# The development of Serpent-OpenFOAM environment for multi-physics calculations in Swierk Computing Centre

**Mikolaj Adam Kowalski**

**Email:** ***mak60@cam.ac.uk***

**6th International Serpent User Group Meeting, Milan**
**27.09.2016**

# Plan of the presentation

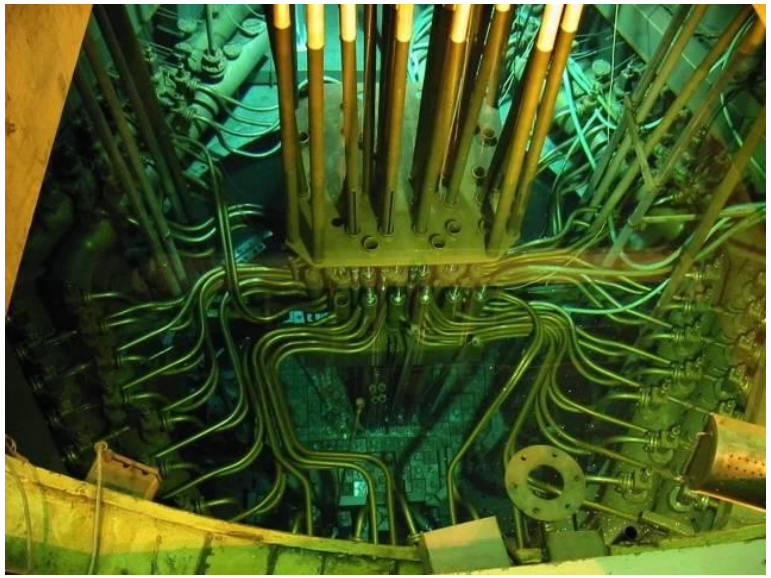
1. Computing Centre Swierk
2. Motivation for the work
3. OpenFOAM
    1. Modifications necessary
    2. Problems encountered
4. Serpent setup
5. Coupling Algorithm
6. Results
7. Final Remarks

# Computing Centre Swierk (CIS)

*NCBJ aerial view*

- **Computing Centre Swierk** was founded in 2009 to satisfy computational need of polish energy industry and support the development of nuclear power industry in Poland

- **Computing Centre Swierk (CIS)** is a part of **National Centre for Nuclear Research (NCBJ)** which:
  - Is the main Institute doing research into Nuclear Technology
  - Operates Research Reactor MARIA, which is a major producing of medical radioisotopes



*MARIA reactor*

# CIS Cluster

**Computational Resources**:

- ~1 PFLOPS
- 178.5 TB RAM
- 18.2 PB HDD

**Available Work Nodes:**

- **HP ProLiant BL685c**
  - 4 AMD Opteron 6276 CPUs (64 Physical cores) at 2.3-3.1GHz
  - 256GB DDR3 RAM
- **SuperMicro TwinBlade**
  - 2 Intel Xenon E5-2680 v2 CPUs (20 Physical Cores, 40 Logical cores with HyperThreading) at 2.8-3.6 GHz
  - 128 GB DDR3 RAM

**Generation Gap -> A lot of resources are unused**

During my Internship about 70 Work Nodes on average were instantly available

# “Science-in-return” idea

**Idea:**
**Computational Resources in exchange for competence**

**Collaboration with NRG Netherlands:**

- 10 769 888 GHz-hours (Equivalent of 220 physical CPUs occupied 24h a day for 2 years) with IT support

**In exchange for:**

- Internships at NRG site (access to NRG’s Best Practices)
- Repetitive one-week-trainings on our site
- Advice on current projects
- Collaborative publications

**Collaboration coordinated by:**

Piotr Prusinski (Piotr.Prusinski@ncbj.gov.pl**)**

# Serpent-OpenFOAM multi-physics environment

Underutilised resources → Could be used to work on computationally expensive problems

