RWTH AACHEN UNIVERSITY
INSTITUTE FOR REACTOR SAFETY AND TECHNOLOGY
UNIV.-PROF. DR. RER. NAT. HANS-JOSEF ALLELEIN

JÜLICH FORSCHUNGSZENTRUM
INSTITUTE OF ENERGY- AND CLIMATE RESEARCH
NUCLEAR WASTE MANAGEMENT AND REACTOR SAFETY (IEK-6)

Coupling of SERPENT with Fluid Dynamics code ANSYS CFX

Anni Schulze
schulze@lrst.rwth-aachen.de

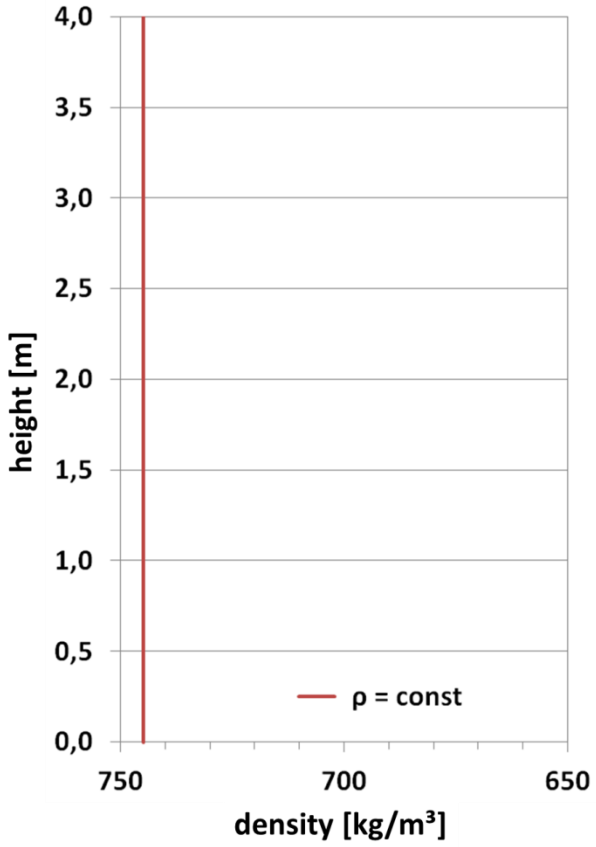
Content

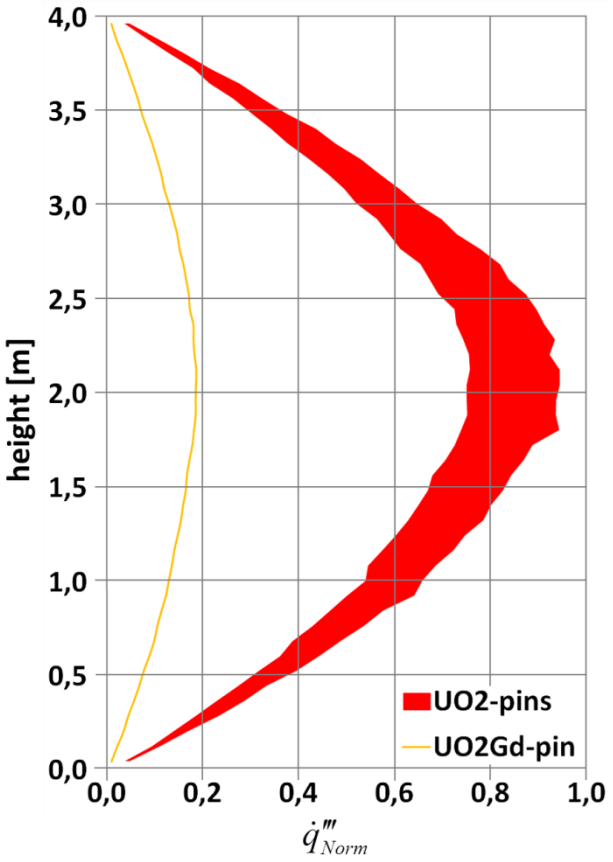
- Introduction
  - Why coupling?
  - The overall coupled system
- Simulation of Fluid Dynamics in ANSYS CFX
  - Geometry, Mesh, Boundary Conditions, User Fortran
- The coupling itself
  - Calling SERPENT
  - Coupling scheme
- Summary and „Wishlist“

Why coupling 3D-codes?

