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)

KAIST

Table of Contents

1. Introduction

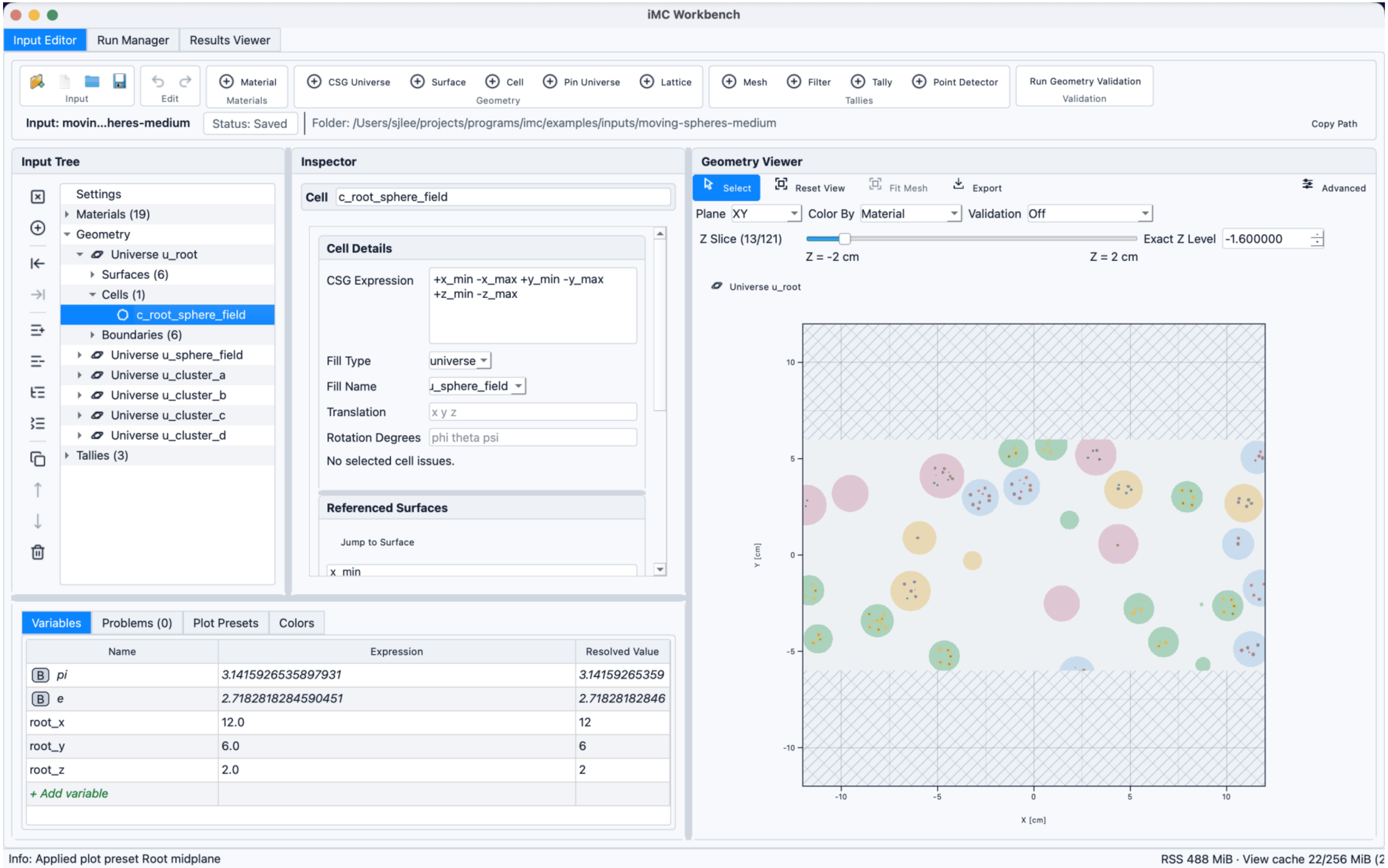
- iMC is a Monte Carlo particle transport code developed by the RPTL group at KAIST.
- It currently supports many advanced features:
  - Multi-group and continuous-energy energy modes
  - CSG geometry with nested universes and lattice support
  - MPI/OpenMP hybrid parallelization
  - Neutron-photon coupled 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

