Multi-scale Simulation of NuScale-like SMR with PARCS/TWOPORFLOW/TRACE

A. Campos-Muñoz, V. Sanchez-Espinoza

Institute for Neutron Physics and Reactor Technology

Content

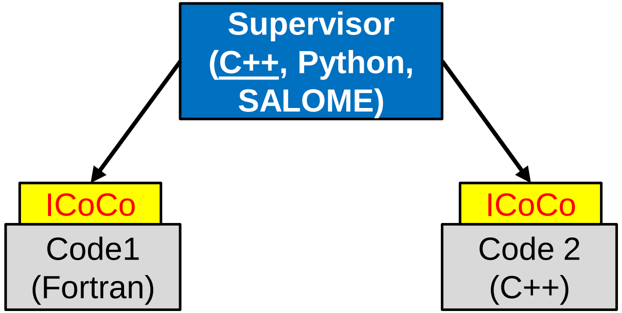
- Introduction to ICoCo
- ICoCo implementation
- TwoPorFlow code
- PARCS/TPF/TRACE steady-state coupling
- NuScale-like SMR description
- First results
- Open Issues
- Remarks and Future Work

Introduction to ICoCo

Karlsruhe Institute of Technology

- Interface for Code Coupling (ICoCo): provides a standard **frame** for code coupling.

Define methods for:
- Initialization and termination
- Time advance
- Save and restore state
- Getting and setting fields



- **Code split** in funtional pieces.
- ICoCo framework MED format **mesh** is compulsory.
- Inherently bound with **MEDCoupling** library.

ICoCo implementation (example)

Karlsruhe Institute of Technology

ParcsProblem.cxx

ICoCo method

```
/* Perform a step in a transient calculation: */
bool ParcsProblem::solveTimeStep() {

   /* Perform the step with PARCS: */
   int conv;
   if (!burnup_finished) {
      parcs_solve_burnup_step_(&conv, &error);
      if (!checkError("Error in parcs_solve_burnup_step_()!", false))
         return false;
   }
   else {
      parcs_solve_time_step_(&conv, &error);
      if (!checkError("Error in parcs_solve_time_step_()!", false))
         return false;
   }

   /* Return success: */
   return true;

};
```

ICoCo method call

Intermidiate function call

parcs_solve_time_step.f90

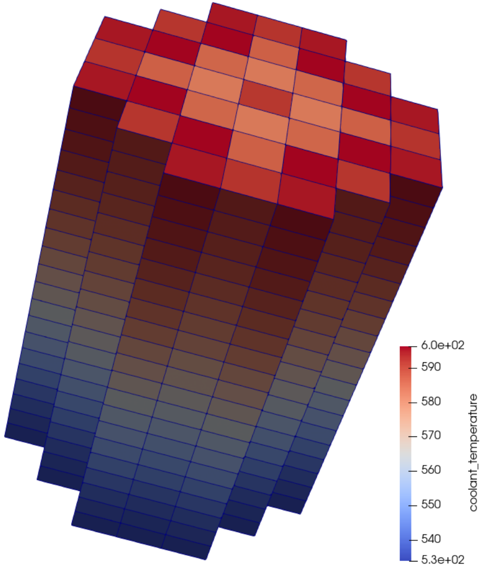
Fortran function

```
SUBROUTINE parcs_solve_time_step(converged, lib_error)

      USE CntlM, ONLY: error, tran
      USE GlobM, ONLY: ferror
      USE TransDriveMe, ONLY: transient

      USE iso_c_binding, ONLY: c_int

      IMPLICIT NONE

      INTEGER(c_int), INTENT(OUT) :: converged, lib_error

      ! WRITE(0, *) 'parcs_solve_time_step'

      converged = 1
      lib_error = 0

      IF (error) ferror = .TRUE.

      IF (tran .AND. (.NOT. error)) CALL transient(.FALSE., .FALSE., .TRUE., .FALSE.)

      IF (error .OR. ferror) lib_error = 1

END SUBROUTINE parcs_solve_time_step
```

