Update on the new HCP version developed by TUM

Chunyu Liu

Joint IAEA-ICTP Workshop on Reactor Physics, Thermal Hydraulics and Plant Design Engineering of Small Modular Reactors (SMR4212)

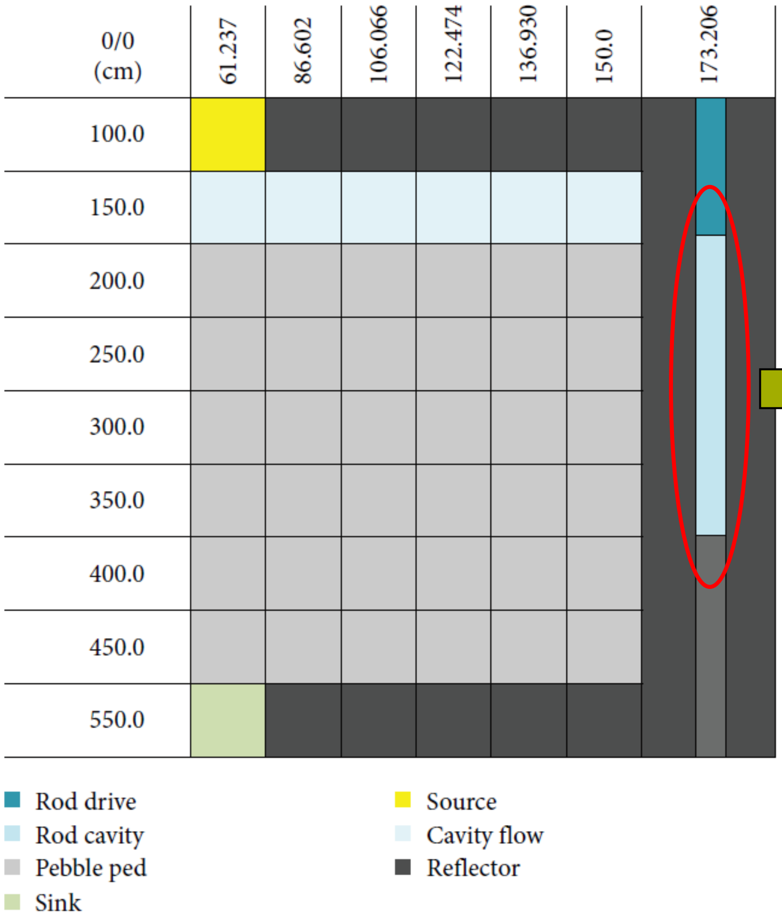
Trieste, Italy, 13 – 17 April 2026

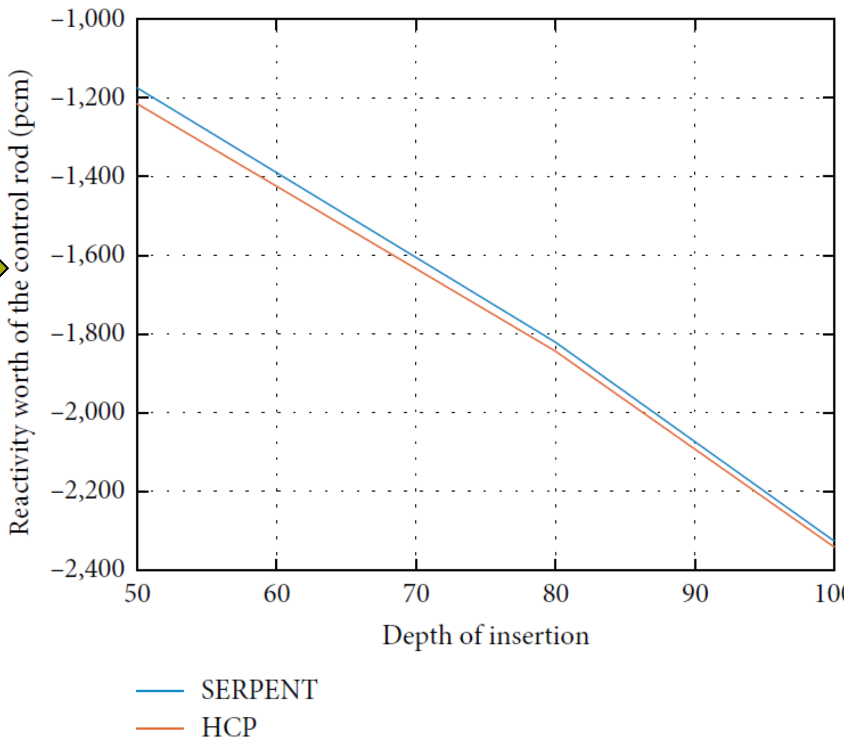
Overview

- Progress in the research project KONHCP I
  - Implementation of control rod
  - IT work
- Outlook for the research project KONHCP II
  - Implementation of molten salt coolant
  - Implementation of prismatic geometry
  - Further integration of the STACY module
- Advances in Independent Research
  - Conductive heat transfer calculation
  - LibGen fixing and I/O improvement