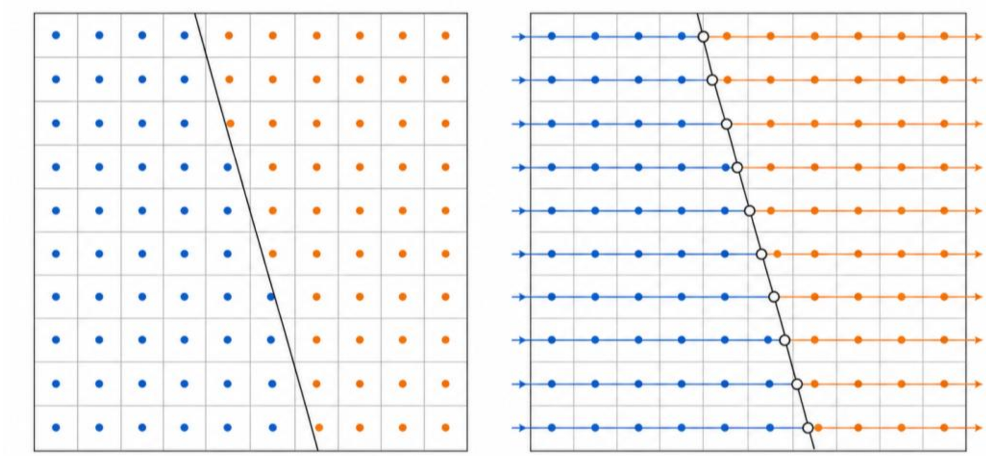
- iMC GUI is the new primary user interface, and is written in Python using the Qt 6 library.
  - It acts as a frontend for the iMC CLI, which has been rewritten to best support this (using Fortran OOP).
- It aims to act as a unified, integrated interface to support all workflows, and is structured around 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



3. 2D Slice-Stack CSG Visualization

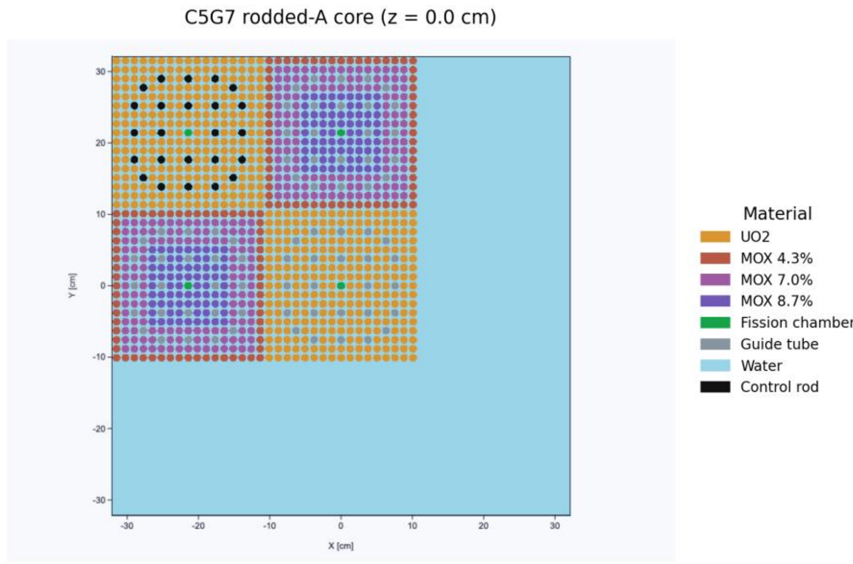
- Several optimization were introduced to make this possible.
  - Ray tracing-based “row-run” 2D slice generation (vs. point grid)

