

# **iMC GUI: An Integrated Environment for Input Authoring and CSG Geometry Visualization for Monte Carlo Simulations**

---

Sangjin Lee

sangjin.lee1012@kaist.ac.kr

2026-05-07 (Thu)

2026 KNS Spring Conference

Reactor Physics and Transmutation Laboratory (RPTL)  
Department of Nuclear and Quantum Engineering  
Korea Advanced Institute of Science and Technology (KAIST)

# Table of Contents

---

1. Introduction
2. iMC GUI Workbench
3. 2D Slice-Stack CSG Visualization
4. Future Works

# 1. Introduction

---

- ❑ iMC is a Monte Carlo particle transport code developed by the RPTL group at KAIST.
  - ❑ It currently supports many advanced features:
    - Multi-group and continuous-energy energy modes
    - CSG geometry with nested universes and lattice support
    - MPI/OpenMP hybrid parallelization
    - Neutron-photon coupled transport
    - IFP-based kinetics parameter calculation
    - iDTMC-based steady-state and depletion acceleration
    - DMC and iDTMC PCQS transient analysis
    - ...
  - ❑ However, even with these advanced capabilities, typical Monte Carlo transport workflows still have a high practical barrier.
    - text-based input construction is difficult for new and occasional users
    - CSG geometry and reactor lattices are difficult to inspect while authoring
    - execution, geometry review, and post-processing often use separate tools
    - errors may remain hidden until a run fails or produces suspicious results
- ➔ iMC is being rewritten to provide a solution to this useability problem.

## 2. iMC GUI Workbench

- ❑ iMC GUI is the new primary user interface, and is written in Python using the Qt 6 library.
  - It acts as a frontend for the iMC CLI, which has been rewritten to best support this (using Fortran OOP).
- ❑ It aims to act as a unified, integrated interface to support all workflows, and is structured around 3 main modes:
  - **Input Editor**: interactive input authoring, real-time validation, real-time visualization
  - **Run Manager**: local and remote job running, real-time result monitoring
  - **Results Viewer**: interactive result postprocessing/viewing/comparing/exporting
- ❑ The Input Editor supports an inspect, edit, and check loop.
  - Input Tree, Inspector, Problems panel, and Geometry Viewer stay visible together
  - Various convenience functions: variables, real-time input validation, geometry overlap/void inspection etc.
- ❑ The Geometry Viewer implements a 2D “slice-stack” CSG geometry visualization mode.
  - It is hard to imagine the actual 3D geometry using conventional selected-plane plotting.
  - The “slice-stack” visualization mode provides an MRI-like experience.

# 2. iMC GUI Workbench

iMC Workbench

Input Editor | Run Manager | Results Viewer

Input: movin...heres-medium | Status: Saved | Folder: /Users/sjlee/projects/programs/imc/examples/inputs/moving-spheres-medium | Copy Path

Material Materials | CSG Universe | Surface Geometry | Cell | Pin Universe | Lattice | Mesh | Filter Tallyes | Tally | Point Detector | Run Geometry Validation Validation

**Input Tree**

- Settings
- Materials (19)
- Geometry
  - Universe u\_root
    - Surfaces (6)
    - Cells (1)
      - c\_root\_sphere\_field
    - Boundaries (6)
    - Universe u\_sphere\_field
    - Universe u\_cluster\_a
    - Universe u\_cluster\_b
    - Universe u\_cluster\_c
    - Universe u\_cluster\_d
  - Tallies (3)

**Inspector**

Cell: c\_root\_sphere\_field

**Cell Details**

CSG Expression:  $+x\_min -x\_max +y\_min -y\_max +z\_min -z\_max$

Fill Type: universe

Fill Name: u\_sphere\_field

Translation: x y z

Rotation Degrees: phi theta psi

No selected cell issues.