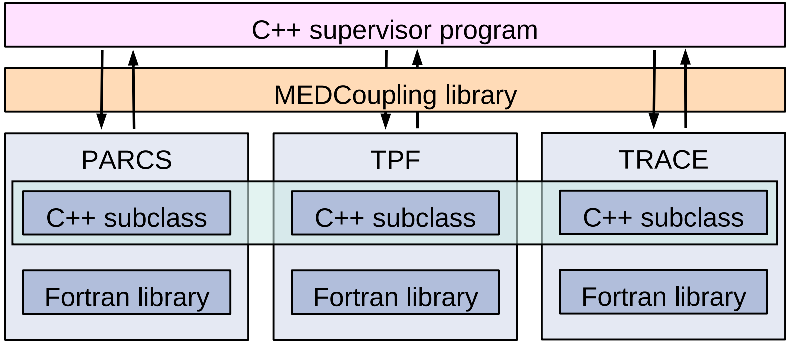
PARCS function call

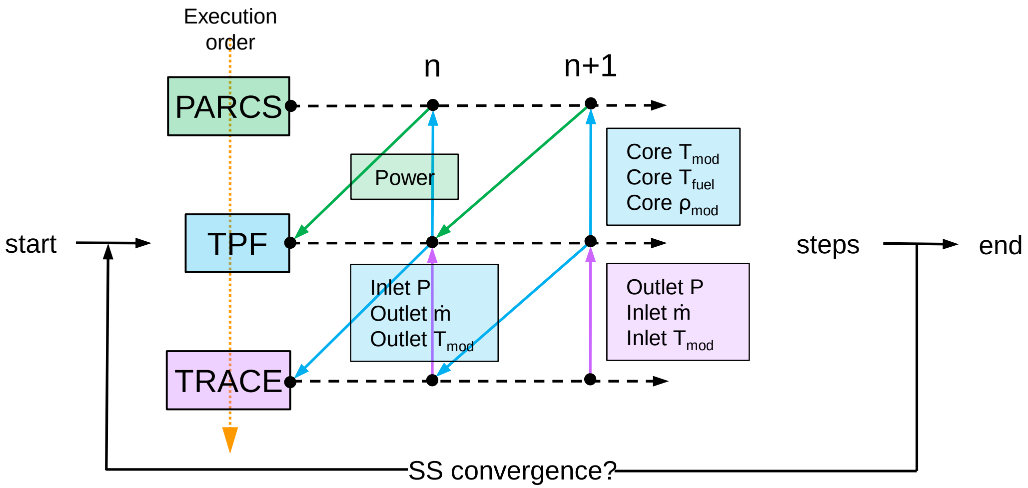
TwoPorFlow code

- Porus-media (FAVOR technique)
- Steady-state and transient solution
- Two-phase flow (6 equations)
- 3D conservation equations
- 2D heat conduction model for fuel rods
- Coarse Cartesian grids