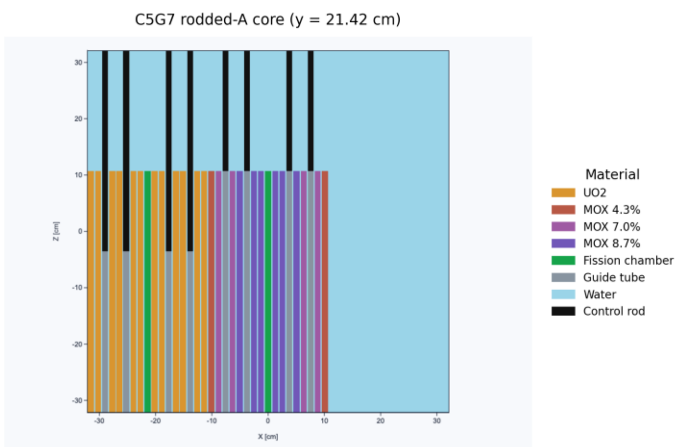


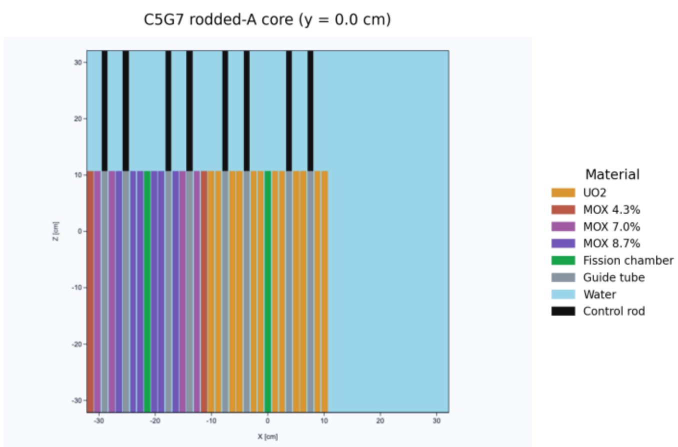
  - OpenMP parallelization
  - Slice prefetching and caching

3. 2D Slice-Stack CSG Visualization

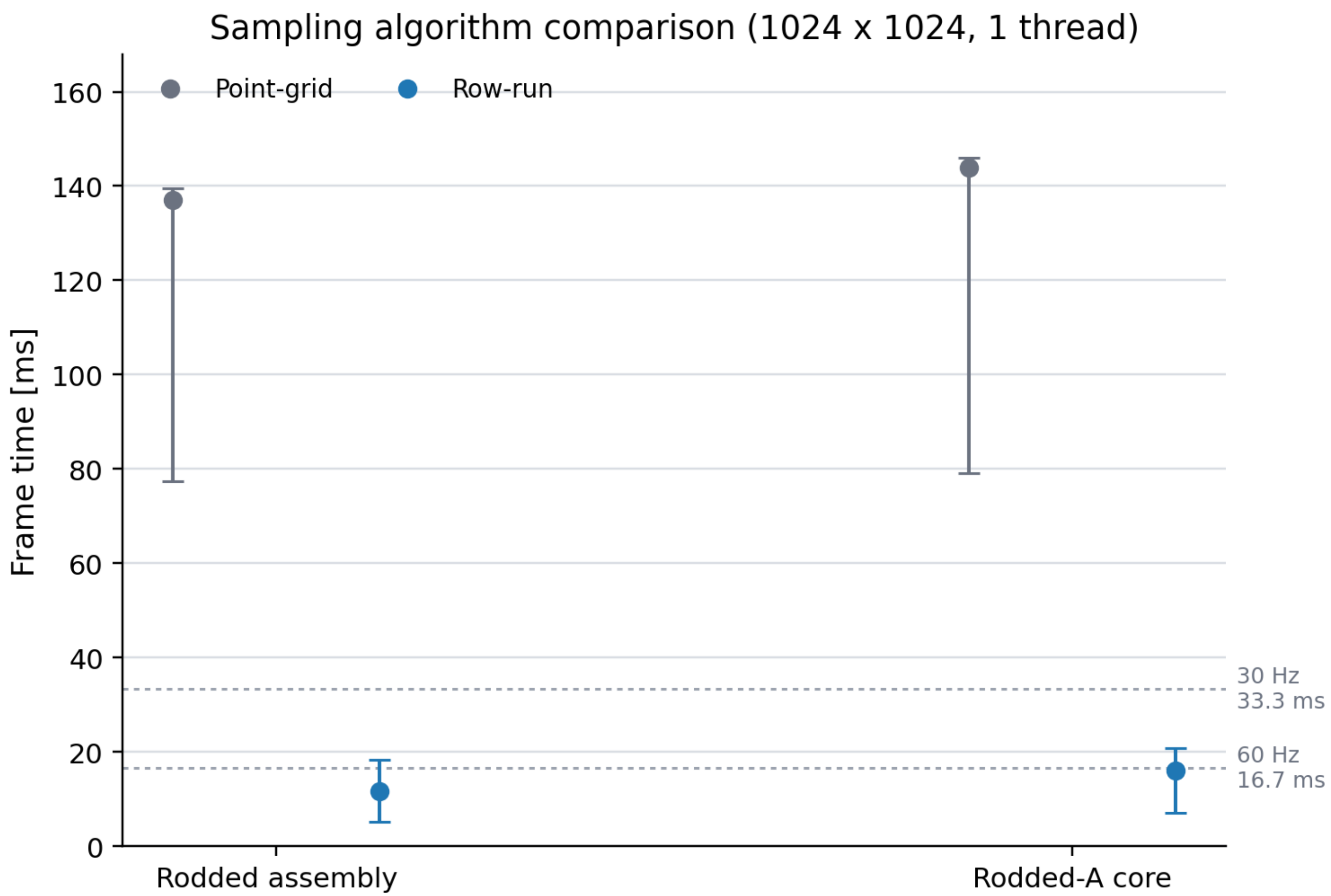
- ❑ The main test case is the OECD/NEA C5G7 MOX 3-D Extension Benchmark.
  - Rodded A configuration
  - 3 x 3 quarter-core radial layout
  - UO2, MOX, and reflector regions
  - 17 x 17 pin lattice
  - 1.26 cm pin pitch and 0.54 cm pin radius
  - 21.42 cm assembly pitch and 64.26 cm height
- ❑ ❑ The Rodded A case inserts control rods 1/3 into the inner UO2 assembly.







