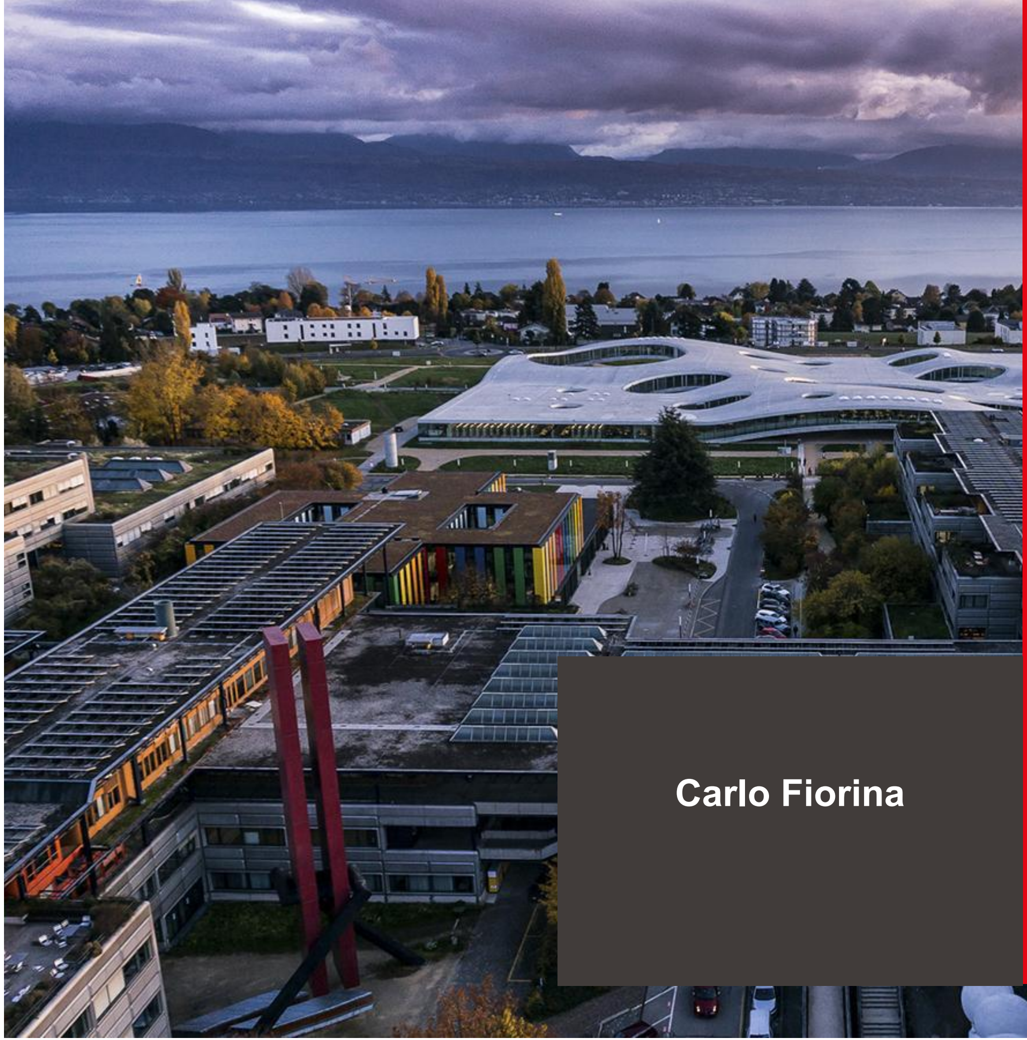
EPFL



Introduction to nuclear reactor modelling using OpenFOAM

Carlo Fiorina

École
polytechnique
fédérale
de Lausanne

Disclaimer

- Focus on the use of OpenFOAM for multiphysics
  - Use of OpenFOAM as CFD tool widely covered by documentation, forums, courses, etc.
- Focus on already existing tool (GeN-Foam) as an example
  - Programming from scratch is not that difficult, but unsuited for a 75 minutes lecture
- In the slides, more material than can actually be covered in this lecture
  - Can help better understanding the slides after the lecture

Content

- General Introduction
- Introduction to the use of OpenFOAM
- Basics of GeN-Foam
- Short introduction on the use of GeN-Foam

Objective

What is it about?