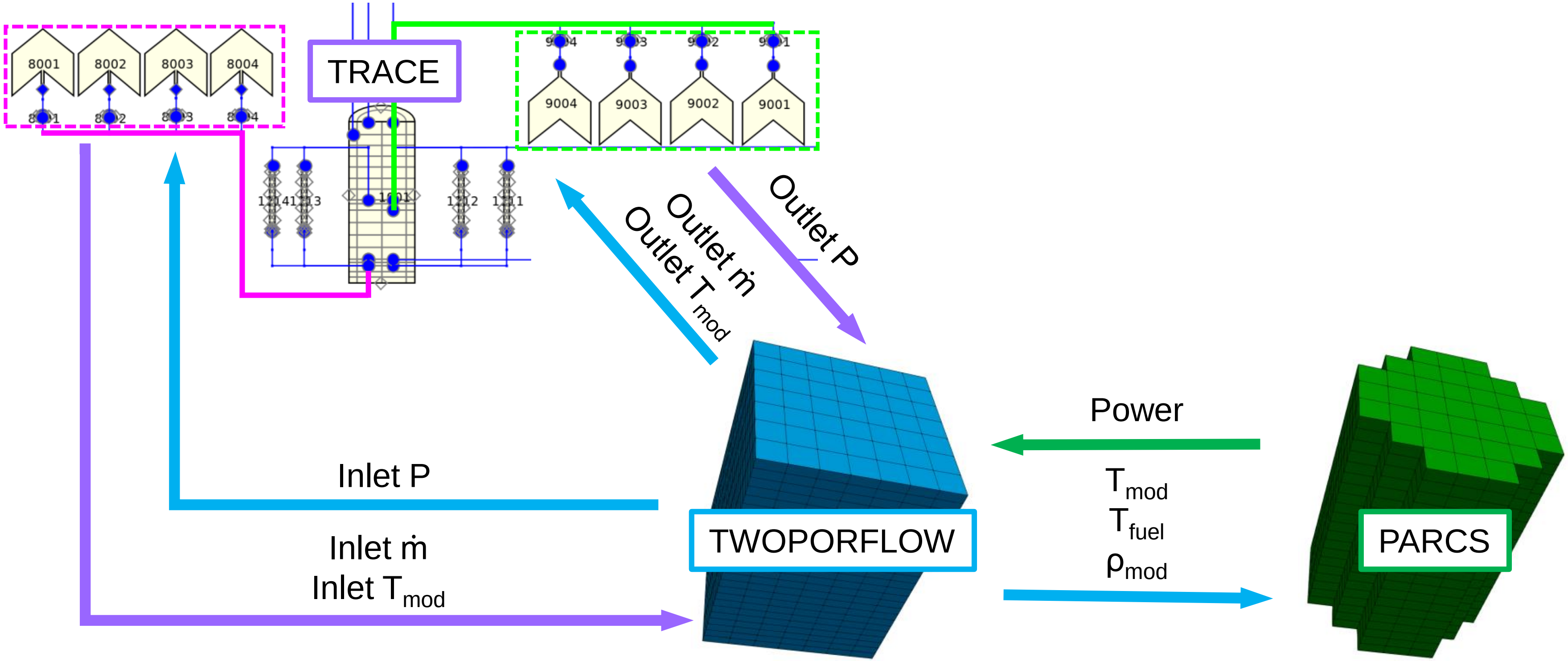


PARCS/TPF/TRACE steady-state coupling





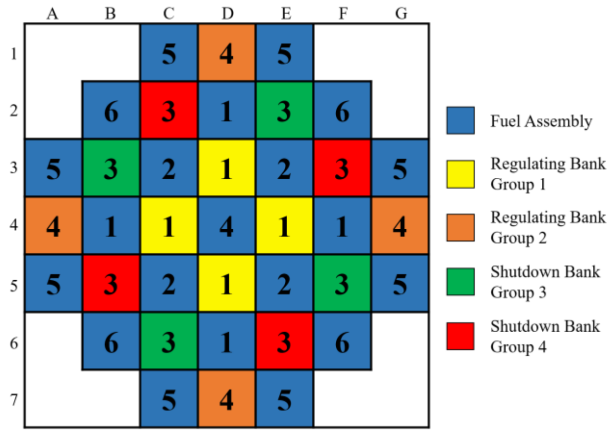
PARCS/TPF/TRACE steady-state coupling

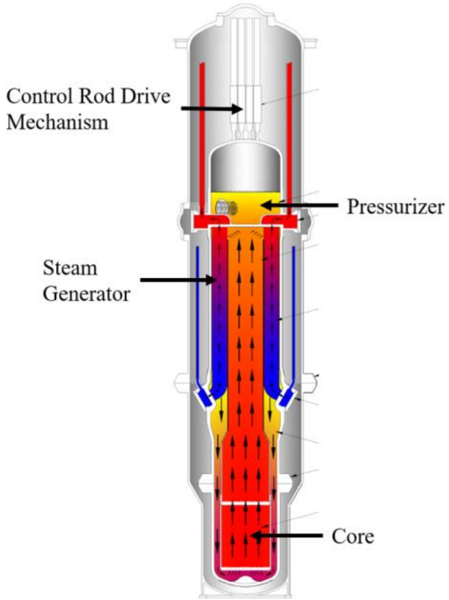


NuScale-like SMR description

Karlsruhe Institute of Technology

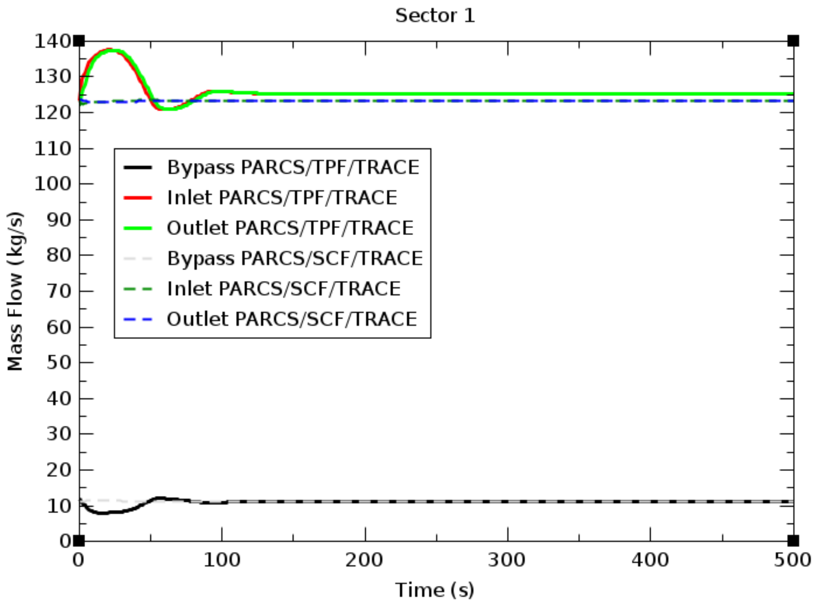
Parameter	Value
Total power	160 MW_{th}
System pressure	12.755 MPa
Core avg temperature	563.71 K
Core mass flow	496.17 kg/s





First results: NuScale-like Steady-State