**Referenced Surfaces**

Jump to Surface: x\_min

**Geometry Viewer**

Select | Reset View | Fit Mesh | Export | Advanced

Plane: XY | Color By: Material | Validation: Off

Z Slice (13/121): Z = -2 cm to Z = 2 cm | Exact Z Level: -1.600000

Universe u\_root

**Variables** | Problems (0) | Plot Presets | Colors

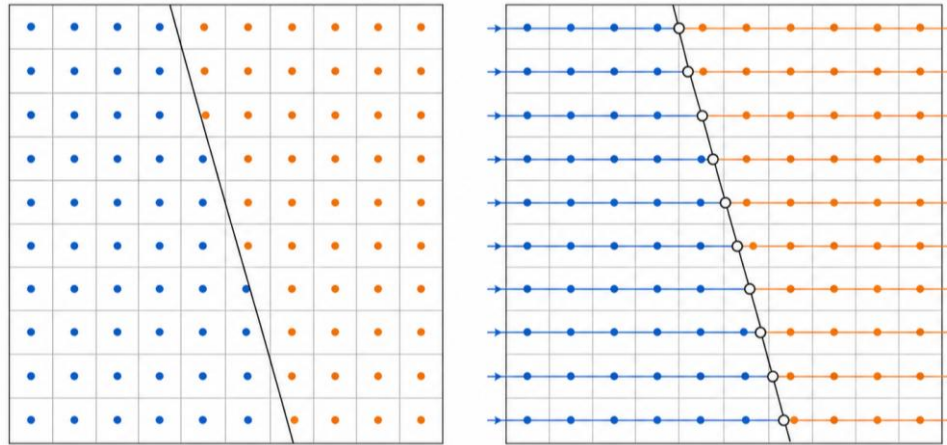
Name	Expression	Resolved Value
$\pi$	3.1415926535897931	3.14159265359
e	2.7182818284590451	2.71828182846
root_x	12.0	12
root_y	6.0	6
root_z	2.0	2
+ Add variable		

Info: Applied plot preset Root midplane

RSS 488 MiB · View cache 22/256 MiB (2)

# 3. 2D Slice-Stack CSG Visualization

- Several optimizations were introduced to make this possible.
  - Ray tracing-based “row-run” 2D slice generation (vs. point grid)