Implementation of control rod

Verification



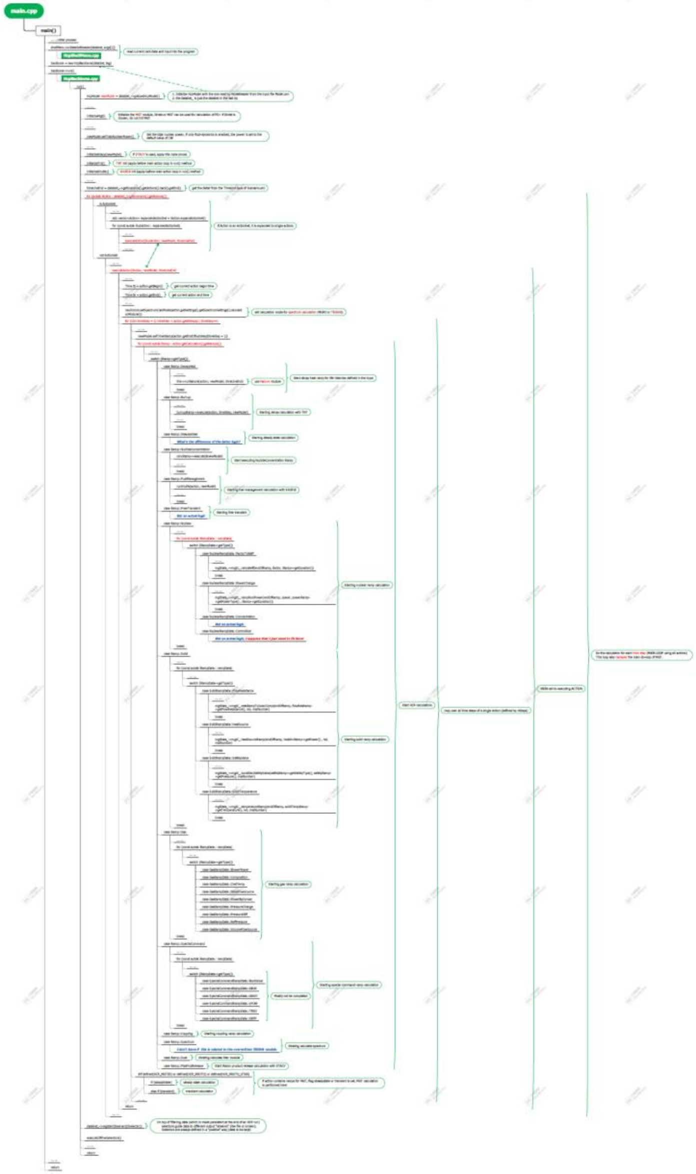


The main code structure of HCP

main

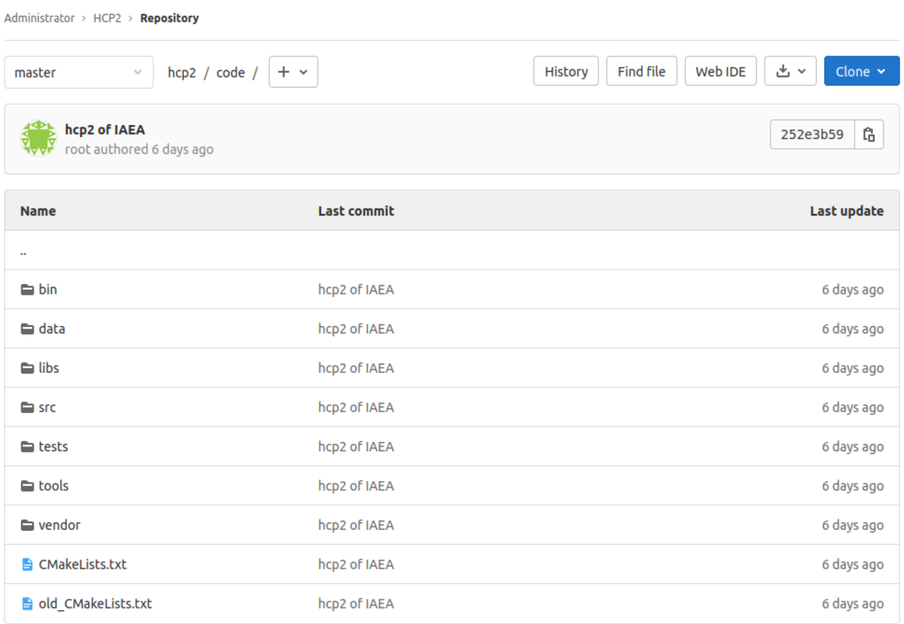
- ...initial process
- shellMenu.runDataSetReader(dataSet, argv[1])
  - read current card data and input into the program
- ...other process
- backbone->run()
  - process MGT
  - process STACY
  - process TNT
  - process SHUFLE
  - execute Actions in loop
  - execute Selectors about output
- ...other process
- return