Thus idea to develop Serpent-OpenFOAM multi-physics environment on the cluster

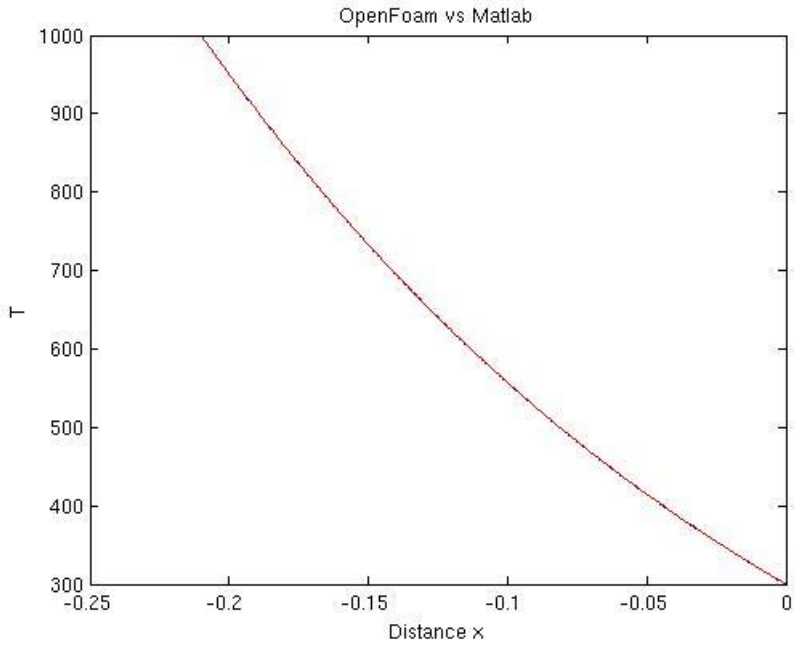
**Project Goal**: Recreate results from Riku Tuominen MSci Thesis at Aalto University "*Coupling Serpent and OpenFOAM for neutronics – CFD multi-physics calculations*" (https://aaltodoc.aalto.fi/handle/123456789/17759)

Presented work has little scientific value but may offer a good perspective on problems a new user may encounter while learning to couple Serpent with OpenFOAM

OpenFOAM in version 2.4 and Serpent in version 2.1.26 were used

# Modifications to OpenFOAM – Fuel and cladding properties

- New Thermophysical Models were created for thermal properties of fuel and cladding
- They were based on FRAPCON correlations
- OpenFOAM implementation was tested against independent implementation in Matlab
- **The same correlations were used in Tuominen's work**




*Comparison of temperature distribution in 1D heat conduction through fuel between OpenFOAM and MATLAB*

# Modifications to OpenFOAM – Corrected coupled Boundary Conditions

- In OpenFOAM “default” steady-state conjugate heat transfer solver (**chtMultiRegionSimpleFoam**) each material region is an independent mesh with independent settings
- Regions communicate through temperature field by boundary condition that preserves temperature and heat flux at region boundaries (**turbulentTemperatureCoupledBaffleMixed**)
- chtMultiRegionSimpleFoam solves energy transport for enthalpy (not Temperature!)
- turbulentTemperatureCoupleBaffleMixed uses temperature to evaluate heat flux

# Modifications to OpenFOAM – Corrected coupled Boundary Conditions

- In General: $\boldsymbol{k\nabla T = \alpha \nabla h}$ **for** $\boldsymbol{\alpha = \frac{k}{c_p}}$
- However $\boldsymbol{k\nabla T \neq \alpha \nabla h}$ if gradient is **evaluated on a discrete mesh** and **cp=fn(T)**
- Thus there is discrepancy in heat flux between regions if cp=fn(T)

- It is necessary to modify **turbulentTemperatureCoupledBaffleMixed** to evaluate heat flux based on enthalpy
- Resulting heat balance equation needs to be solved numerically (i.e. with Newton's iteration)

This problem was originally found and solved by Tuominen in his Thesis

# Modifications to OpenFOAM – Water Properties

Open source implementation of IAPWS-97 in Freesteam 2.1 library was used

Freesteam library was connected to OpenFOAM with open source IAPWS-IF97-OF wrapper class written by Roman Thiele from KTH.