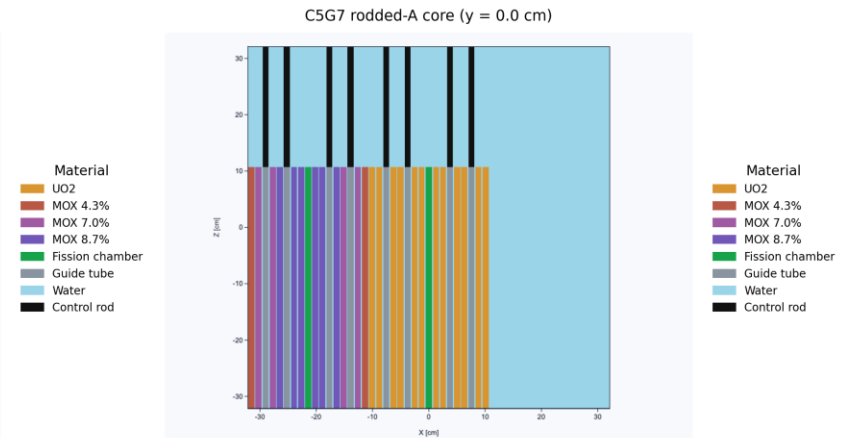
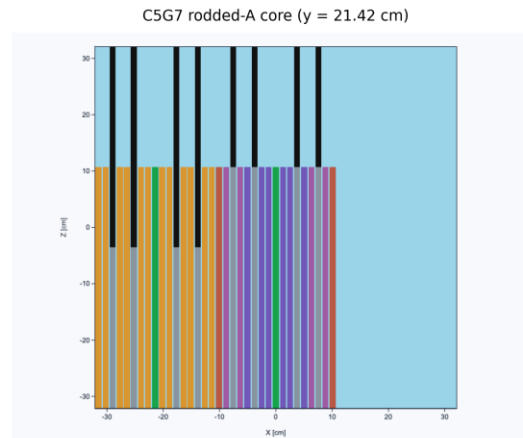
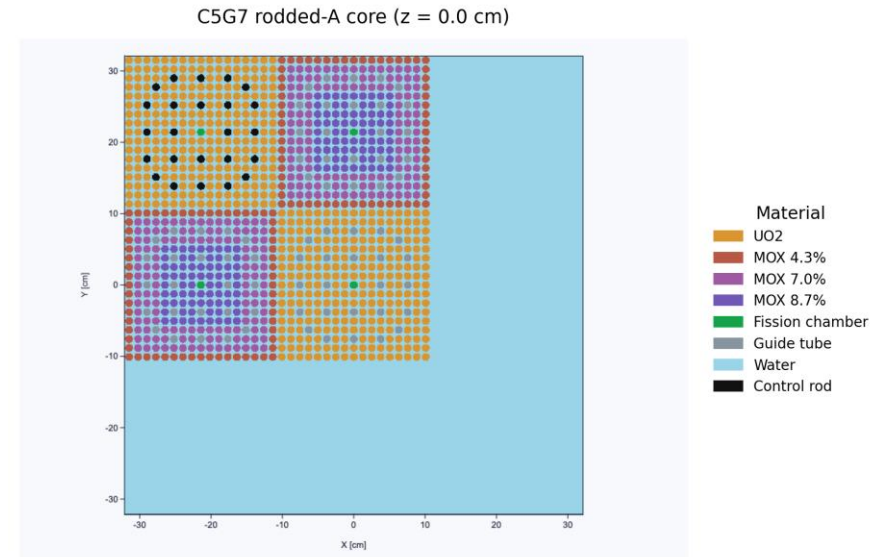
- OpenMP parallelization
- Slice prefetching and caching

