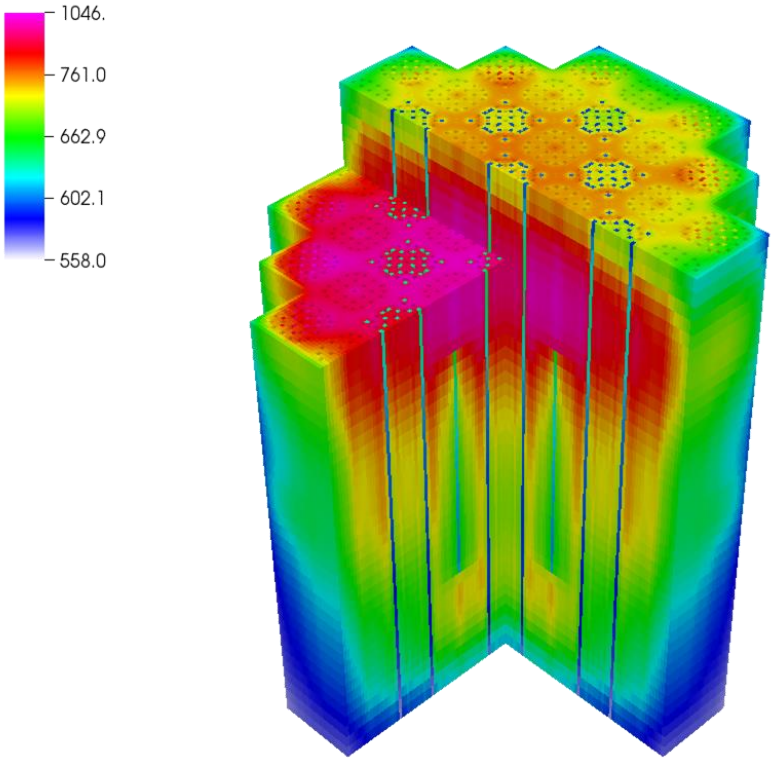
Applications

- **STREAM3D CUSMR analysis results at BOC**
  - **Calculation options: T-H feedback/Eq-Xe feedback/CBC search**

Pin power distribution



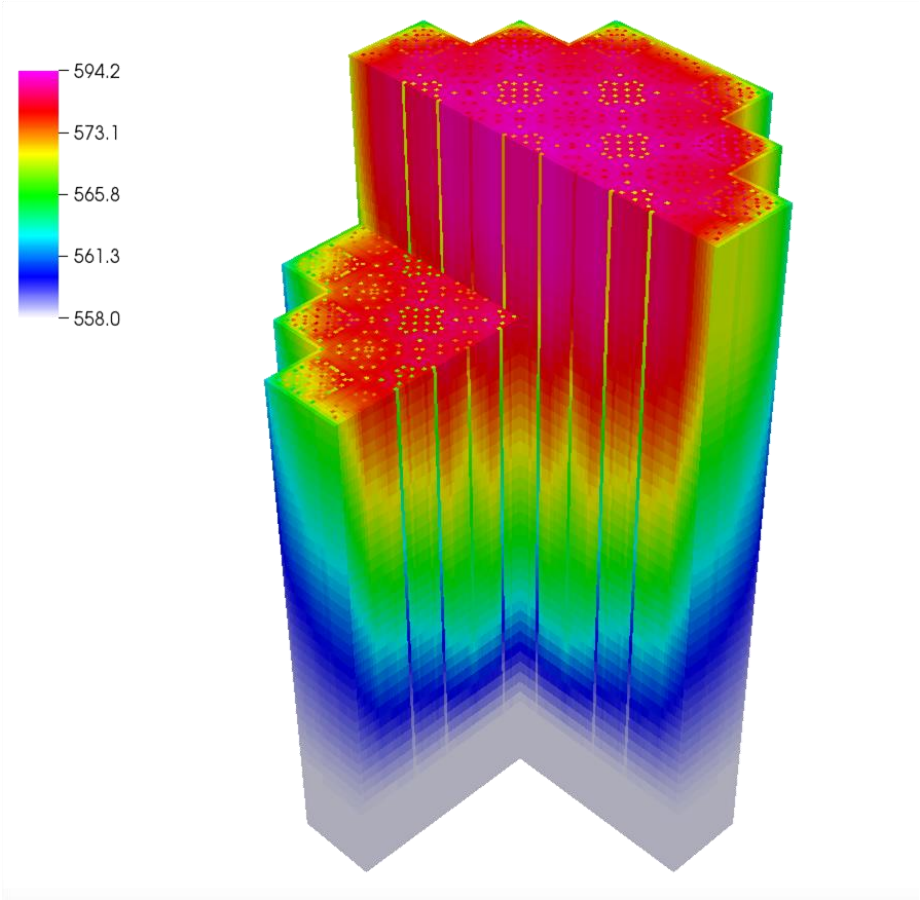
Fuel temperature (K)