PARCS/TPF/TRACE coupling assumptions:
TRACE→TPF: P_outlet: fixed | ṁ_inlet, tl_inlet: interp
TPF→TRACE: P_inlet, tl_outlet: fixed | ṁ_outlet: interp

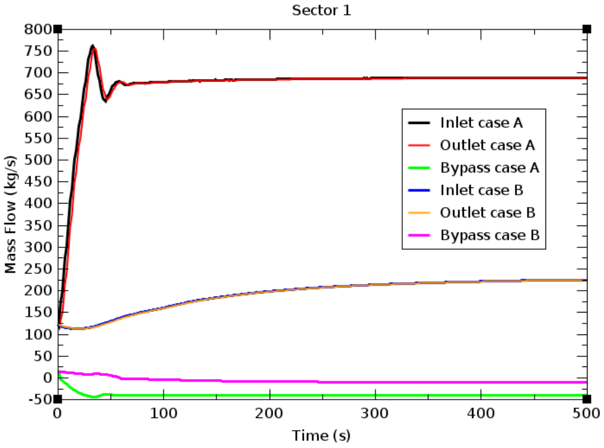


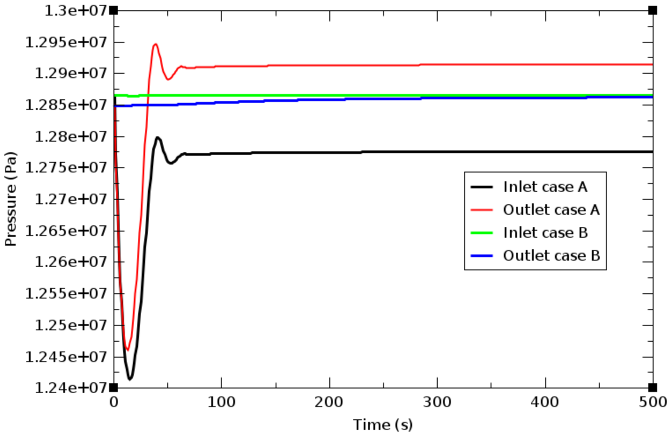
Parameter	**Reference	*Values (error %)
Total power [MW_{th}]	160.0	160.0 (0.0)
Inlet pressure [MPa]	12.2864	12.2864 (0.0)
Outlet pressure [MPa]	12.2864	12.2864 (0.0)
Core avg temperatura [K]	560.0	562.0 (0.3)
Core mass flow [kg/s]	492.8	500.8 (1.6)