- Provide you with general information, references, suggestions, terminology and lessons learnt that can facilitate your approach to the OpenFOAM world
- Provide with slides that can help you out orienting yourself if you decide to embrace the use of OpenFOAM

What is not about?

- Detailed course on the use of OpenFOAM
- Hands-on training

The IAEA-facilitated ONCORE initiative

ONCORE to support the open-source nuclear community and help addressing typical shortcomings of open-source development (scattered community, documentation, QA, loss of knowledge)

- Promote collaboration and facilitate communication (connect the community)
- Provide guidelines for code contribution (documentation, QA)
- Provide development best practices (QA)
- Preserve knowledge
  - Incl. compiling a list of open-source codes

https://www.iaea.org/topics/nuclear-power-reactors/open-source-nuclear-code-for-reactor-analysis-oncore

A first important outcome: list of available codes

- https://nucleus.iaea.org/sites/oncore/SitePages/List%20of%20Codes.aspx
- A vibrant community with an impressive R&D output
- ~35 codes already identified so far:
  - OpenMC
  - Raven
  - Dragon
  - MOOSE
  - Salome platform (Code_Saturne, Code_Aster)
  - TrioCFD
  - …
  - **Several OpenFOAM-based tools**

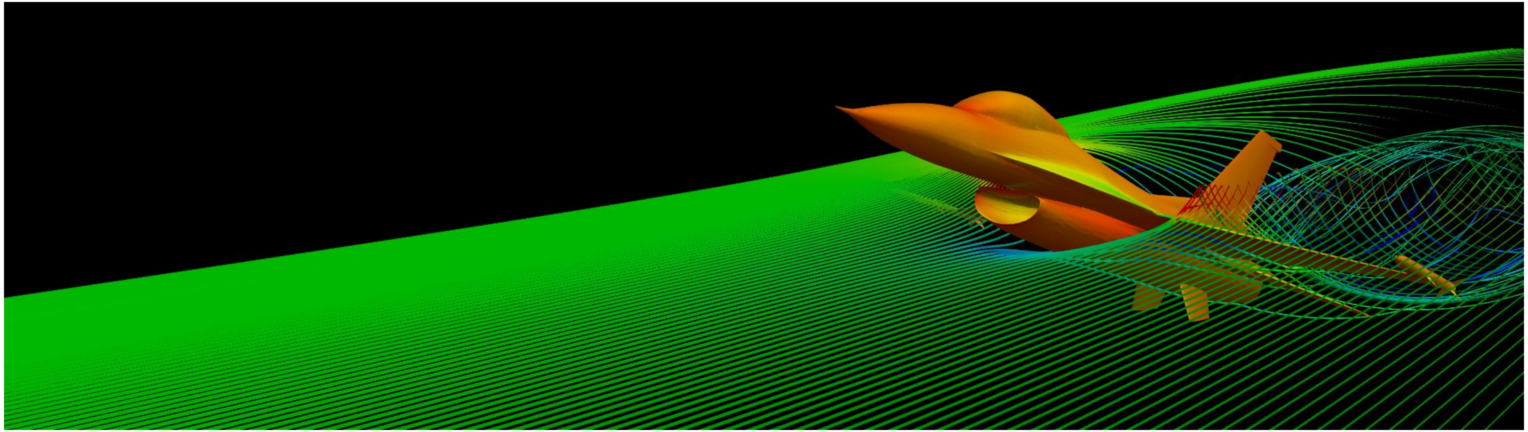
A central tool for open-source simulation

C. Fiorina

- What is OpenFOAM?
  - Distributed as CFD toolbox
  - ~10k to 20k estimated users worldwide

OpenFOAM

The Open Source CFD Toolbox



A central tool for open-source simulation

C. Fiorina

OpenFOAM

The Open Source CFD Toolbox

- What is OpenFOAM?
  - Distributed as CFD toolbox
  - ~10k to 20k estimated users worldwide
  - OpenFOAM = Open Field Operation And Manipulation
  - Essentially a large, well organized, HPC-scalable, C++ library for the finite-volume discretization and solution of PDEs, and including several functionalities like ODE solvers, projection algorithms, and mesh search algorithms
  - Object-oriented, with a high-level “fail-safe” API

$$\frac{1}{v_i}\frac{\partial \varphi_i}{\partial t} - \Delta(D_i \varphi_i) = S$$