# 3. 2D Slice-Stack CSG Visualization

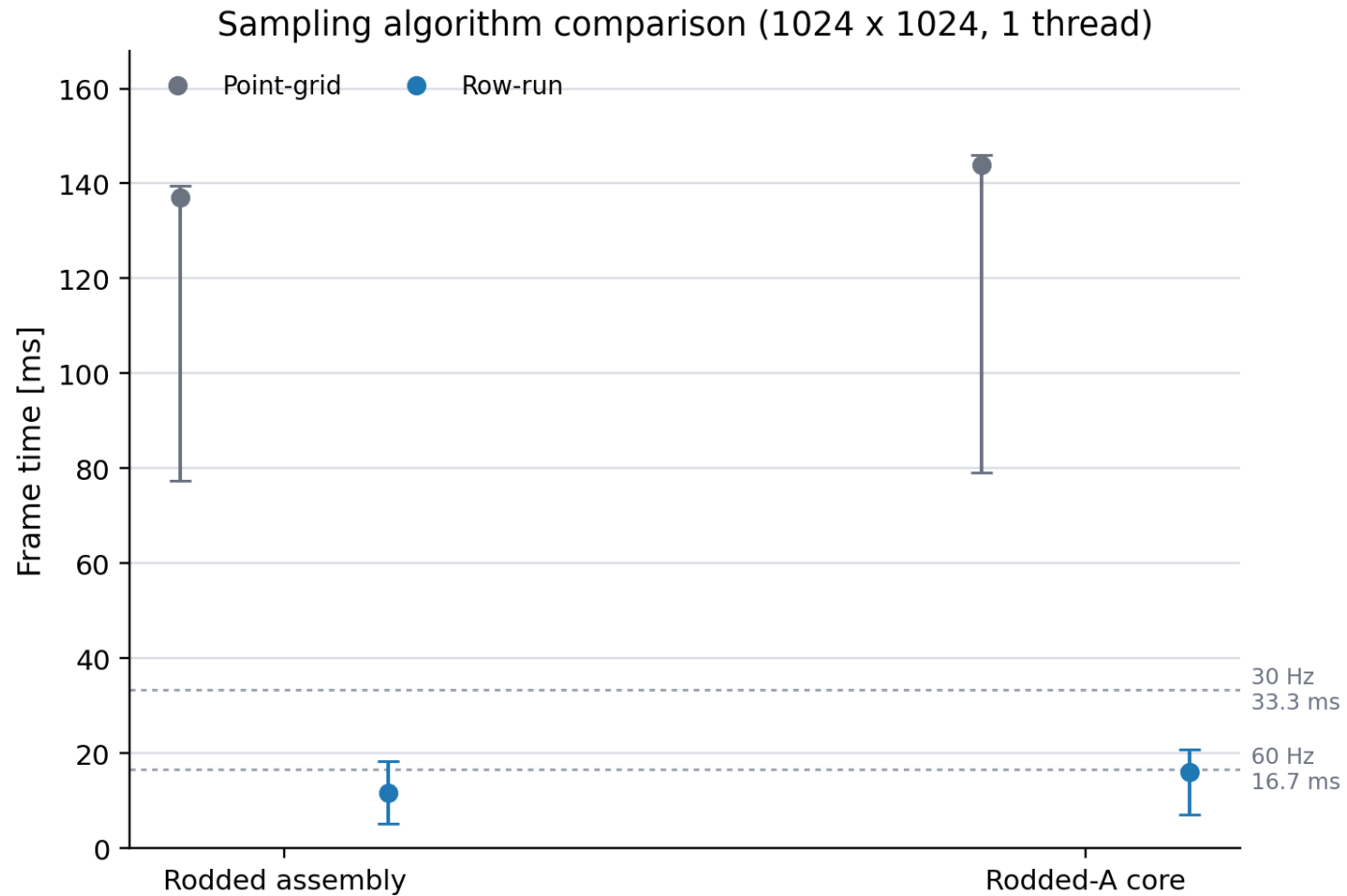
□ The main test case is the OECD/NEA C5G7 MOX 3-D Extension Benchmark.

- Rodded A configuration
- 3 x 3 quarter-core radial layout
- UO2, MOX, and reflector regions
- 17 x 17 pin lattice
- 1.26 cm pin pitch and 0.54 cm pin radius
- 21.42 cm assembly pitch and 64.26 cm height

□ □ The Rodded A case inserts control rods 1/3 into the inner UO2 assembly.