IT work

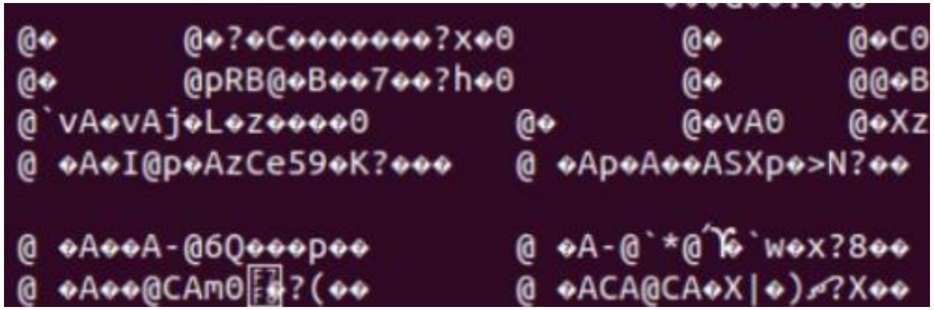


IT work

- Construction of project hosting warehouse based on Gitlab



- Locate some memory leak points as shown below

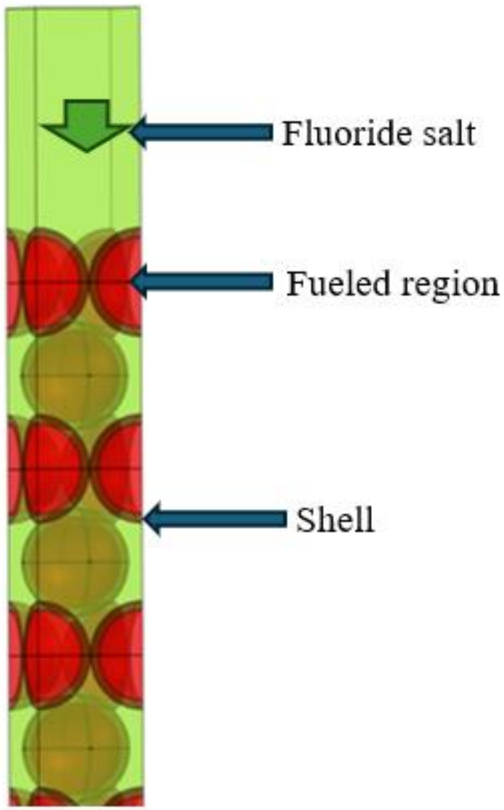


KONHCP II

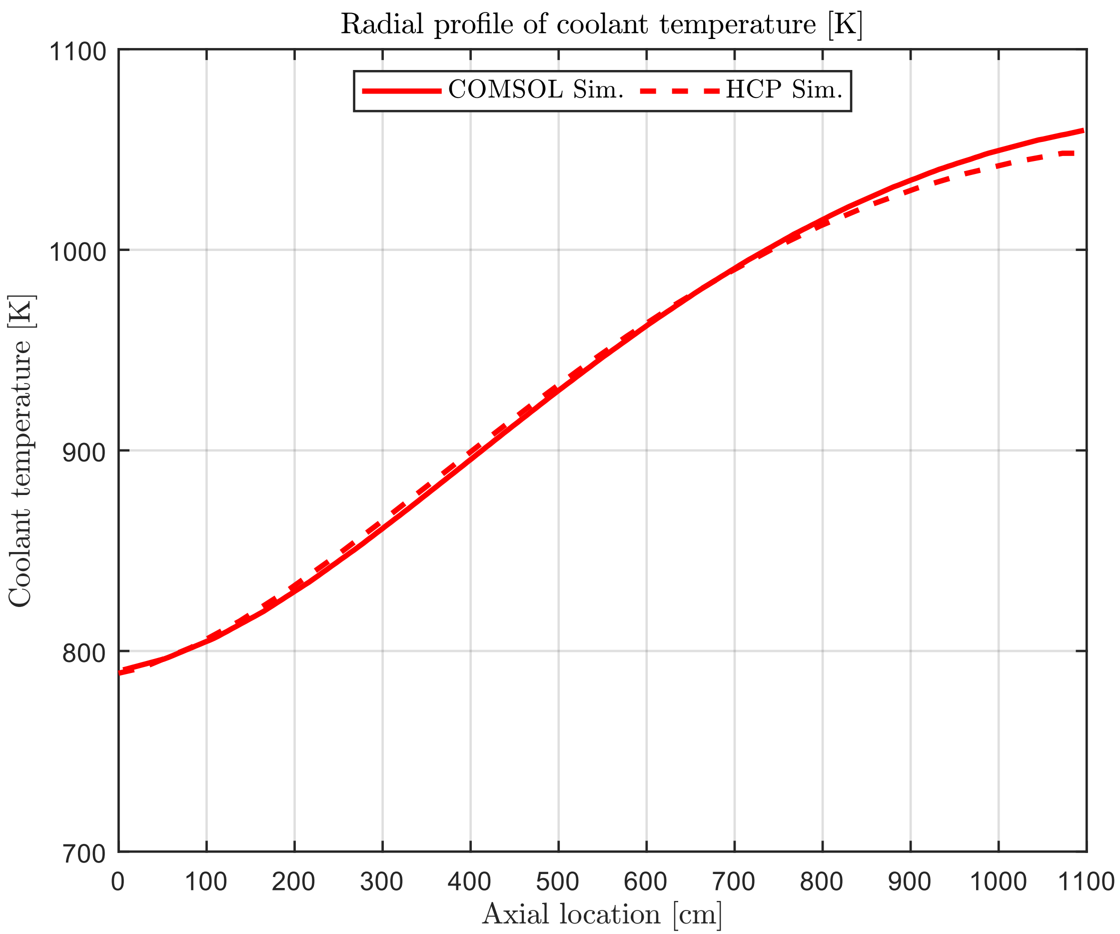
- **Expected duration**: 2026/03/01 – 2029/08/31
- **Partners**:
  - Technical University of Munich (2 PhD positions)
  - Becker Technologies GmbH (1 PhD position)
- **Work packages**:
  - **WP1**: Development of HCP to support prismatic-type HTRs.
  - **WP2**: Extension of HCP to integrate molten salt as an additional coolant option.
  - **WP3**: Further development and expansion of the source-term calculation module STACY and integration of the STAR module.
  - **WP4**: Harmonization of code modifications and expansion of the code documentation.