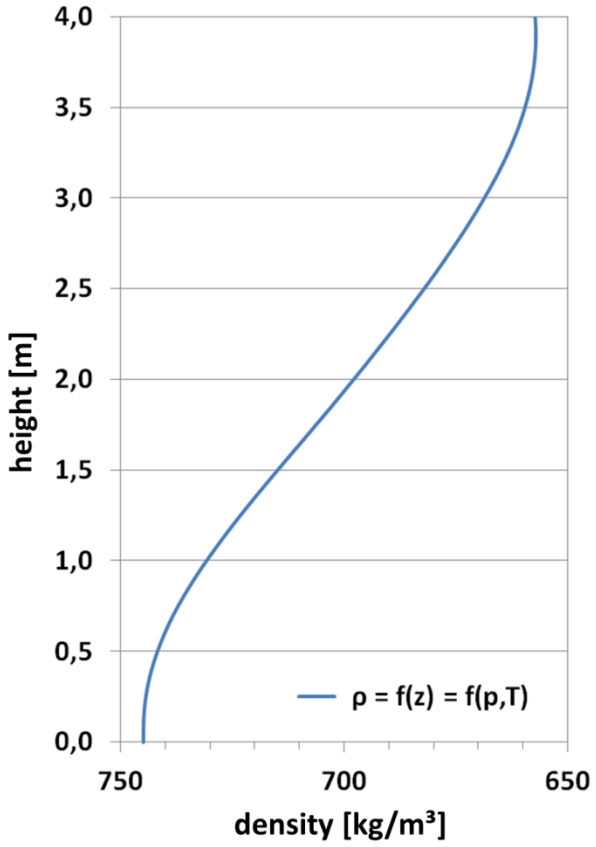
- neutronics and thermal hydraulics are strongly influencing each other via moderating and temperature feedback

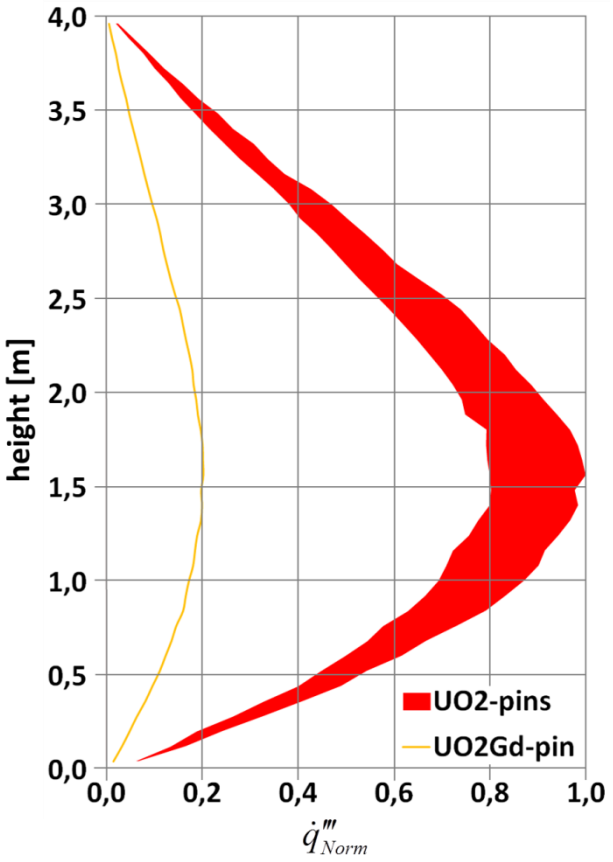
influence of axial density distribution: ρ = constant





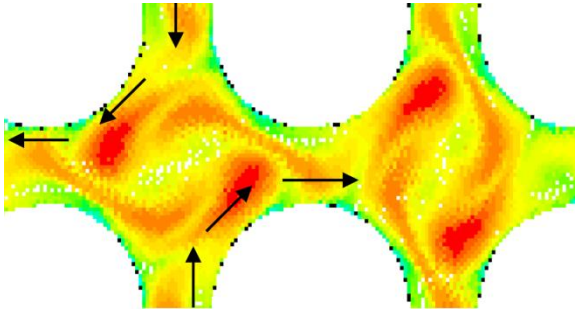
influence of axial density distribution: ρ = f(z) = f(T,p)