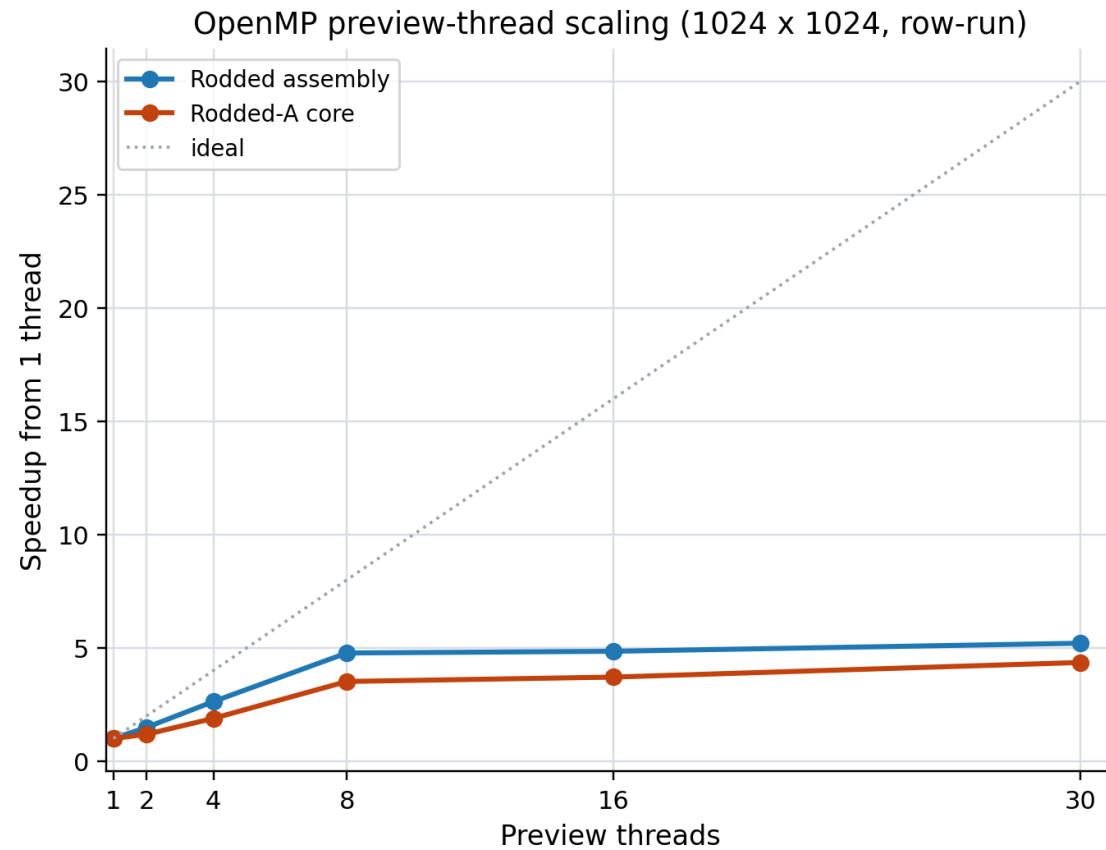
# 3. 2D Slice-Stack CSG Visualization



Dot = median; whisker = 5-95% interval.

# 3. 2D Slice-Stack CSG Visualization

- ❑ The observed limit follows Amdahl's law.
  - The serial parts of the plotter module limits the speedup gain.
- ❑ ❑ Most practical gain is reached by 8 threads.