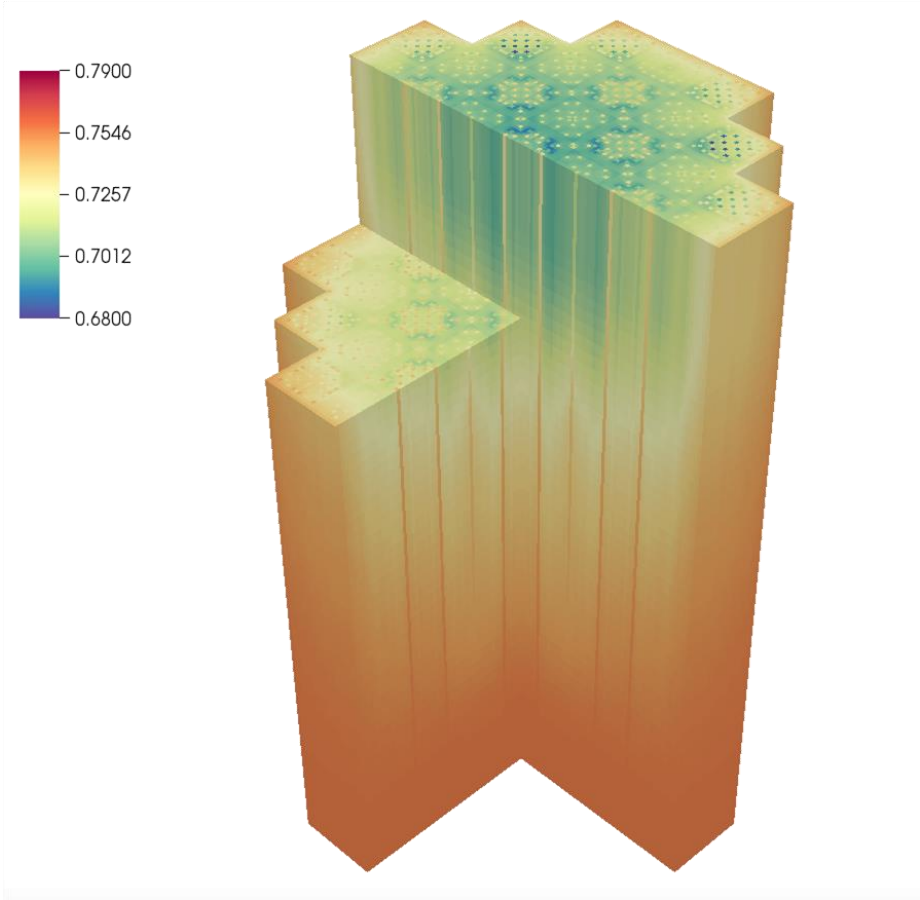
Applications

- **STREAM3D CUSMR analysis results at BOC**
  - **Calculation options: T-H feedback/Eq-Xe feedback/CBC search**

Moderator temperature (K)