SST k-ω turbulence model was used

# Modifications to OpenFOAM – Minor changes to the solver

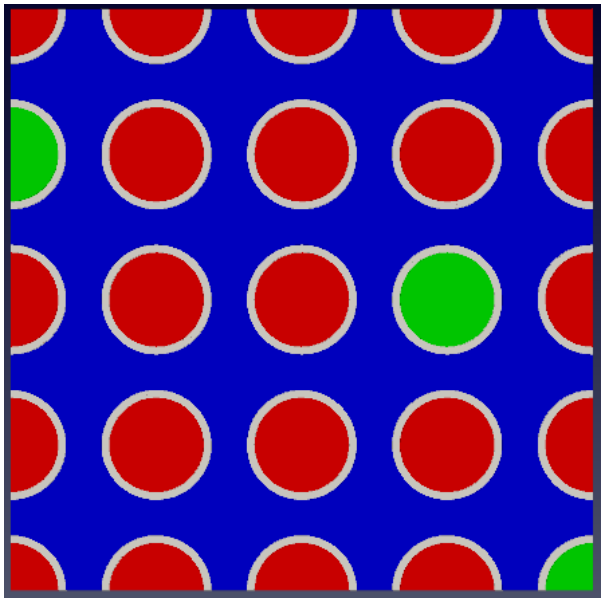
Additional field **volP** was defined in the solver to provide with spatially variable explicit source term in energy equation.

Additional code was written to monitor energy balance in each region.

Following values are printed at each time step for each region:

- Total heat generation in region volume
- Total heat flow across region boundaries

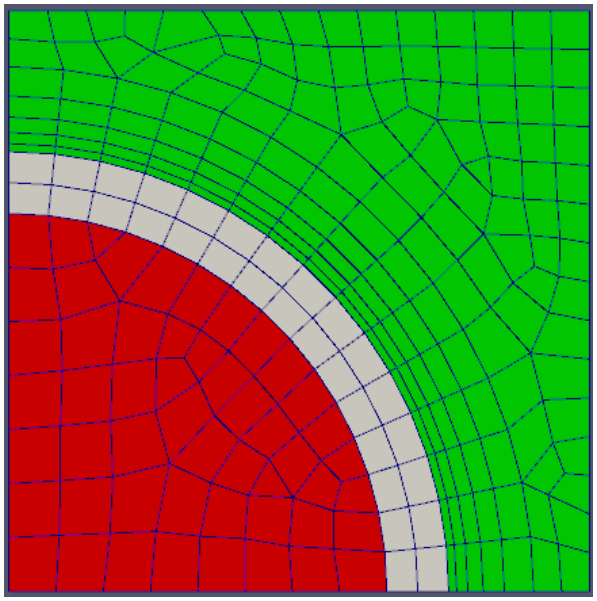
# OpenFOAM Mesh



Tuominen's 5x5 PWR assembly

3 701 760 Cells

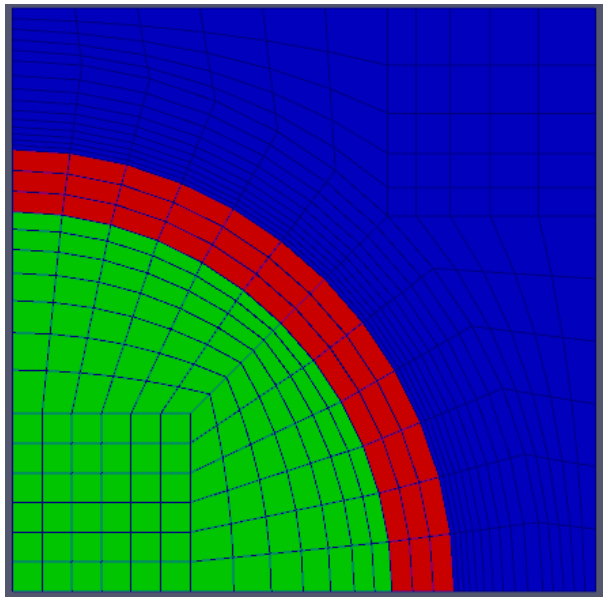
Tuominen's Geometry was obtained by reflecting ANSYS ¼ Pin Mesh