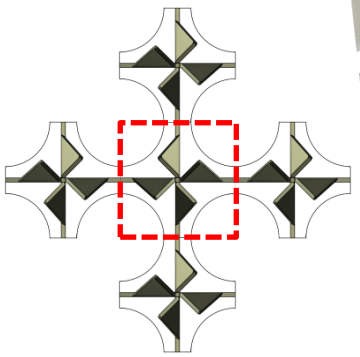


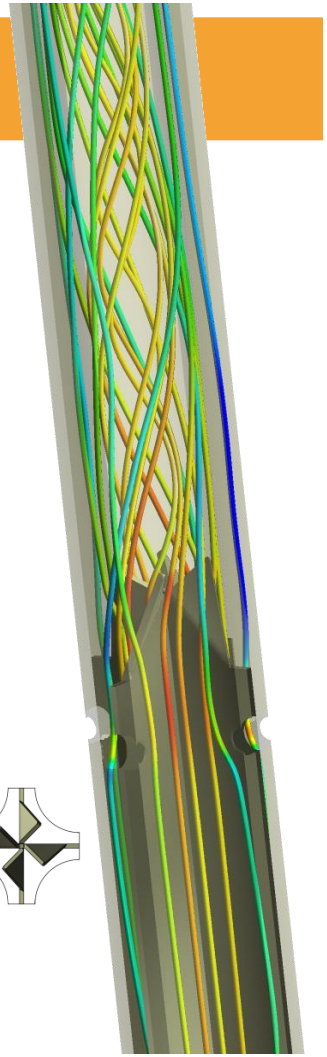


Why coupling 3D-codes?

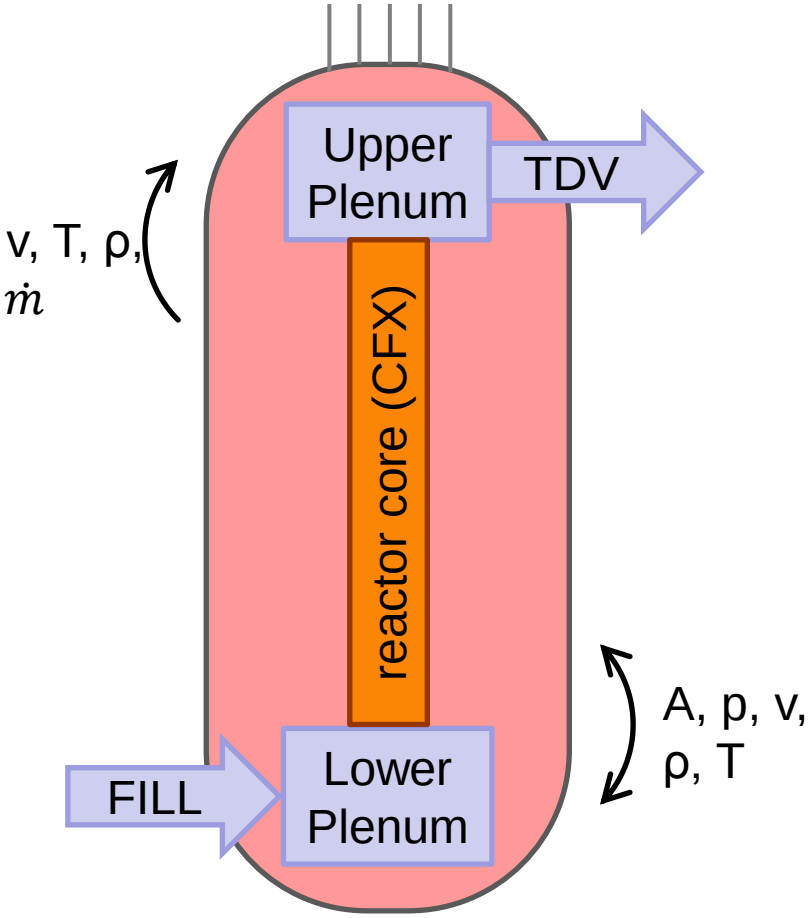
- neutronics and thermal hydraulics are strongly influencing each other via moderating and temperature feedback
- three-dimensional effects
  - spacer grids
  - cross flow