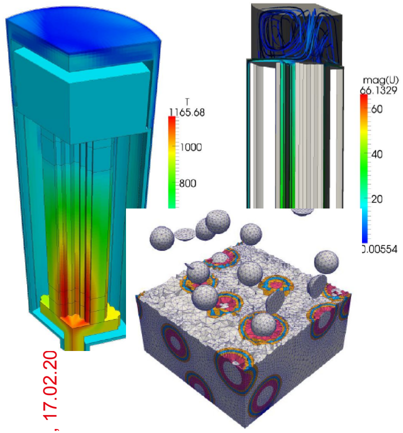
```
fvm::ddt(IV,flux_i])- fvm::laplacian(D,flux_i])= S
```

Use of OpenFOAM for multi-physics

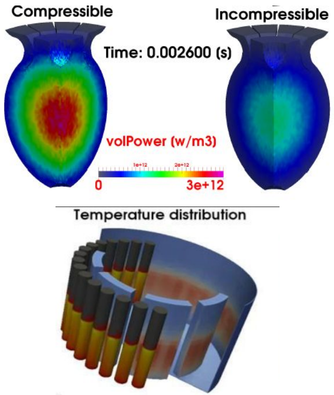
2000-2010
First activities

2010-2015
First widespread use

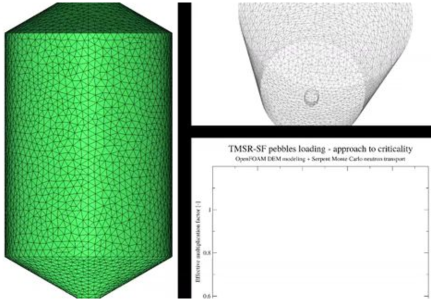
2015-2021
First coordinated and persistent developments