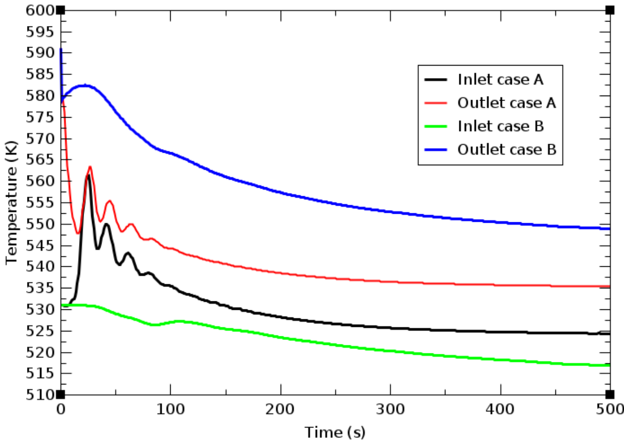
**Reference: PARCS/SCF/TRACE
*Values: PARCS/TPF/TRACE

Open issues

- *Case A: all variables interpolated*
- *Case B: just pressure fixed at inlet and outlet*







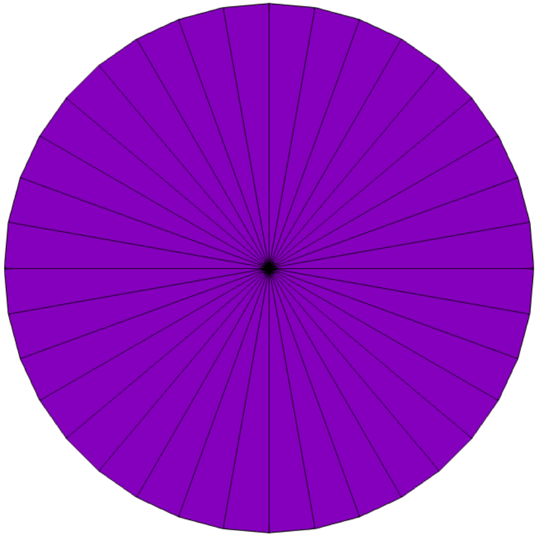
Remarks and Future Work

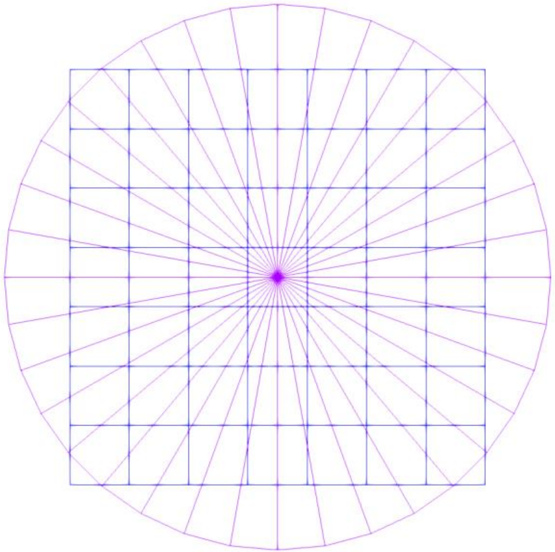
- Successful coupling of PARCS, TWOPORFLOW, TRACE based on ICoCo.
- Assumptions on exchange variables must be taken to get steady-state solution.
- When all variables are interpolated, oscillations and wrong convergence are presented.