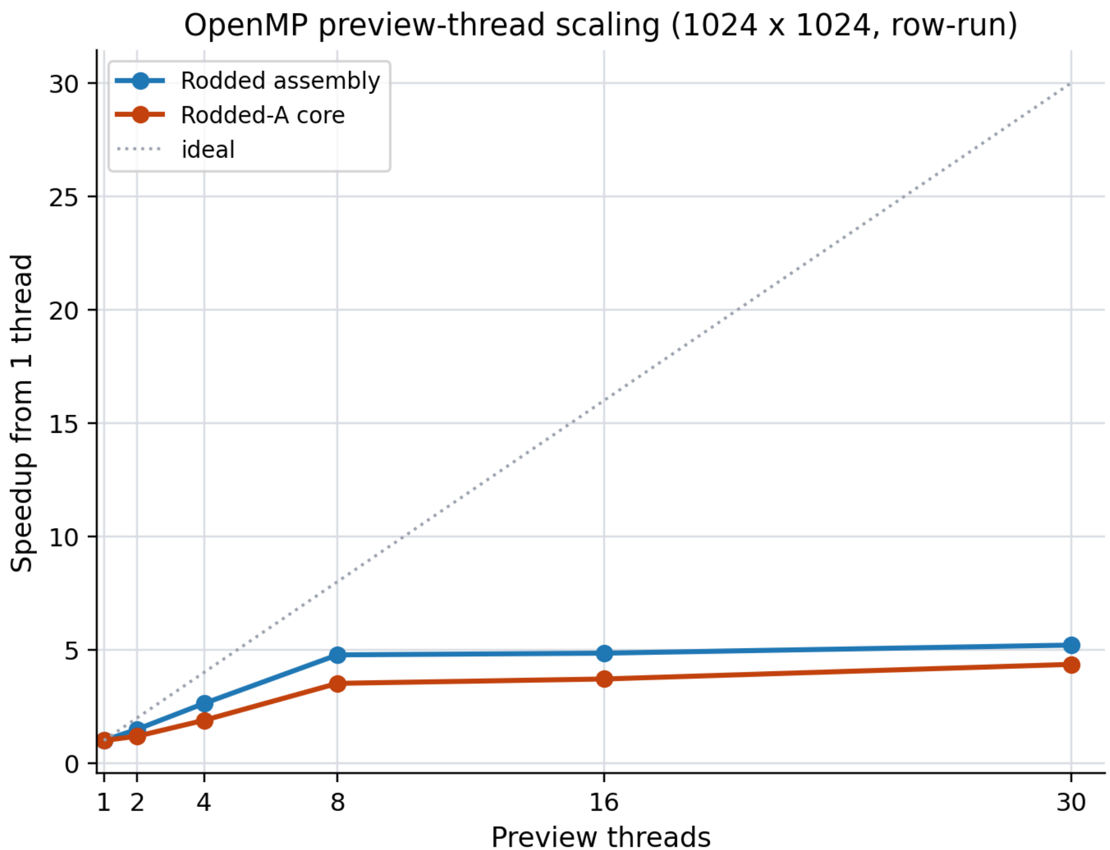
3. 2D Slice-Stack CSG Visualization



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

3. 2D Slice-Stack CSG Visualization