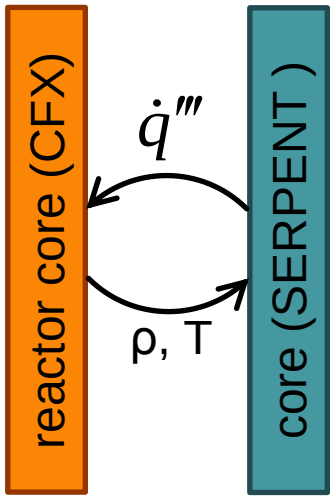
coupled system of ATHLET-CFX-SERPENT



ATHLET

- thermal hydraulic system code (1D) by GRS
- providing BC arising in primary circuit to CFX‘s core geometry
- called by CFX whenever new BC‘s are needed (timestep)

coupled system of ATHLET-CFX-SERPENT



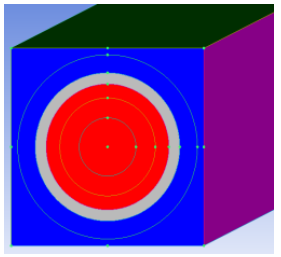
CFX

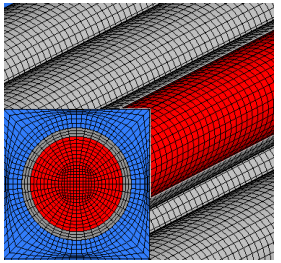
- thermal hydraulic code (3D)
- is calling SERPENT to get power distribution
- providing SERPENT with distribution of density and temperature

Simulation of Fluid Dynamics in ANSYS CFX

1. build up the geometry
2. create the mesh
3. setup simulation conditions
   - solid + fluid domains (materials, interfaces)
   - inlet + outlet (p, T, v), Heat Source $\dot{q}'''$
4. simulation
   - resolving Reynolds-Averaged-Navier-Stokes equations
   - computationally intensive