Implementation of molten salt coolant

Benchmark: coolant temperature distribution in hot channel (SS)

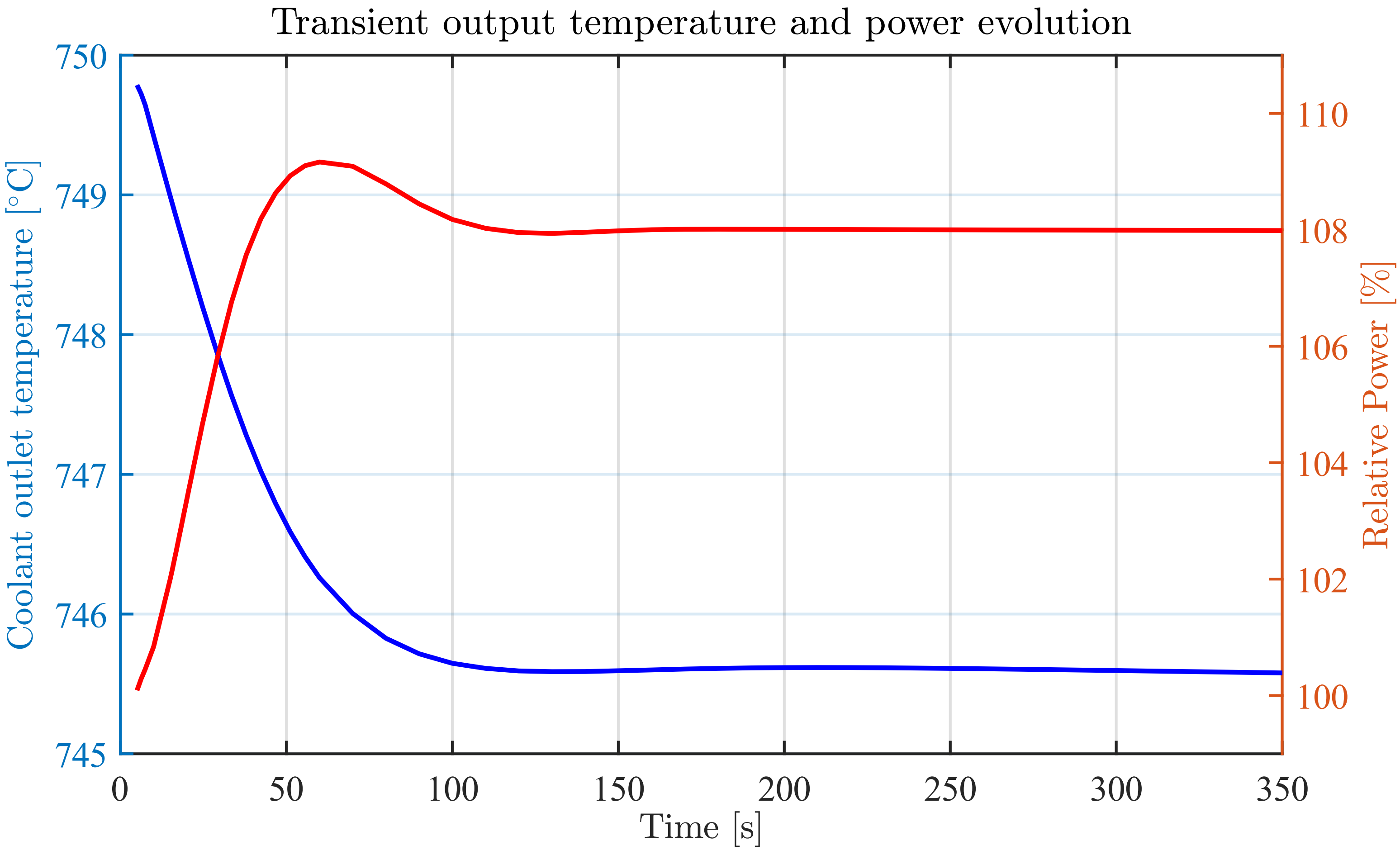


COMSOL model



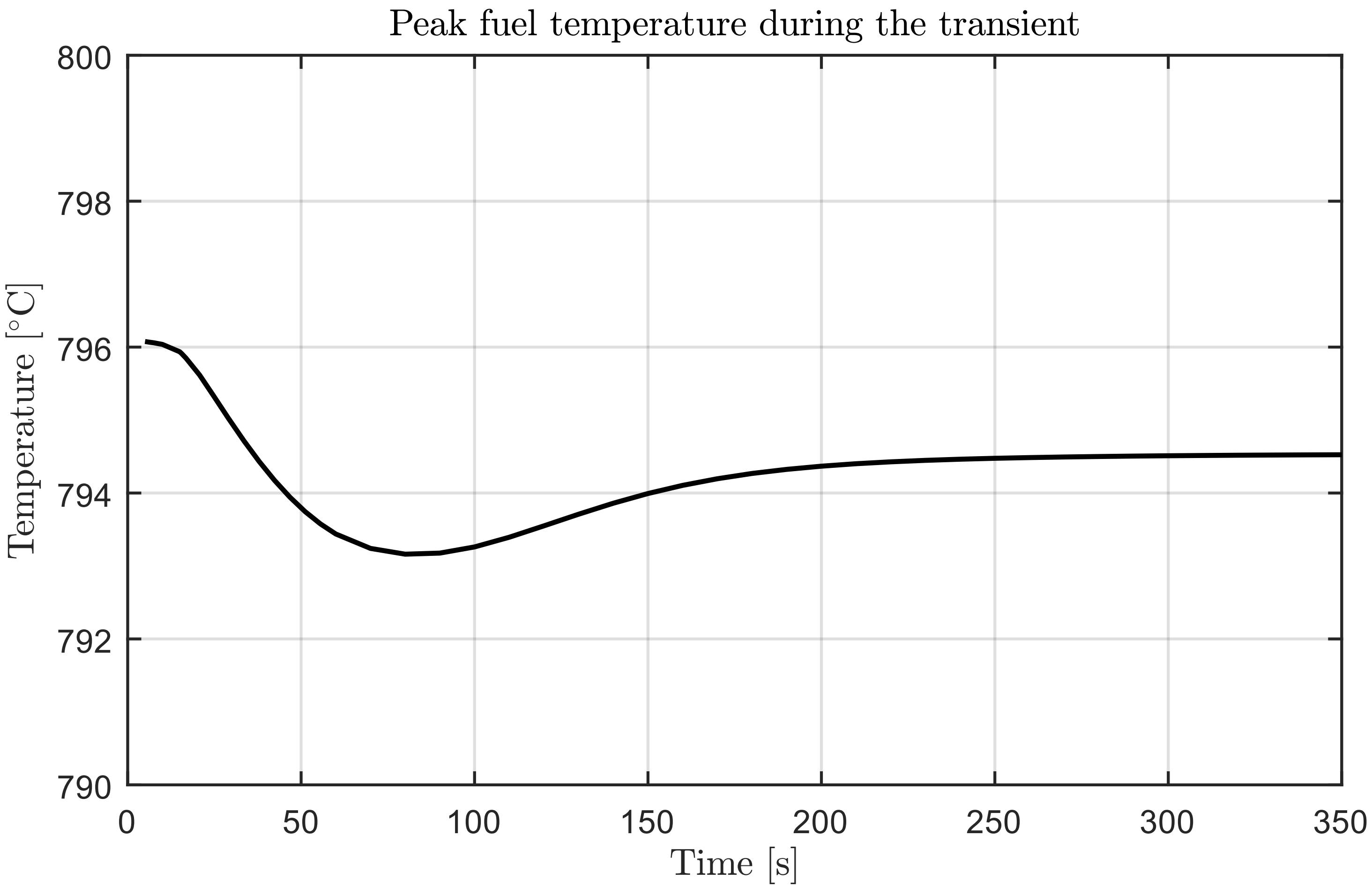
Implementation of molten salt coolant

Transient of coolant mass flow rate variation: +10% in 10 seconds



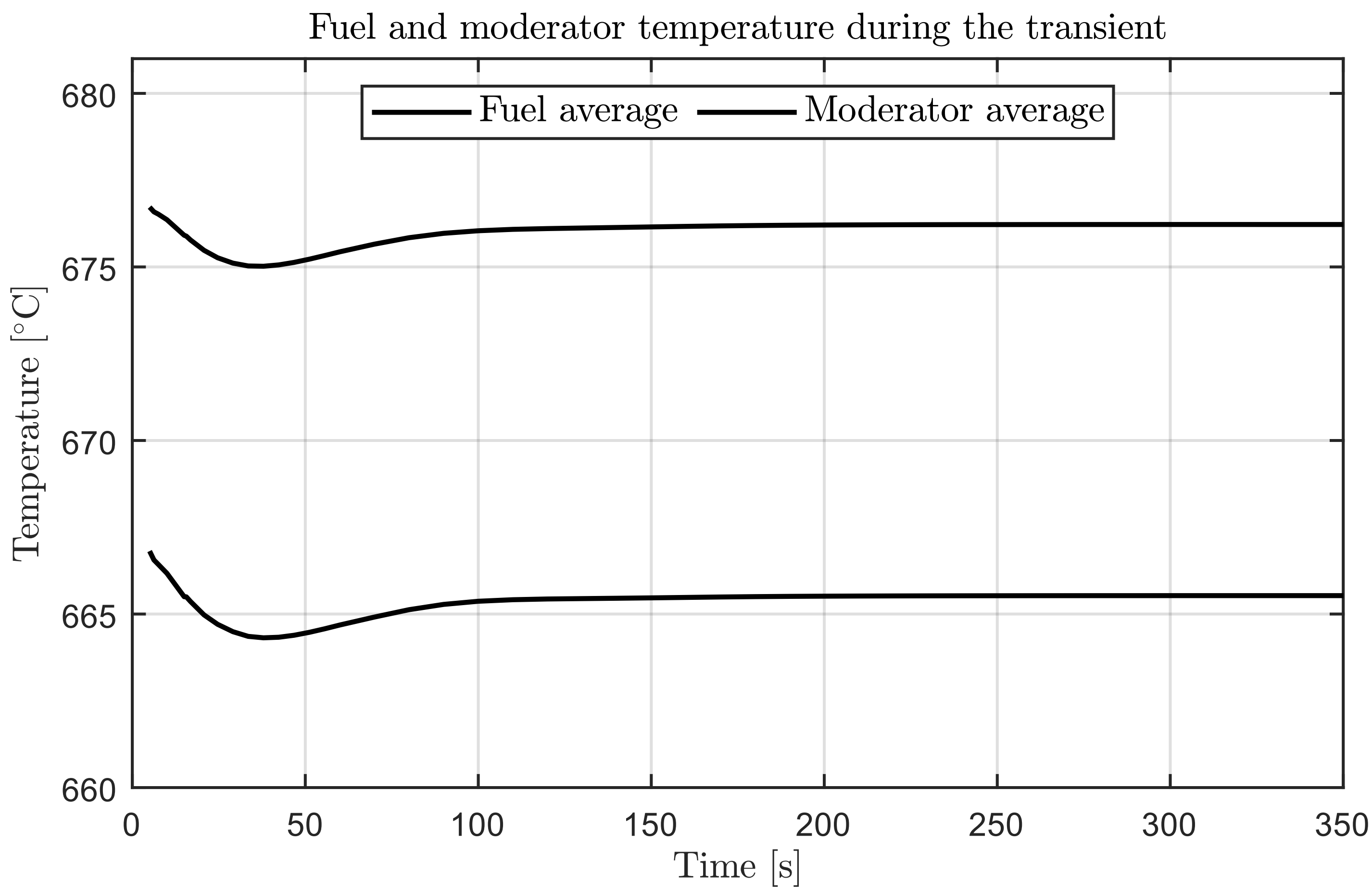
Implementation of molten salt coolant

Transient of coolant mass flow rate variation: +10% in 10 seconds



Implementation of molten salt coolant

Transient of coolant mass flow rate variation: +10% in 10 seconds



Implementation of prismatic geometry

1. Considerations for geometry
   - Input reading: “prisBlock” already definded in ModelReader.cpp, but nuclide vectors cannot be assigned (only spherical supported for now), extension needed