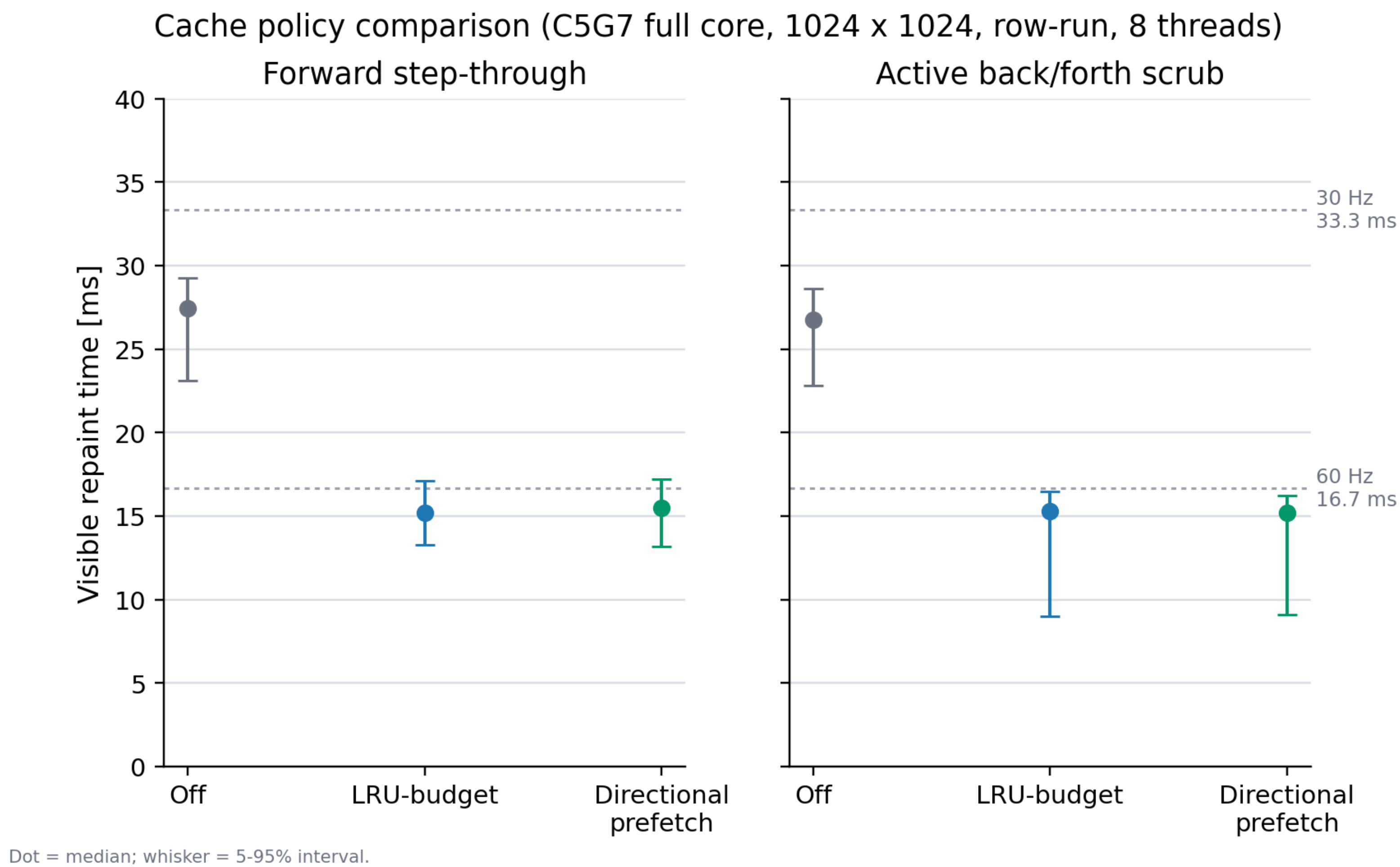
- ❑ The observed limit follows Amdahl's law.
  - ▪ The serial parts of the plotter module limits the speedup gain.
- ❑ ❑ Most practical gain is reached by 8 threads.