ANSYS Meshing ¼ Pin Mesh

241 Faces

57 840 Cells



BlockMesh ¼ Pin Mesh

361 Faces

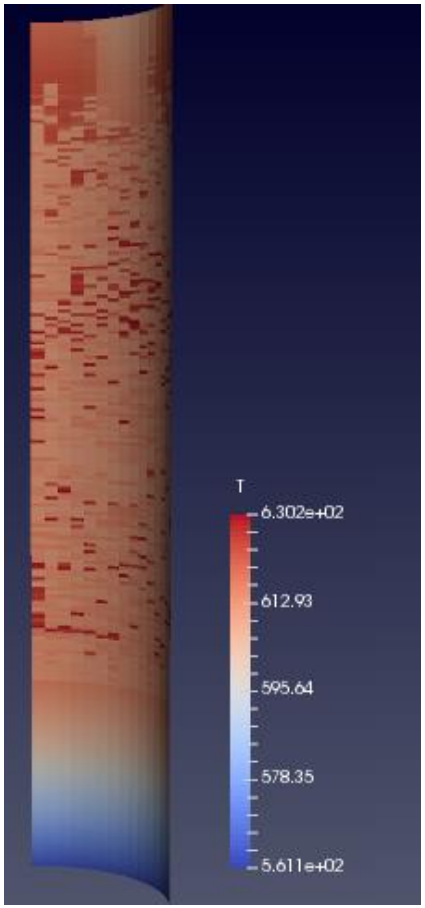
86 640 Cells

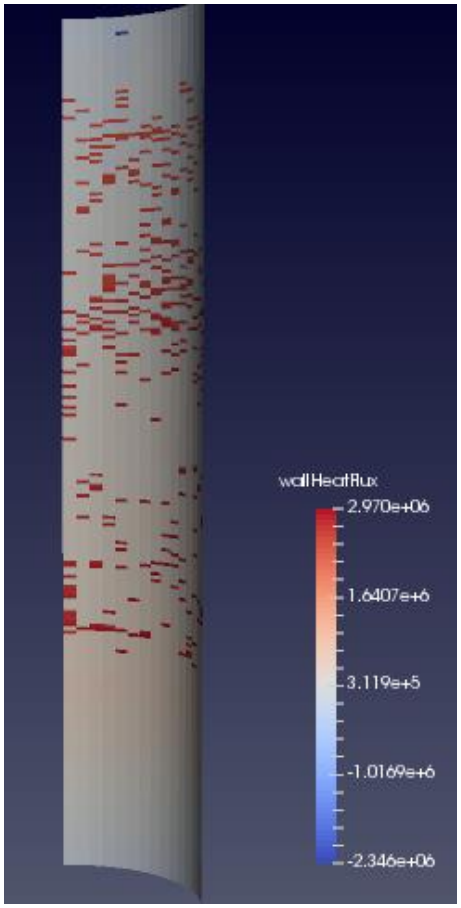
For 240 elements along the pin

Pin length 360cm

# OpenFOAM – Problems with stability

OpenFOAM case was found to be unstable for high heating values. Pattern of “Hot Spots” emerged.




- Stability was improved when water transport properties (k,μ) were assumed constant
- Problem is persistent for every “fixed Gradient” boundary condition
- Problem disappears for fixed temperature BC
- This behaviour may be caused by some feedback between thermophysical model and BC
- More investigation is necessary

# Serpent setup

- Serpent and OpenFOAM were used with separate geometry definitions.
- Unfortunately Serpent was compiled on the cluster only with MPI
  - It was impossible to run 5x5 PWR Assembly case on more then 4 threads
  - It was probably caused by memory multiplication caused by MPI parallelism
- Separate type 7 multi-physics interfaces were defined for each material region

Serpent “Quirks”:

- When serpent is writing power to OpenFOAM field it sets boundary patches to “Fixed Value”. This crates problems when symmetry BC are used in OpenFOAM case
- Serpent cannot read uniform OpenFOAM fields