     → only 3x3 pin geometry





programming inside CFX → coupling with SERPENT

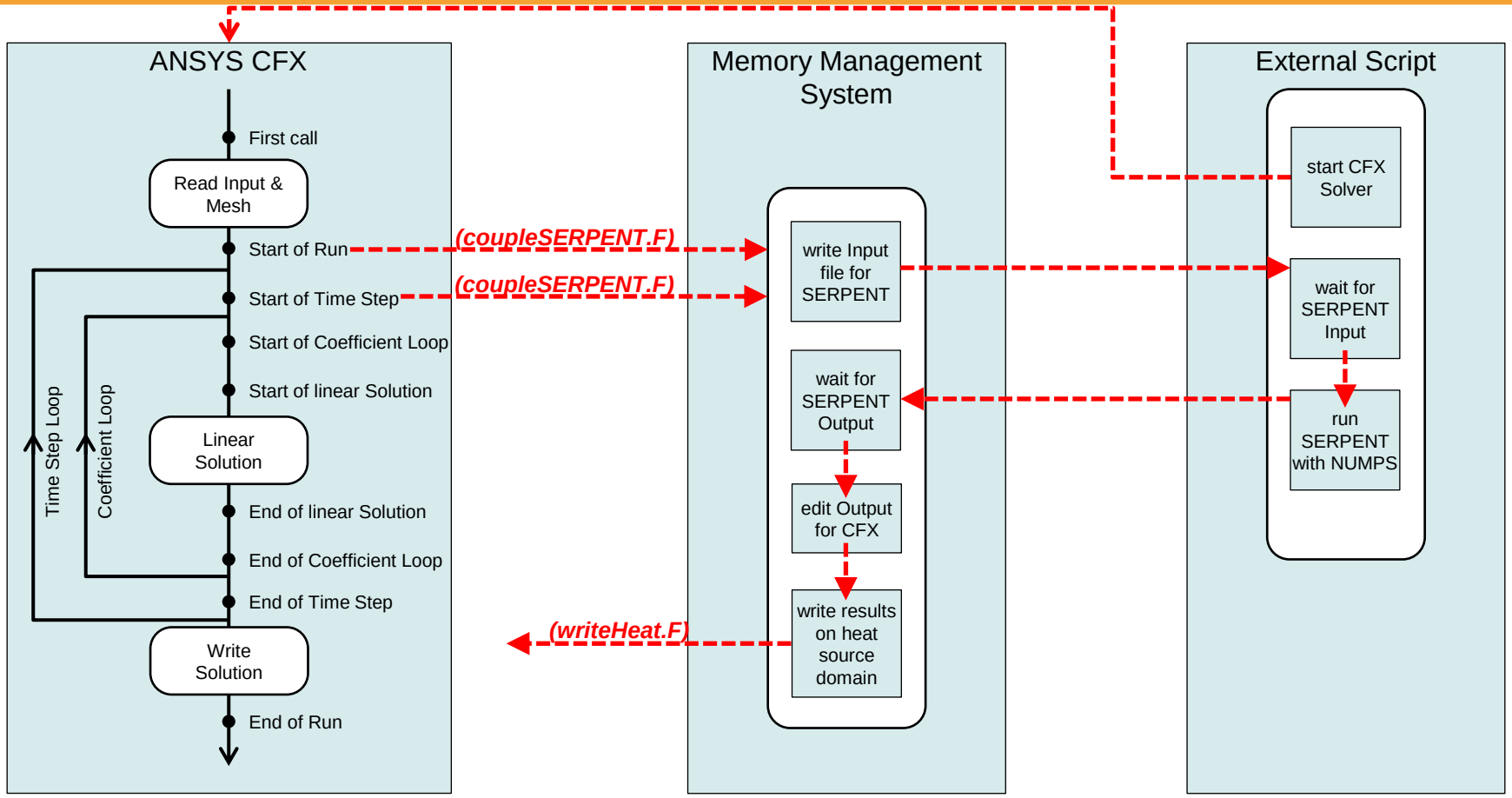
- Setup can be done via User Fortran
  - not too easy to handle
  - physical quantities inside variable stacks, access via pointers
  - Data handling by means of program flow or data controlled subroutines

→calling SERPENT as a need in heat source data

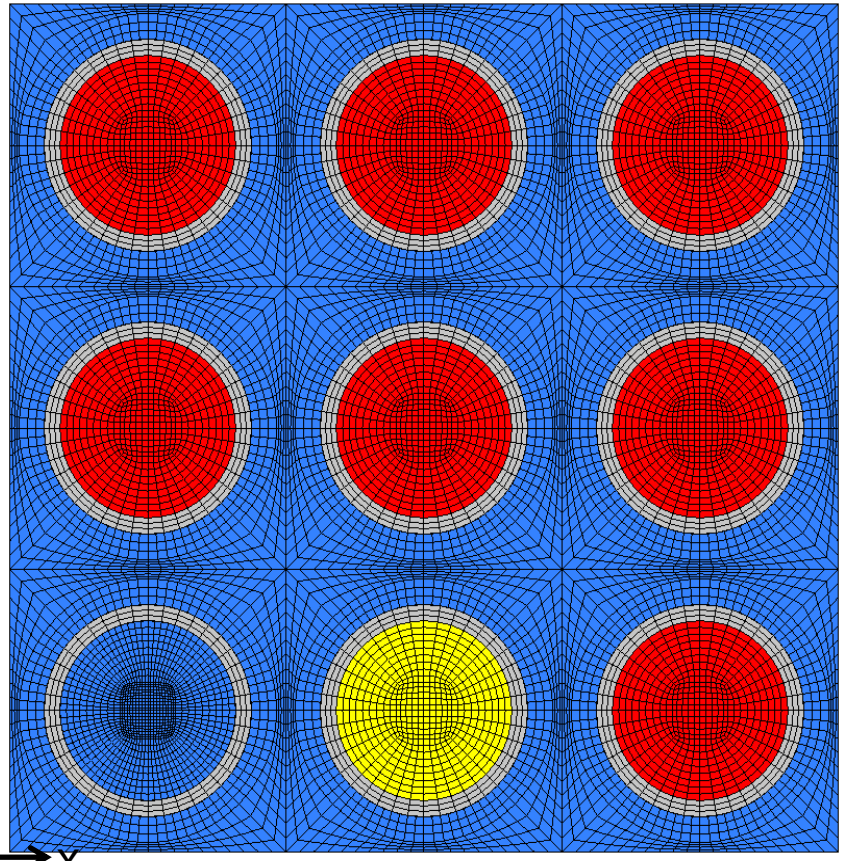
```
1 cool 1
power 40 0 400 4
3 10 1
16040
-1.897 -1.897 0 -0.7448 -1
```