   - Class definition: extend the existing classes to include the prismatic block structure and hexagonal mesh (datamodel -> fuel element related files)
   - Material properties: already defined in “SpecificHeat.f90”
   - Transfer hexagonal meshes from backbone to MGT directly, high accuracy, but solver has to be adjusted
   - <u>Trick</u>: implement a subroutine in backbone to map prismatic blocks between hexagonal meshes and cylindrical meshes, MGT will only see cylindrical meshes and do calculation as usual

Implementation of prismatic geometry

2. Considerations for neutronics calculation
   - Governing equations unchanged
   - Multiple fuel/coolant channels in one block: homogenization
   - Spatial discretization of control volumes has to be adjusted for hexagonal mesh: FDM -> FEM?
   - Special attention for irregular outer boundary
3. Considerations for thermal-hydraulics calculation
   - Governing equations unchanged
   - Convective heat transfer in coolant channels, conductive heat transfer between blocks
   - Dedicated constitutive models have to be added
   - Same issue for discretization

Implementation of prismatic geometry

4. Considerations for burnup calculation
   - Take batch-wise nuclide inventory and flux distribution directly from backbone
   - No modification expected
5. Considerations for control rod
   - SHUFLE module for control rod movement, movable solid volume inside the prismatic block (as fuel element)
   - Maybe a little modification for the fuel element volume division and a correction factor for coolant channels
6. Considerations for nuclear data library
   - Different correction methods for geometric heterogeneities
   - Some influences on the accuracy, but not urgent