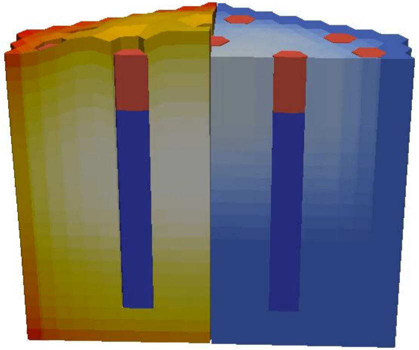
PBMRs and HTRs



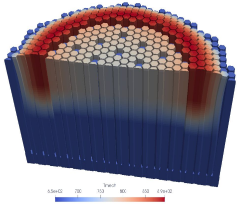
MSRs



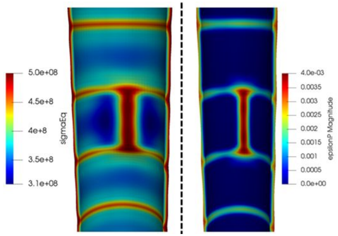
FHRs



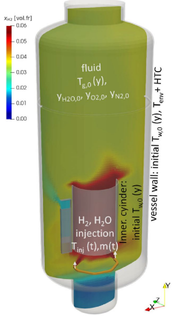
SFRs



GeN-Foam



OFFBEAT



containmentFoam

Use of OpenFOAM for multi-physics

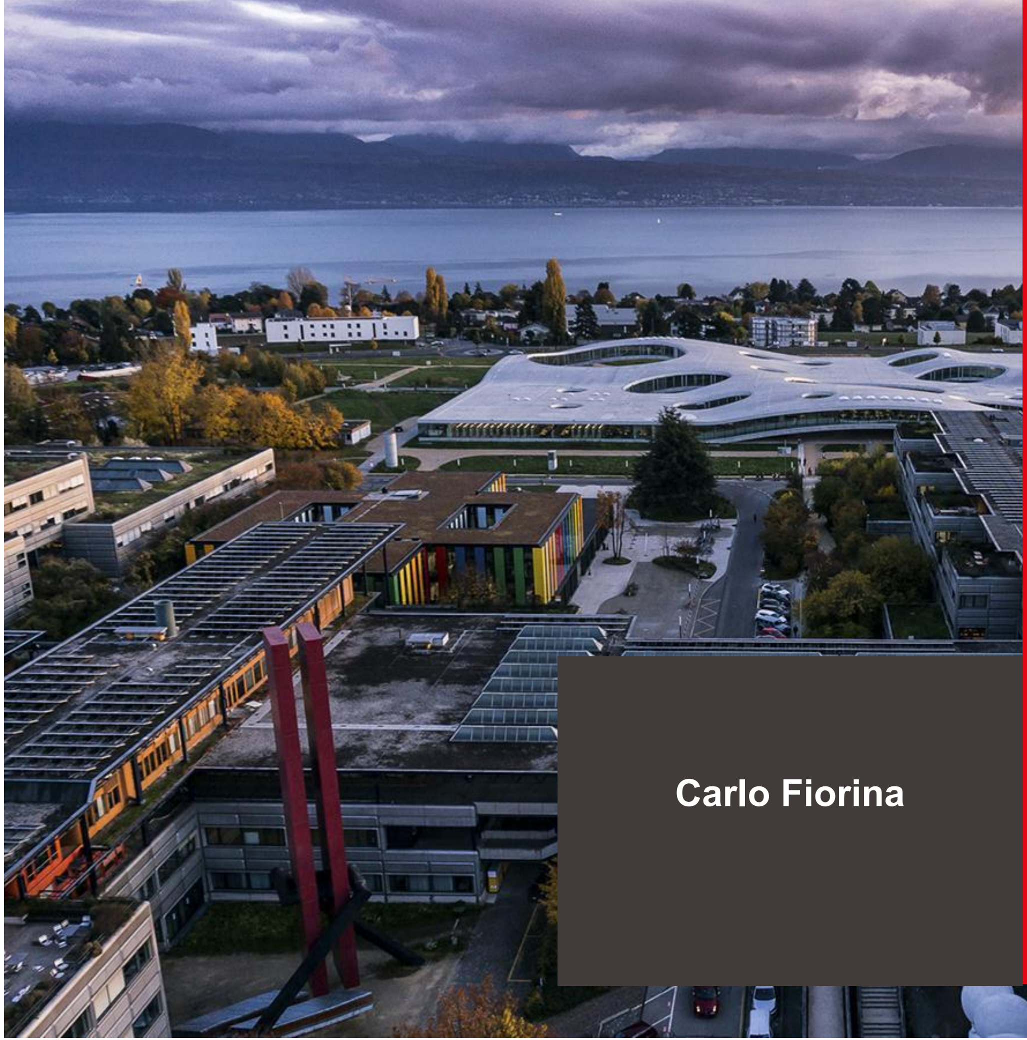
- With some practice and commitment OpenFOAM will allow you to model pretty much everything, often in a way legacy nuclear tool can't take

- No need to start from scratch:
    - Many developers will share their work
    - Some solvers are already available online

EPFL

Thank you

Carlo Fiorina



Introduction to OpenFOAM

Carlo Fiorina

École
polytechnique
fédérale
de Lausanne

OpenFOAM versions

C. Fiorina

- Two main versions of OpenFOAM
  - openfoam.com
  - openfoam.org
- And in addition, foamExtend project
- If you want to use an available solver, or take features from available solvers for your own solver, be very careful and select the right one!

Some essential features

C. Fiorina

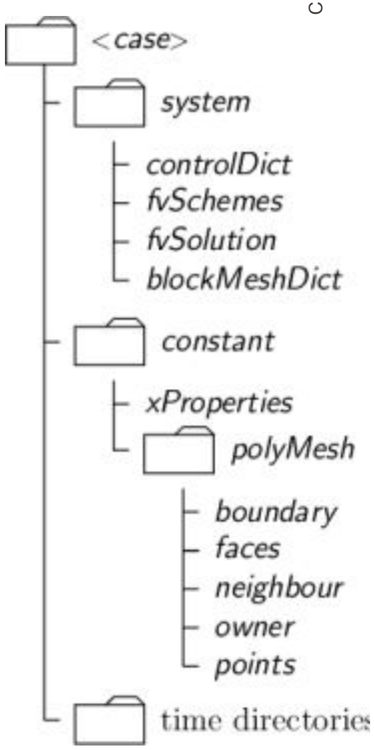
- OpenFOAM runs natively on Linux systems! (but you can use MAC, or the Linux subsystem for Windows of course)
- Mesh creation, input data, running and post-processing are 4 distinct steps
    - Mesh creation
    - Input data and mesh are gathered inside a Case Folder
    - Running
    - Post-processing