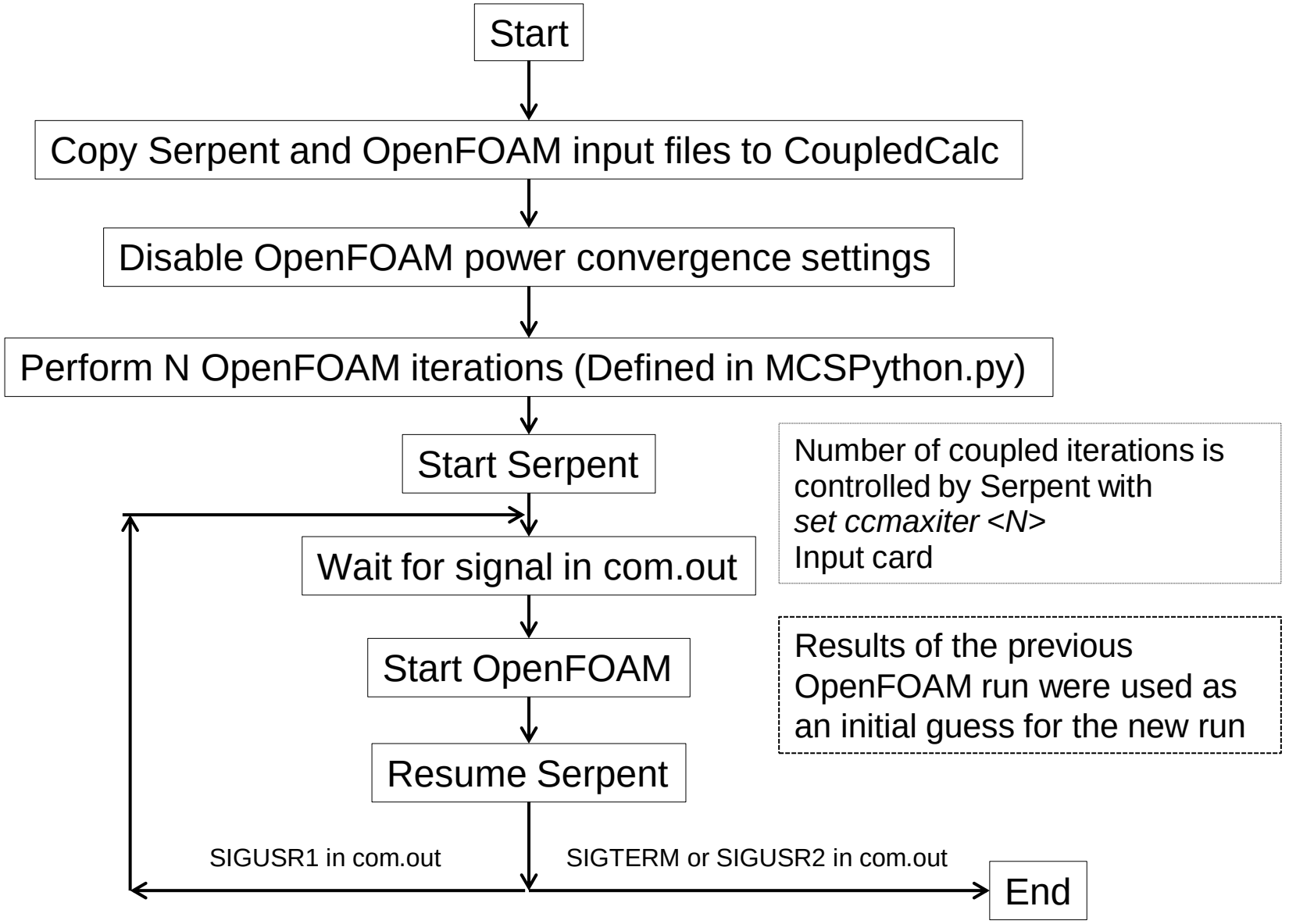
# Coupling Algorithm

- Coupling script was written in Python and was based on “Minimal Serpent Coupling Script“ by Vile Valavirta available on Serpent wiki.
- Communication between script and Serpent took place through UNIX signals and  com.in and com.out files