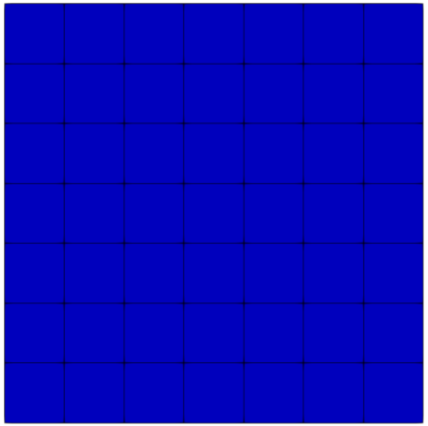
- Next steps:
  - Solve oscillations through interpolation criteria, for example.
  - Analyse transients at RPV and plant level, e.g. MSLB.

Extra Slides

Boundary Interpolation TRACE/TPF







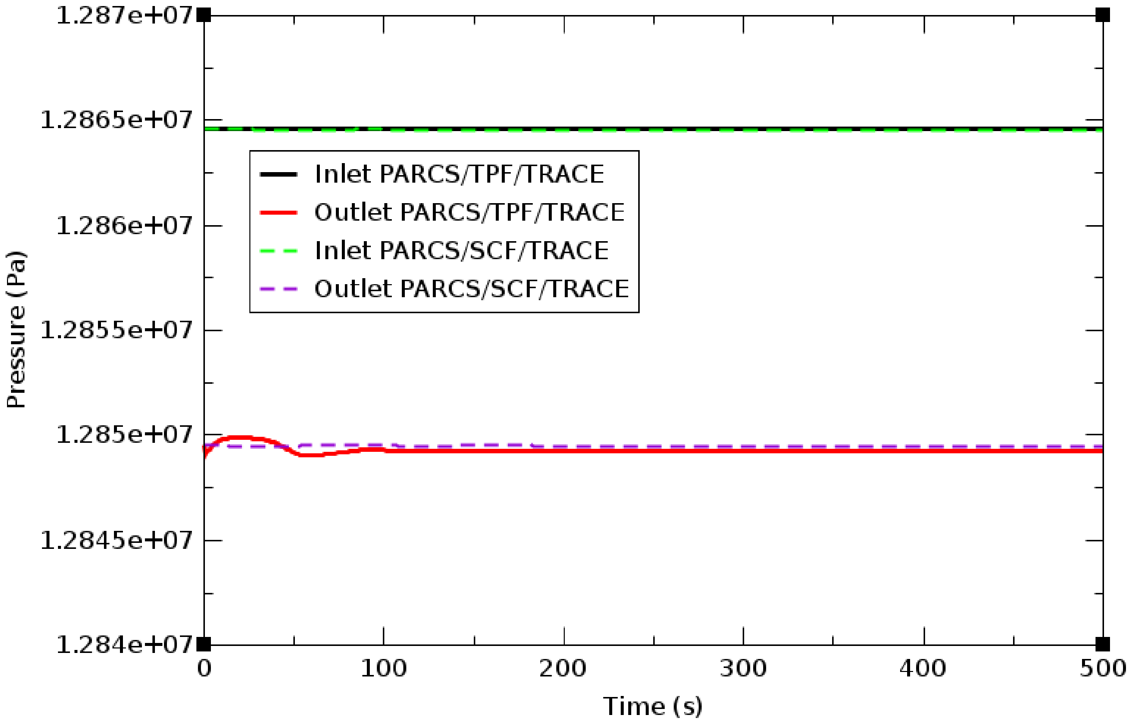
FAVOR Technique

- Fractional Area Volume Obstacle Representation

$$\vec{V}_k = \begin{pmatrix} \varphi_x & V_x \\ \varphi_y & V_y \\ \varphi_z & V_z \end{pmatrix}$$

Where,φ is the flow area fraction, x, y, z represent the Cartesian coordinates, and k will become the fluid (l when liquid and v when vapor).

Pressure comparison



Coolant Temp comparison