Mesh creation

- Don't take it lightly:
  - one of the most time consuming steps
  - requires good understanding of methods to decide the type of mesh and its refinement
  - a bad mesh will give a bad solution (especially for CFD)
  - in some unlucky cases, a bad mesh will give a non-convergent solution
- Several available free tools: blockMesh (embedded in OpenFOAM), Salome, gmsh, cfMesh, snappyHexMesh…
- Complex geometries and situations where high-quality mesh are needed may require the use of commercial software
- Make sure that the tool you chose allows you to separate your mesh into zones (called cellZones in OpenFOAM). They are necessary to assign different physical properties to different materials!

Input data

- All data (inlc. mesh) that OpenFOAM needs are collected into a Case Folder
- Inside a case folder you'll find at least 3 sub-folders
  - The folder "0", and possible other time directories, containing, for each field (viz., velocity, pressure, density):
    - Initial conditions
    - Boundary condition
  - The folder "constant" containing:
    - the mesh
    - all physical properties, gathered into "dictionaries"
    - the types of models (for instance k-epsilon or k-omega for turbulence), alse gathered into "dictionaries"
  - The folder "system" containing at least:
    - "controlDict", that gathers main simulation parameters like initial time, time steps, final time, etc.
    - "fvSchemes" that allows to set the type of discretization for various equations
    - "fvSolution" that allows mainly to set the parameters of the linear solvers

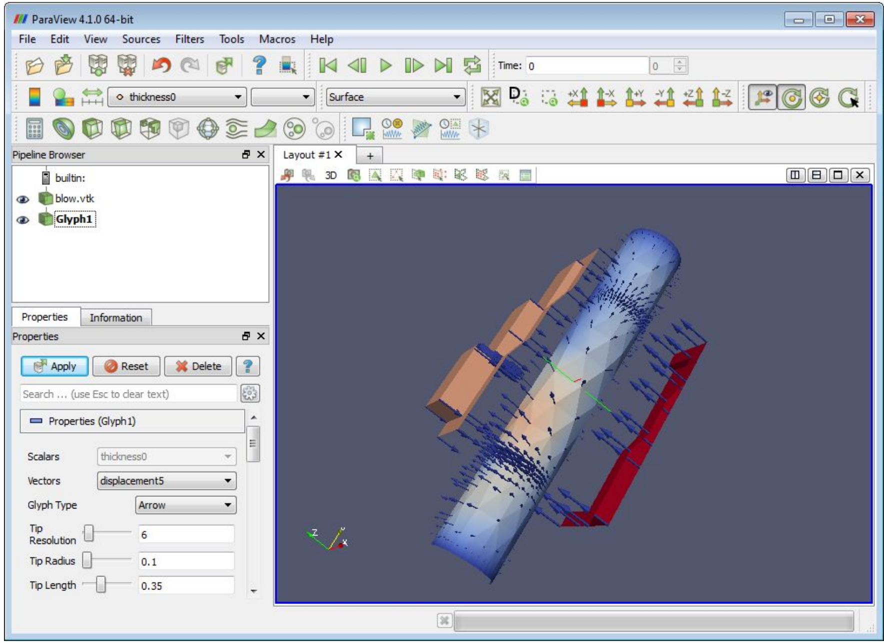


Running

- Via command line:
    - “name of the solver”, such as: icoFoam, pimpleFoam or… GeN-Foam
- If parallel
    - decomposePar
    - mpirun -np “number of mpi processes” “name of the solver” -parallel
    - reconstructPar

Post-processing

- Typically with paraview
- OpenFOAM also has some mechanisms (not discussed here) to directly output, during or after simulation, specific quantities of interest



How to learn OpenFOAM

- Documentation and tutorials available at:
    - openfoam.com
    - openfoam.org
- Various forums, such as https://www.cfd-online.com/