**Folder Structure**

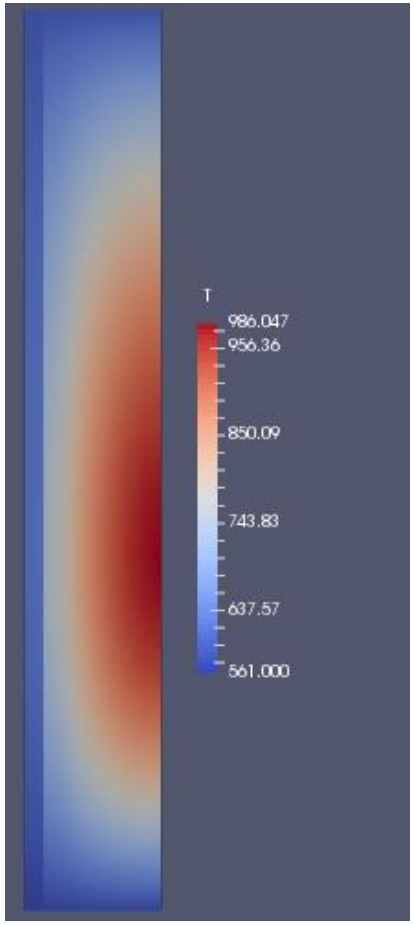
-Case Main Folder
- -Serpent_Init
  - - Serpent Input file
  - - Interfaces definitions
- -OpenFOAM_Init
  - - 0,constant and system folders
- -CoupledCalc
  - -empty folder
- -MCSPython.py (coupling Script)

# Coupling Algorithm

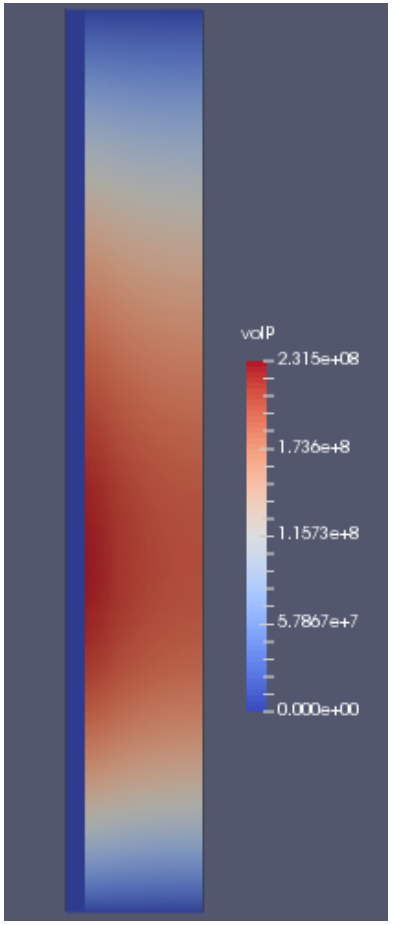


# Simple Coupled Results

- Total heating power was low (9 kW -> 2.5 kW/m)
- Coupling script worked correctly
- Results were obtained on ANSYS mesh
- Python script to evaluate temperature field convergence was written
- Power relaxation factor was set to 0.8