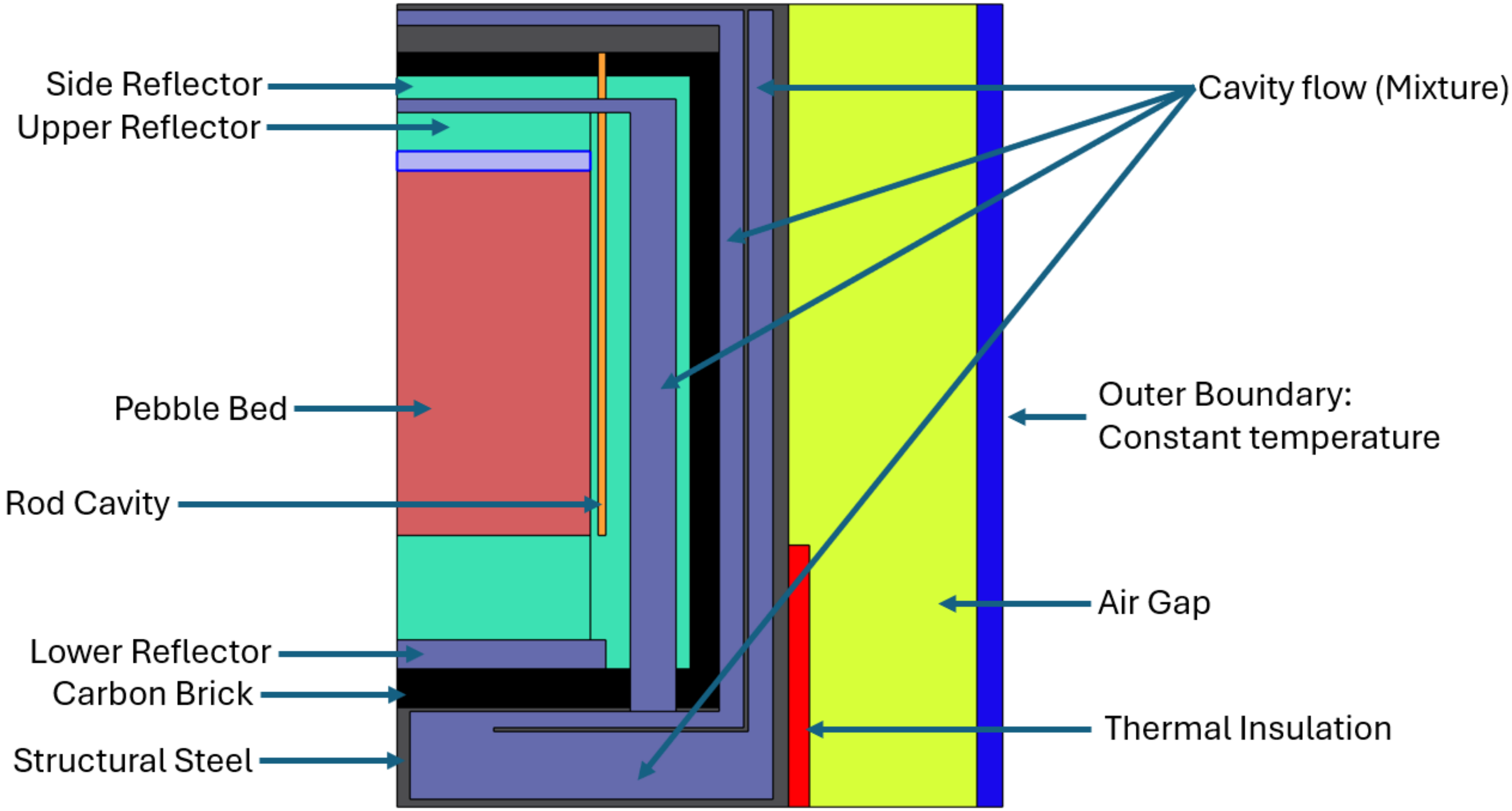
Further integration of the STACY module

This part will be conducted at **Becker Technologies**.

- Continuation of the integration of the source term calculation module STACY
- Extension of the STACY module to the prismatic core
- Integration of the STAR module