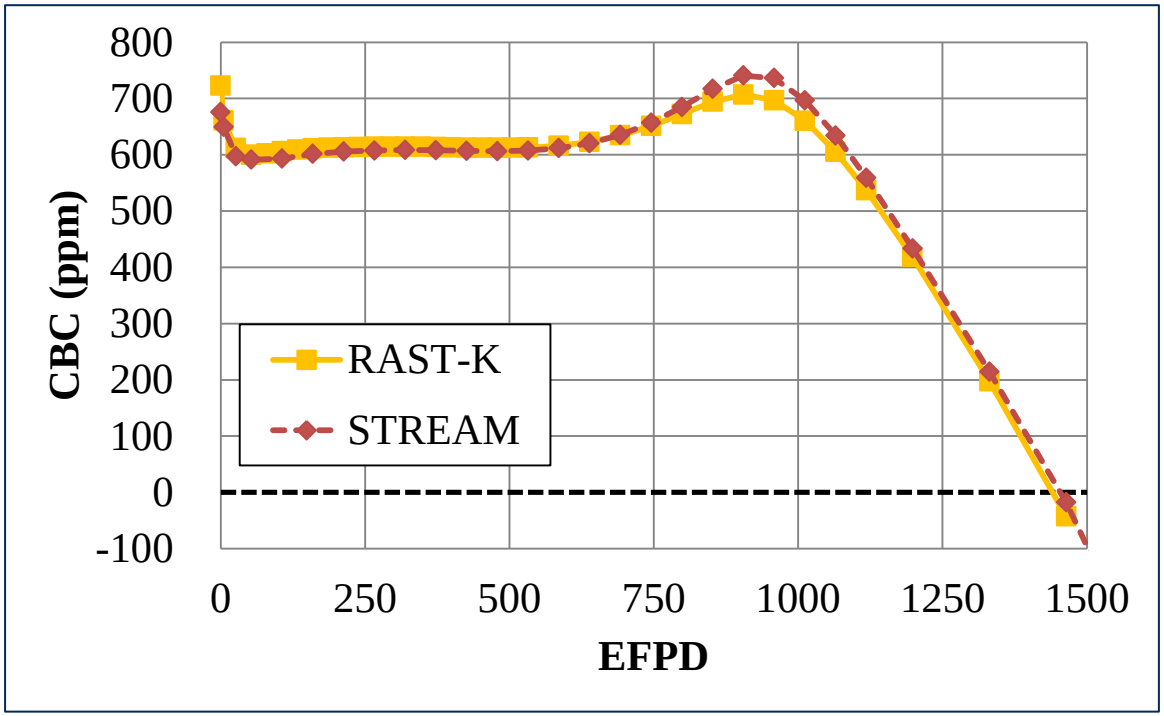


Moderator density (g/cc)



Applications

- **STREAM3D CUSMR analysis**
  - Calculation options: T-H feedback/Eq-Xe feedback/CBC search
  - 0.1 cm ray spacing/24 azimuthal angles/6 polar angles
  - 155 planes with less than 2 cm axial height
  - 28 burnup steps
  - 1.5 days spent with 84 computing cores / ~200 GB memory in total



Conclusions & Future work

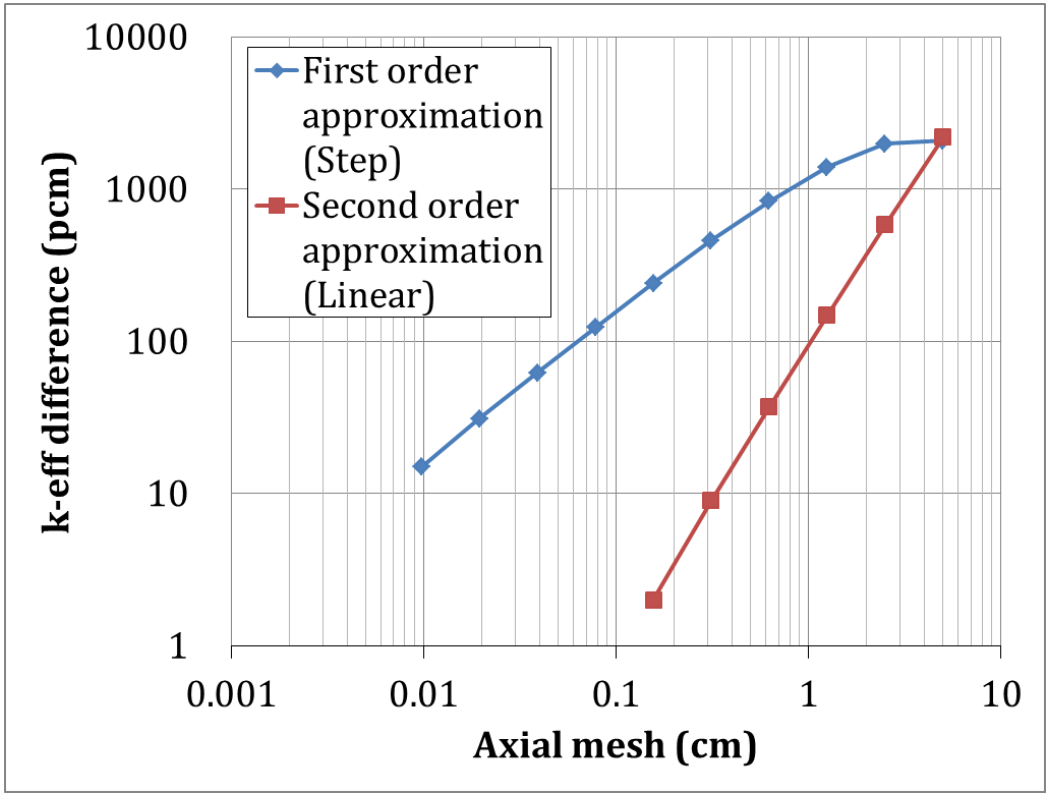
- **Conclusions**
  - **STREAM has been parallelized to solve a large size 3D problem**
  - **Algorithm for 2D/3D method has been developed**
  - **Hybrid MPI/OPENMP parallel algorithm has been developed**
  - **STREAM adopts assembly-wise domain decomposition to store required memory in distributed storage**
  - **Accuracy and performance have been examined**

- **Future work**
  - **Parallel algorithm of whole subroutines/functions will be implemented and improved**
  - **Application to 3D whole core problem**

UNIST CORE

Linear Approximation in Axial Flux Distribution

- **Extremely severe case in term of axial neutron leakage**
  - **5 cm height square cell with void boundary condition in top/bottom**



  - **Linear approximation gives much better accuracy**
  - **In general case, 2 ~ 3 cm height axial mesh gives reasonable solution**