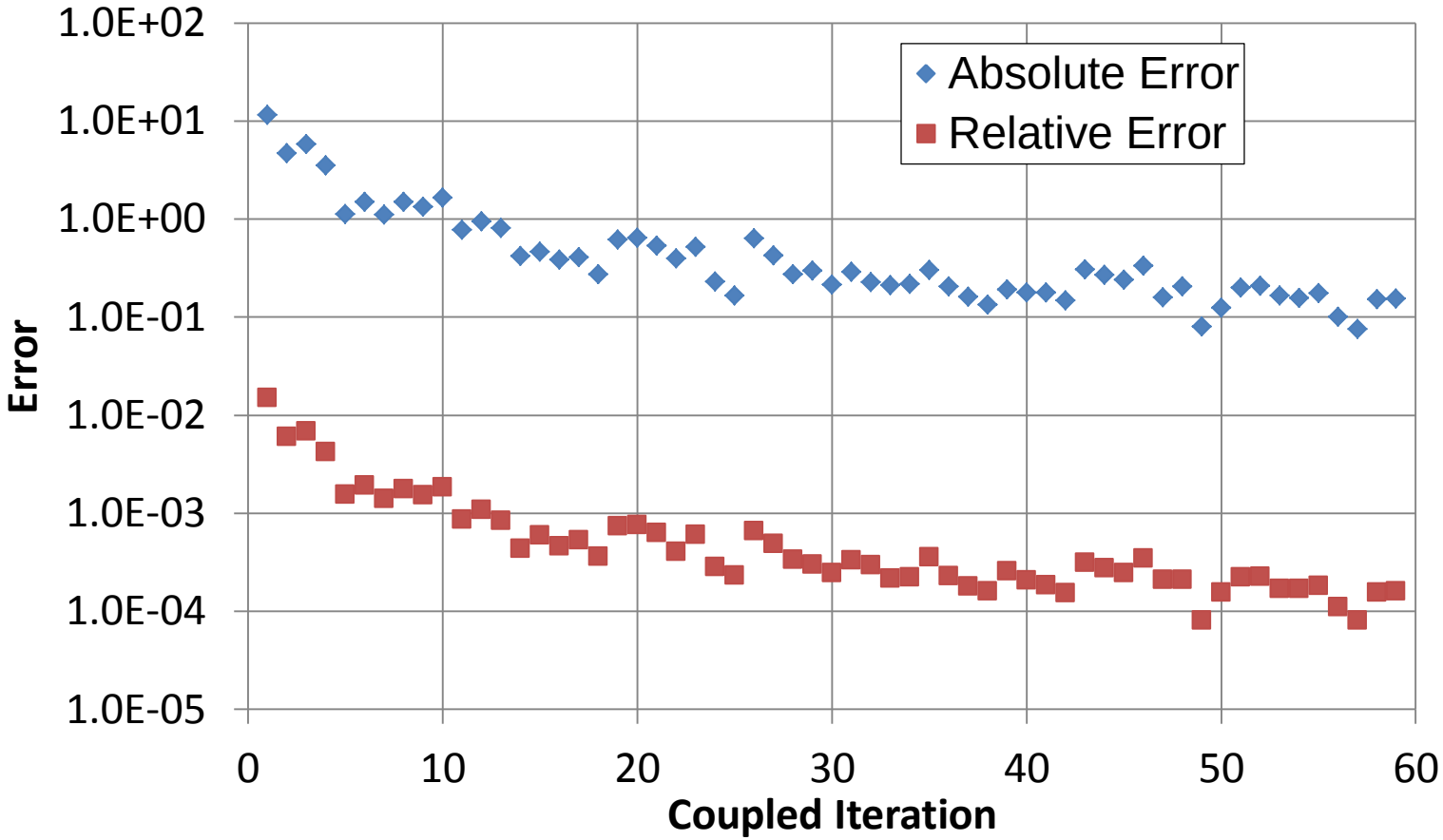



Temperature in fuel and cladding




Power generation in fuel and cladding

# Simple Coupled Results




Coupled solution converges

# Final remarks

**What was accomplished:**

- Serpent 2 and OpenFOAM were successfully made to communicate
- Python coupling script was successfully developed

**What problems/bugs need to be solved:**

- Serpent 2 with OpenMP based parallelism needs to be compiled and tested on the cluster
- Problem of OpenFOAM simulation stability at high(realistic) heating values needs to be addressed (I see no simple solution. Help would be appreciated!)

**Future work:**

- Try to couple OpenFOAM and Serpent in transient simulation
- Test coupling for other reactor geometries (i.e. MARIA core)
- Create a tutorial to help guide new users into Serpent-OpenFOAM coupling (gather all key information in one document)

# Thank you for your attention!

*This work was partially supported by the EU and MSHE grant no. POIG.02.03.00-00-013/09 - Świerk Computing Centre*