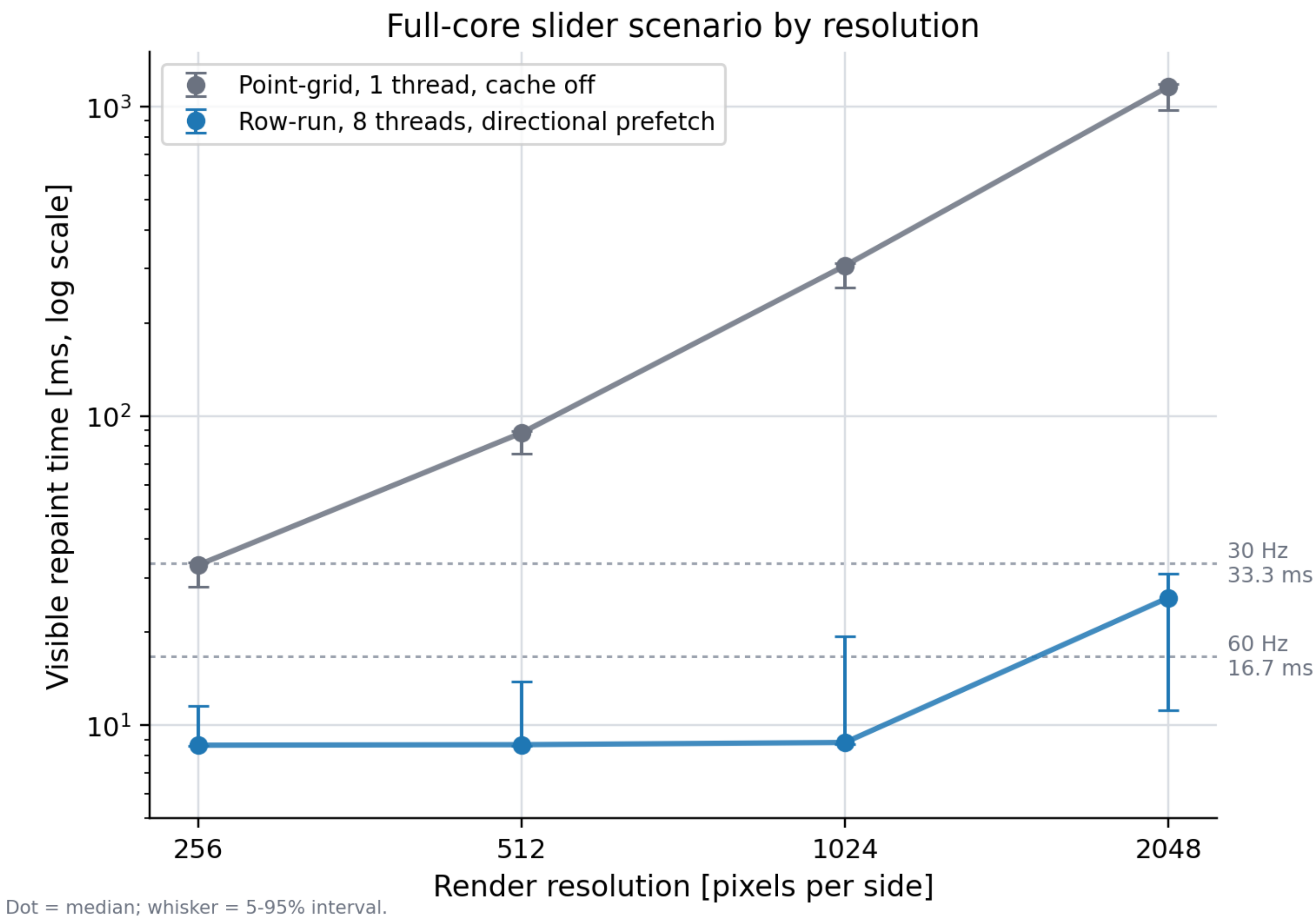
3. 2D Slice-Stack CSG Visualization

- 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

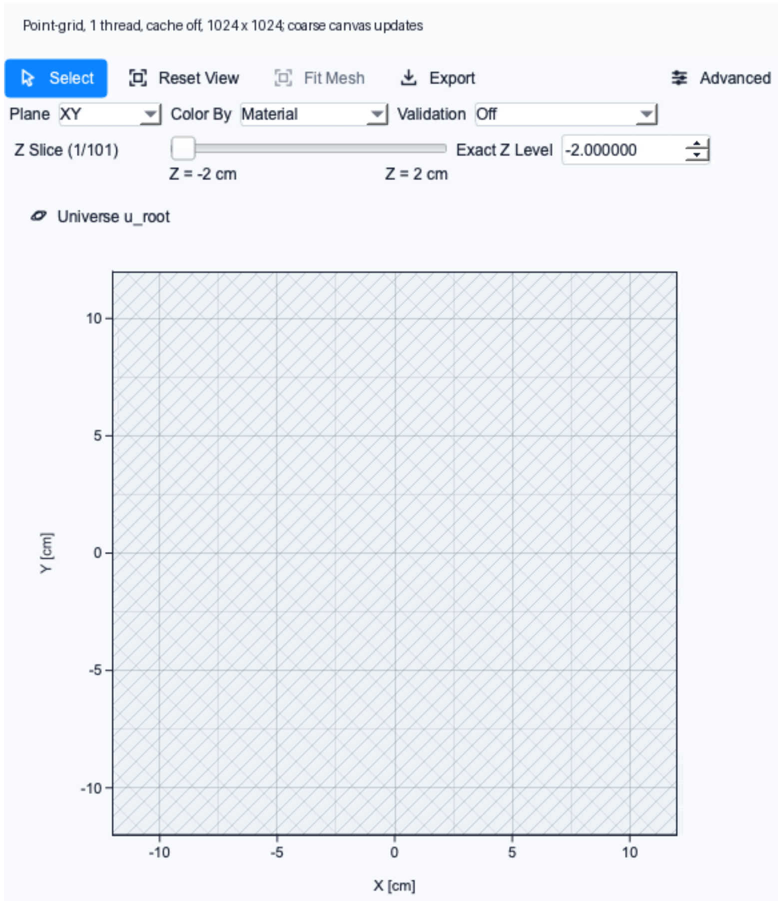


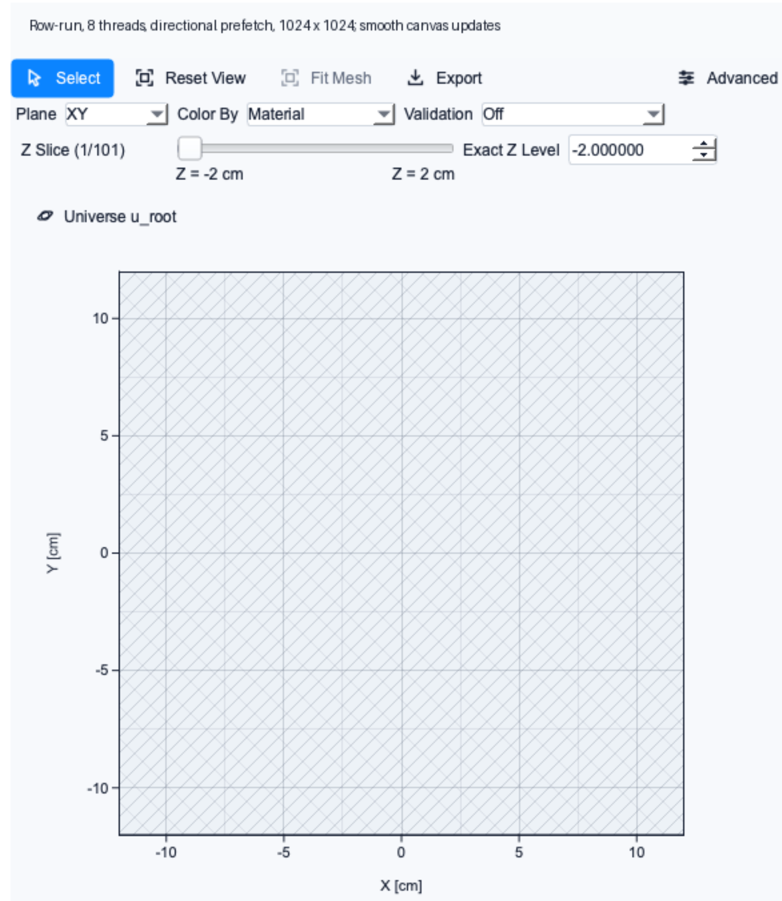
3. 2D Slice-Stack CSG Visualization



3. 2D Slice-Stack CSG Visualization

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





4. Future Works

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

Thank You For Your Attention