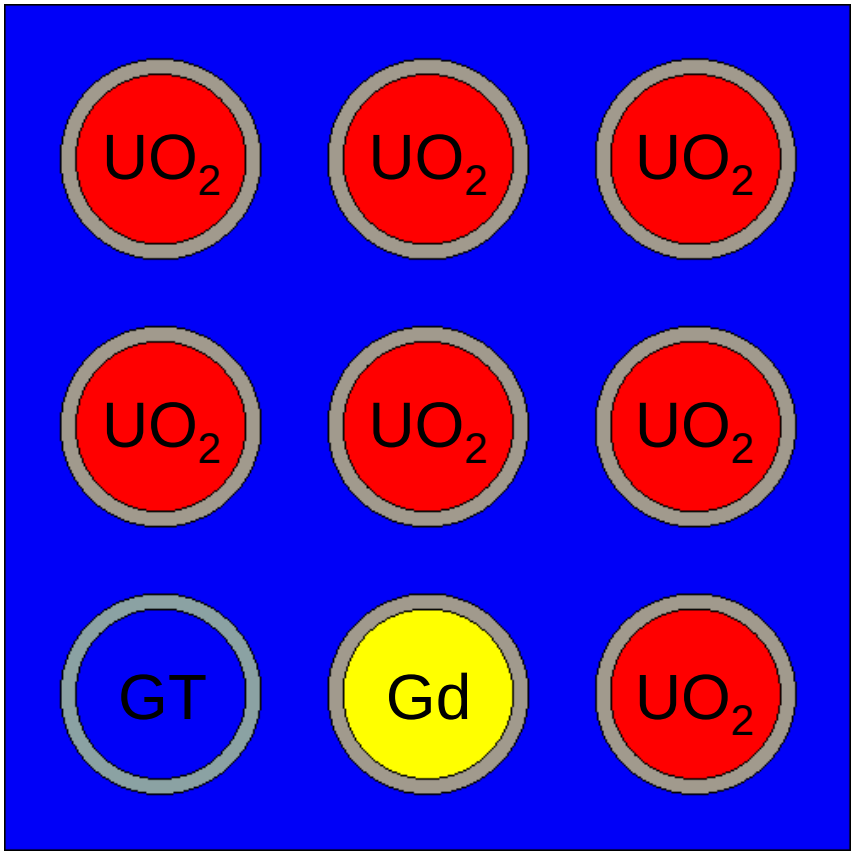
  - writing Input-file for SERPENT run
  - running SERPENT (until now, CFX cannot call external solvers)
  - handling SERPENT output for the needs of CFX
  - resuming CFX simulation with new heat source data