Conductive heat transfer calculation

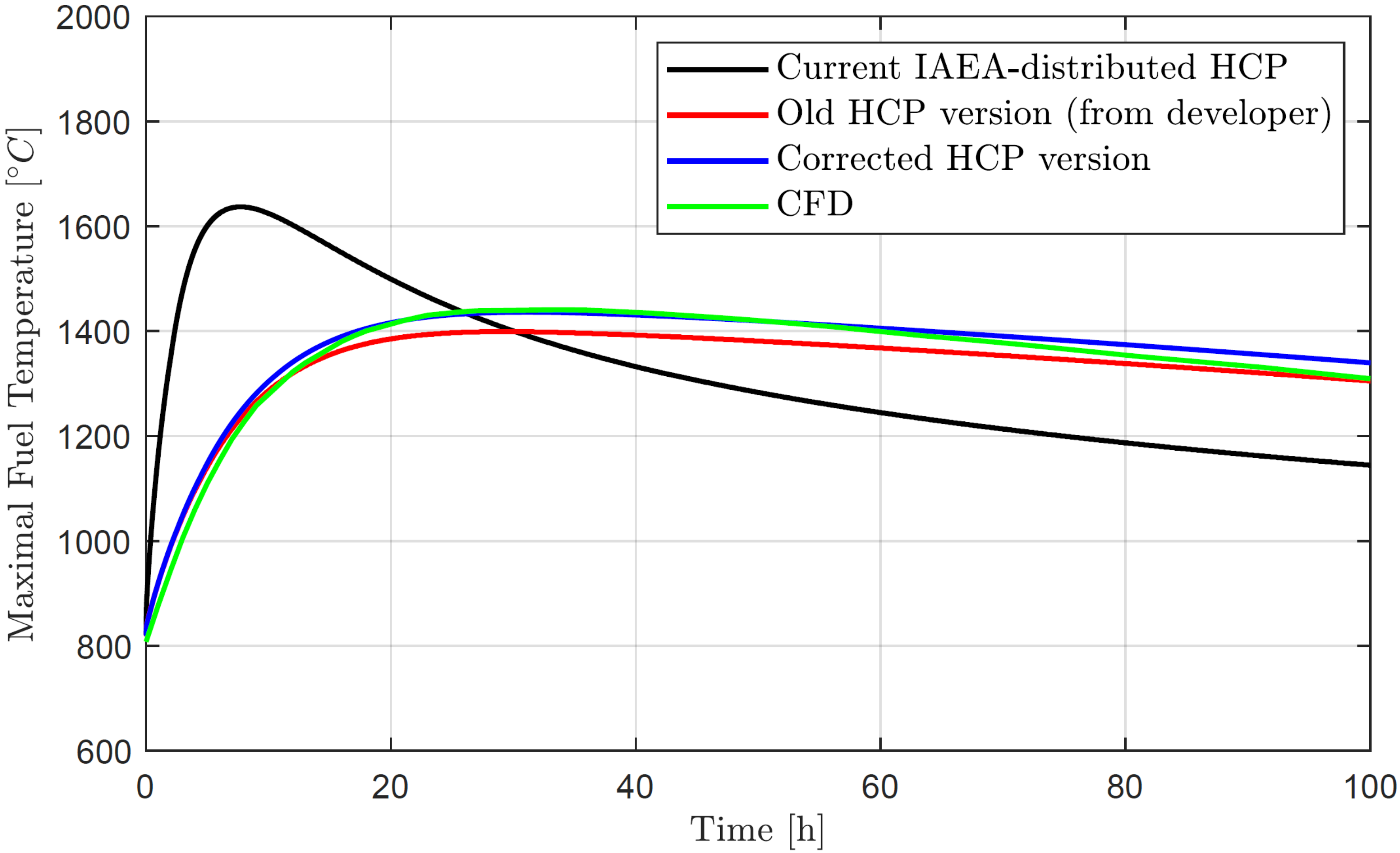
DLOFC transient: HTR-MODUL-200



COMSOL model

Conductive heat transfer calculation

DLOFC transient: HTR-MODUL-200

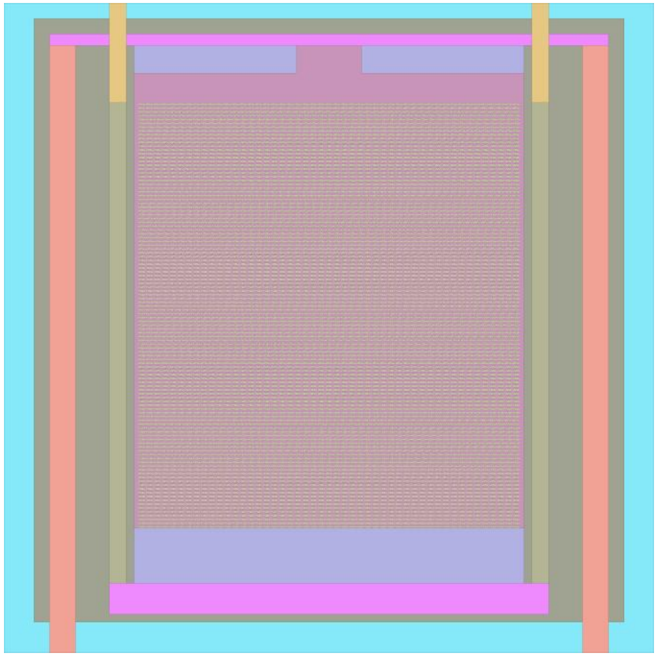


LibGen fixing

No input folders: reverse-engineered

Error in subroutine “readFissionSpectrumData”: rewritten

Several libraries generated and benchmarked



Serpent model

HTR-PM keff values calculated using different nuclear data libraries

Coolant	Library	HCP	Serpent	Relative error [%]
Helium	ENDF/B-VII.0 (coarse)	1.34254	1.33417	0.627
	ENDF/B-VII.0 (fine)	1.34198	1.33417	0.585
	ENDF/B-VII.1 (fine)	1.33220	1.32555	0.502
	ENDF/B-VIII.0 (fine)	1.33792	1.32618	0.885
Air	ENDF/B-VII.0 (coarse)	1.34191	1.33041	0.864
	ENDF/B-VII.0 (fine)	1.34134	1.33041	0.822
	ENDF/B-VII.1 (fine)	1.33157	1.32236	0.694
	ENDF/B-VIII.0 (fine)	1.33730	1.32263	1.109

I/O improvement

- Selector -> “Restart”, but power history missing -> flaw in decay heat calculation! -> external preprocessing code: add power history items automatically
- “Restart.xml” very large -> not all the nuclides are crucial for reactor dynamics (maybe important for STACY) -> external preprocessing code: filter to remove low density nuclides -> 126MB to 3MB without losing accuracy, faster
- No output for burnup calculation -> several functions in subroutine “TntOutputWriter” rewritten -> output: time-dependent batch-wise nuclide inventory
- Very limited transient data in “output.txt” -> binary file “data.ptr” -> data extraction through reverse-engineering -> but only 1D data -> extended to include time-dependent 3D data

Thanks!