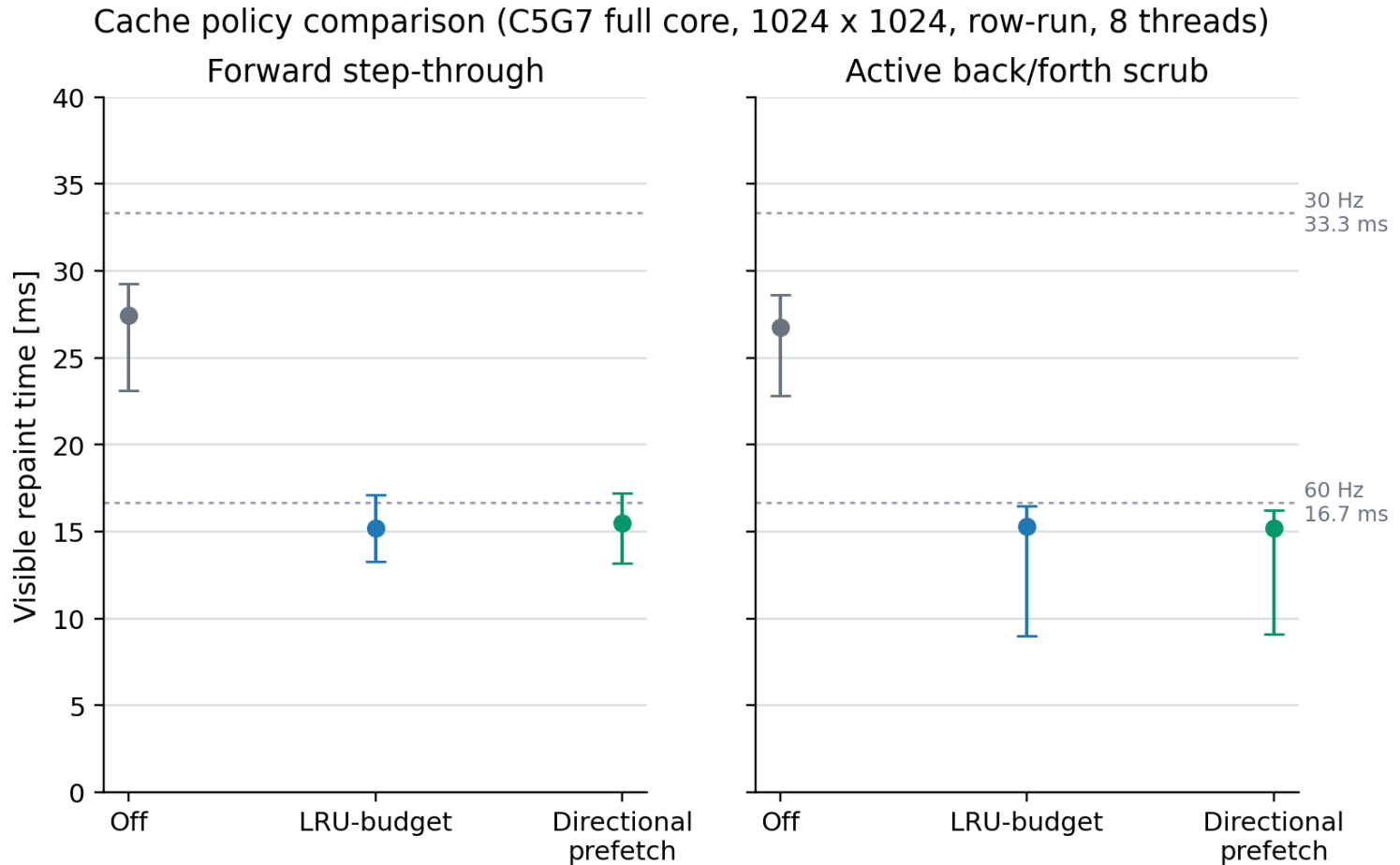


# 3. 2D Slice-Stack CSG Visualization

---

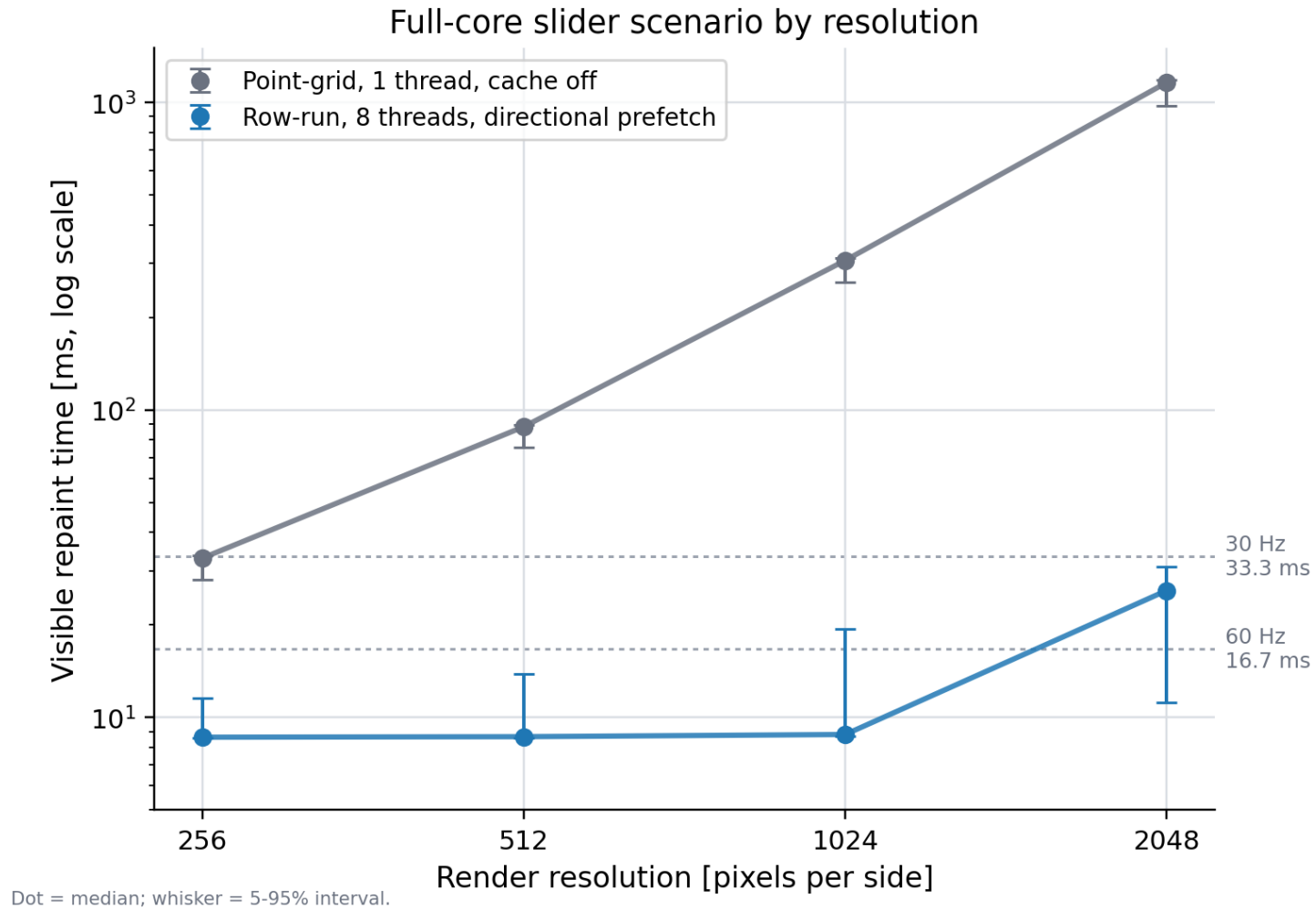
- ❑ Cache policy matters during repeated slider motion.
- ❑ Three policies were compared.
  - Off: recompute every requested slice
  - LRU-budget: "least recently used", demand cache within 256 MiB
  - Directional Prefetch: idle-time nearby slices (4 slices forward, 2 slices backward)
- ❑ The cache is memory-bounded.
  - 1024 px budget: about 32 frames
  - 201-slice stack: about 1.61 GiB