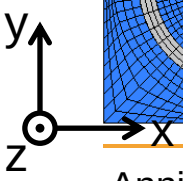
CFX ⇔ SERPENT: coupling scheme



CFX ⇔ SERPENT: geometries

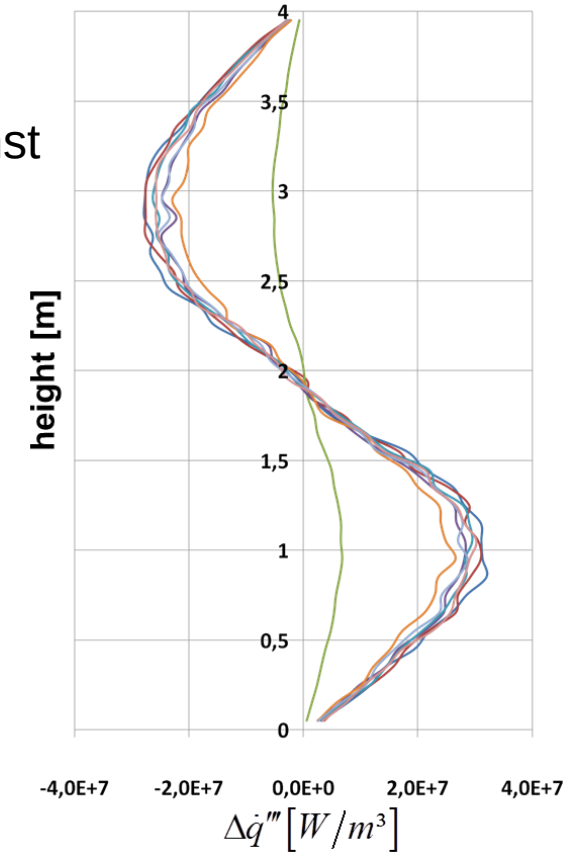




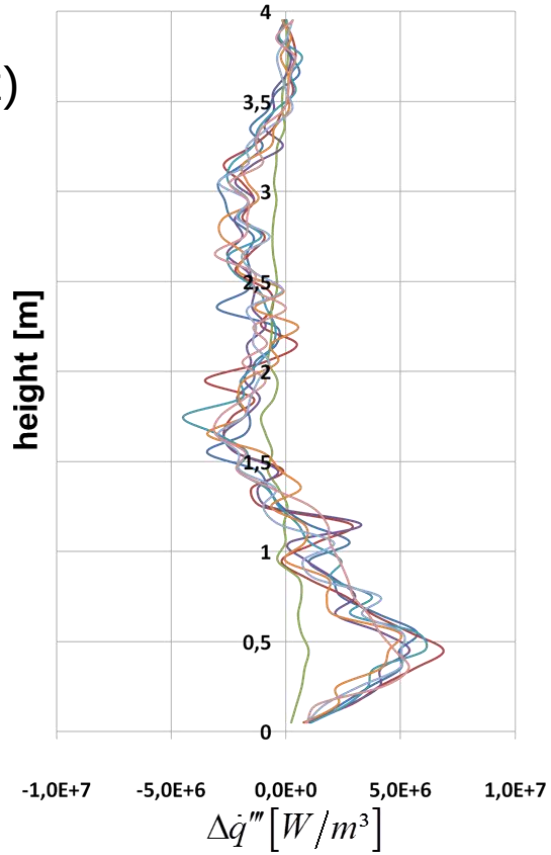


CFX ⇔ SERPENT: first outcome

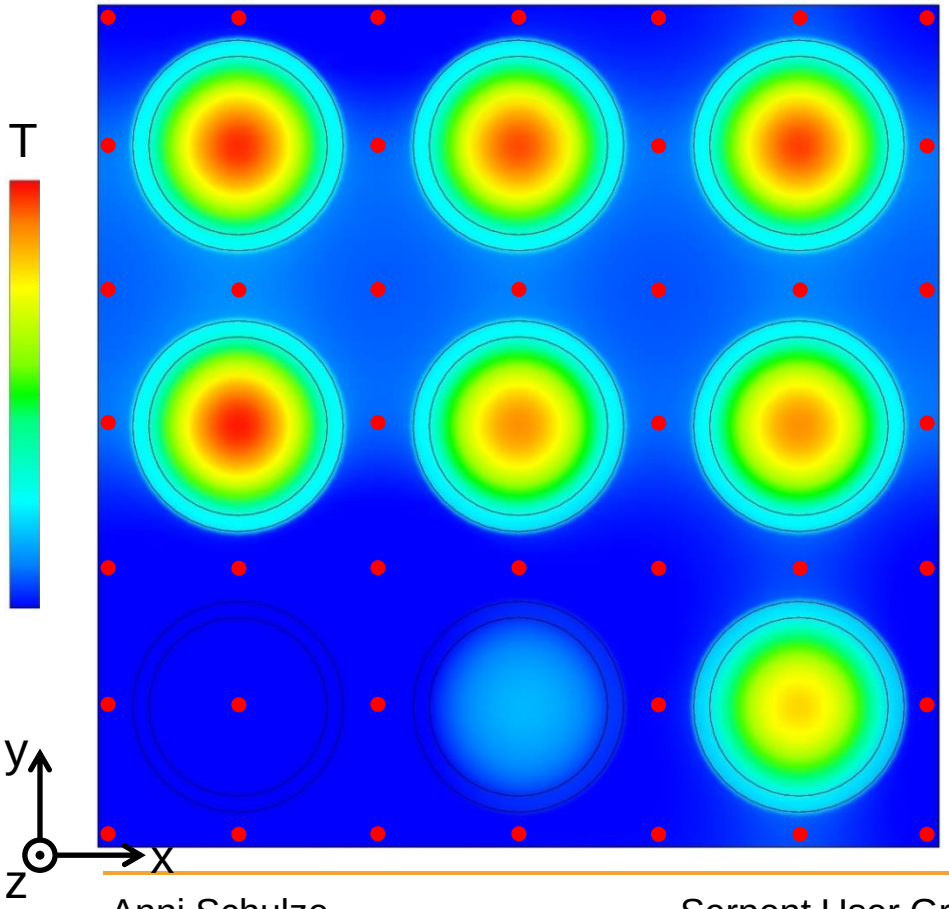
ρ=f(z)
- ρ=const

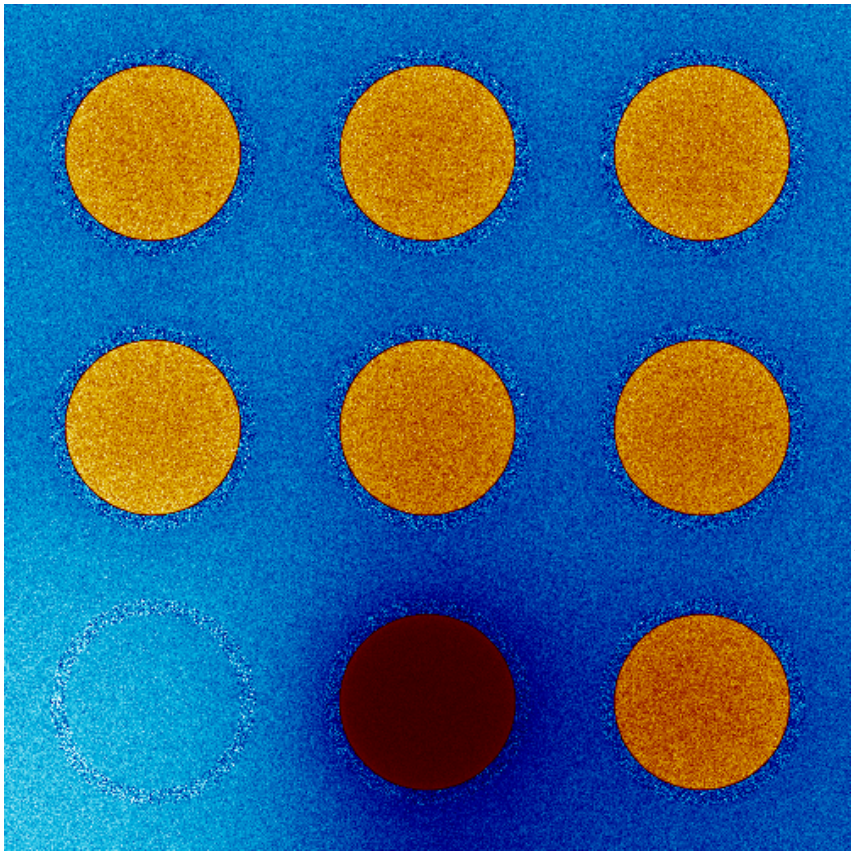


ρ=f(x,y,z)
- ρ=f(z)



CFX ⇔ SERPENT: first outcome





Summary and „Wishlist“

- coupling CFX and SERPENT
  - coupling in two different languages
  - coupling via two (or more) controlling routines
- first outcome
  - results make sense from a physics point of view
  - computationally intensive !!!
- accelerate SERPENT calculation
  - interpolation of density (and temperature) at the beginning
  - power distribution in NUMPS only for Uranium zone
  - … transient simulations