# 3. 2D Slice-Stack CSG Visualization



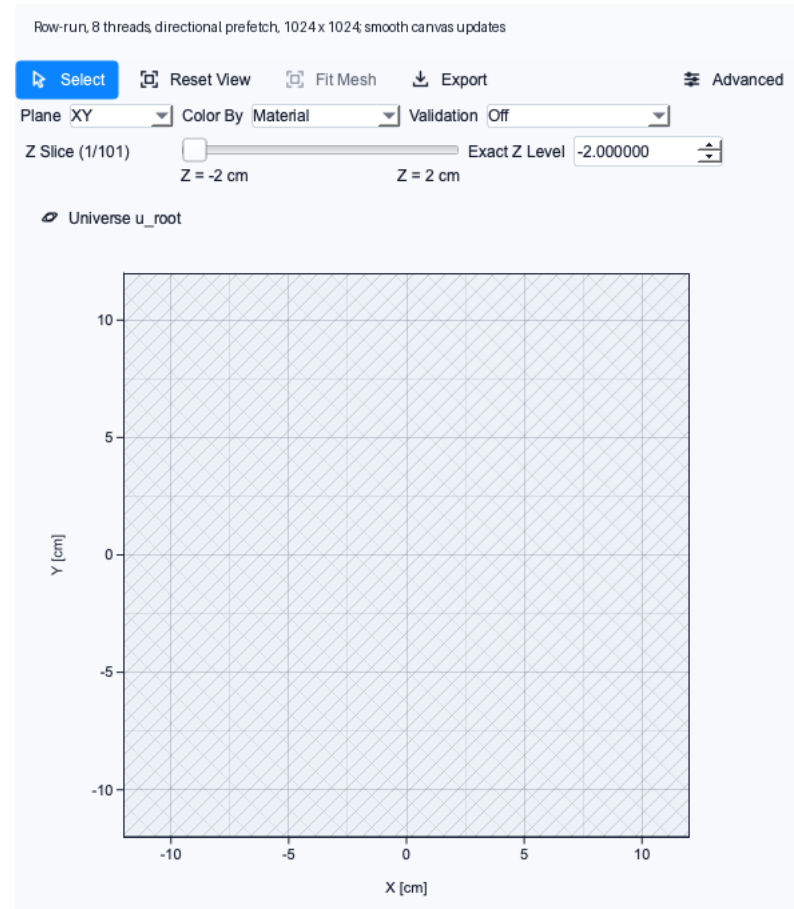
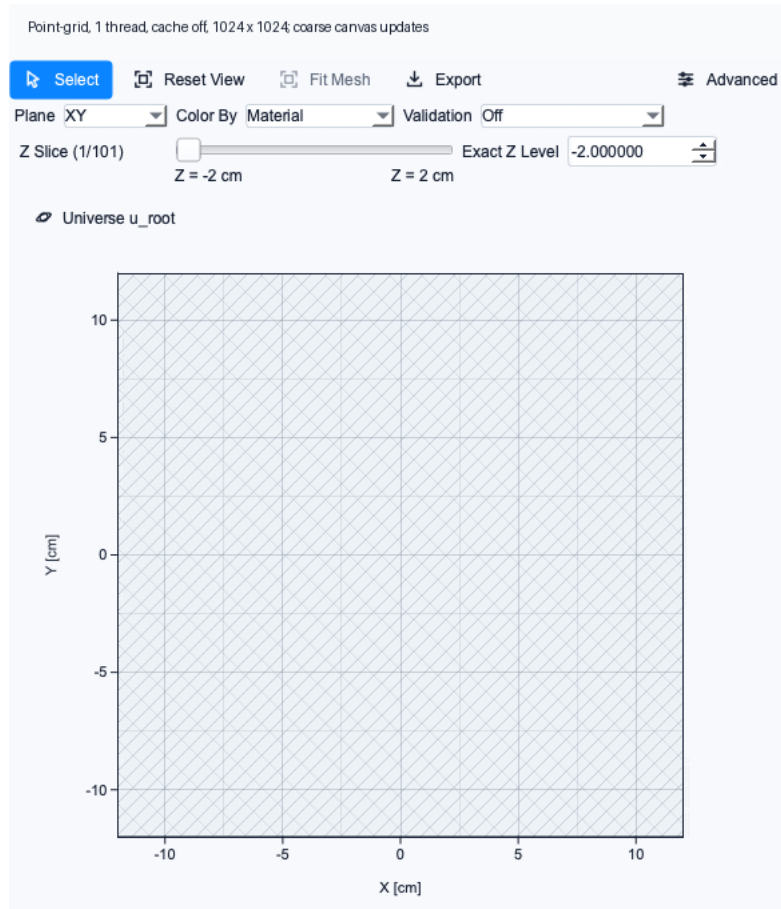
Dot = median; whisker = 5-95% interval.

### 3. 2D Slice-Stack CSG Visualization



# 3. 2D Slice-Stack CSG Visualization

- ❑ Demonstration of applying all optimizations on the nested sphere input.



## 4. Future Works

---

- Implement remaining planned modes: Run Manager, Results Viewer
- Migrate original iMC features to new iMC.
- Open source iMC (Github or OECD/NEA)

Thank You For Your Attention