Take away messages

- Not necessarily an easy tool:
  - Make sure you are relatively familiar with CFD methods, or with numerical methods for PDEs (possibly finite volumes)
  - Make sure you are creating a mesh that is suitable for your problem
  - Don’t forget that initial and boundary conditions determine your solution as much as the equations themselves
  - Go through the OpenFOAM tutorials: don’t try to use complex solvers like GeN-Foam without being familiar with OpenFOAM itself
- Don’t get frustrated: there is always a way out with OpenFOAM and, most likely, someone who had your same problem and will be happy to help
- Don’t get discouraged: the entry barrier may seem steep, but skills you’ll learn will allow you to tackle any kind of problems

EPFL

Thank you

Carlo Fiorina

Ecole Polytechnique
Fédérale de Lausanne
EPFL

Basics of GeN-Foam

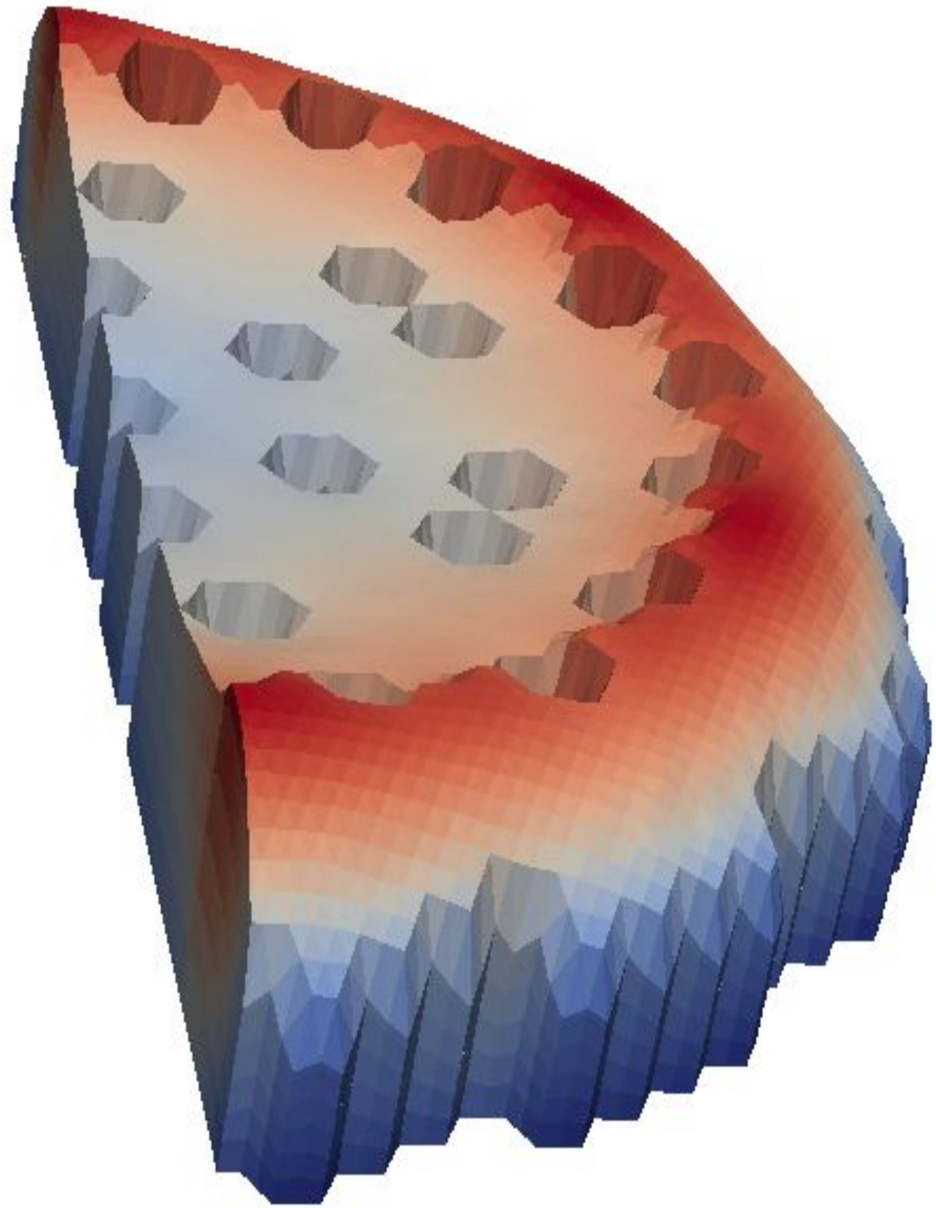
Carlo Fiorina – carlo.fiorina@epfl.ch

Objective of this set of slides:

- Overview of what GeN-Foam does
- Features of GeN-Foam that you do not typically find in standard OpenFOAM solvers
- Some terminology

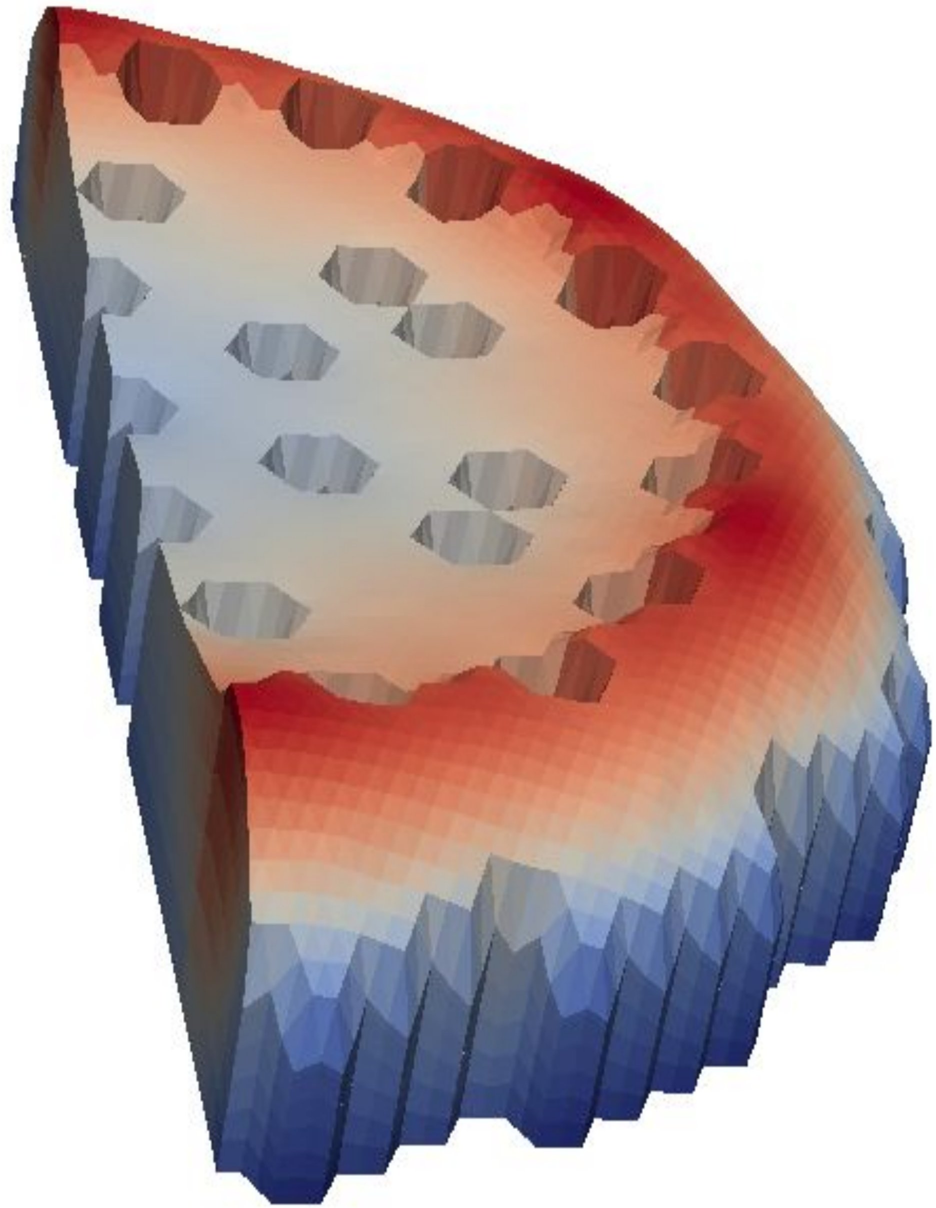
Outline

- Introduction
- Which physics?
- Multi-mesh approach
- Multi-material approach
- Coupling and coupling error
- Time stepping
- The source code



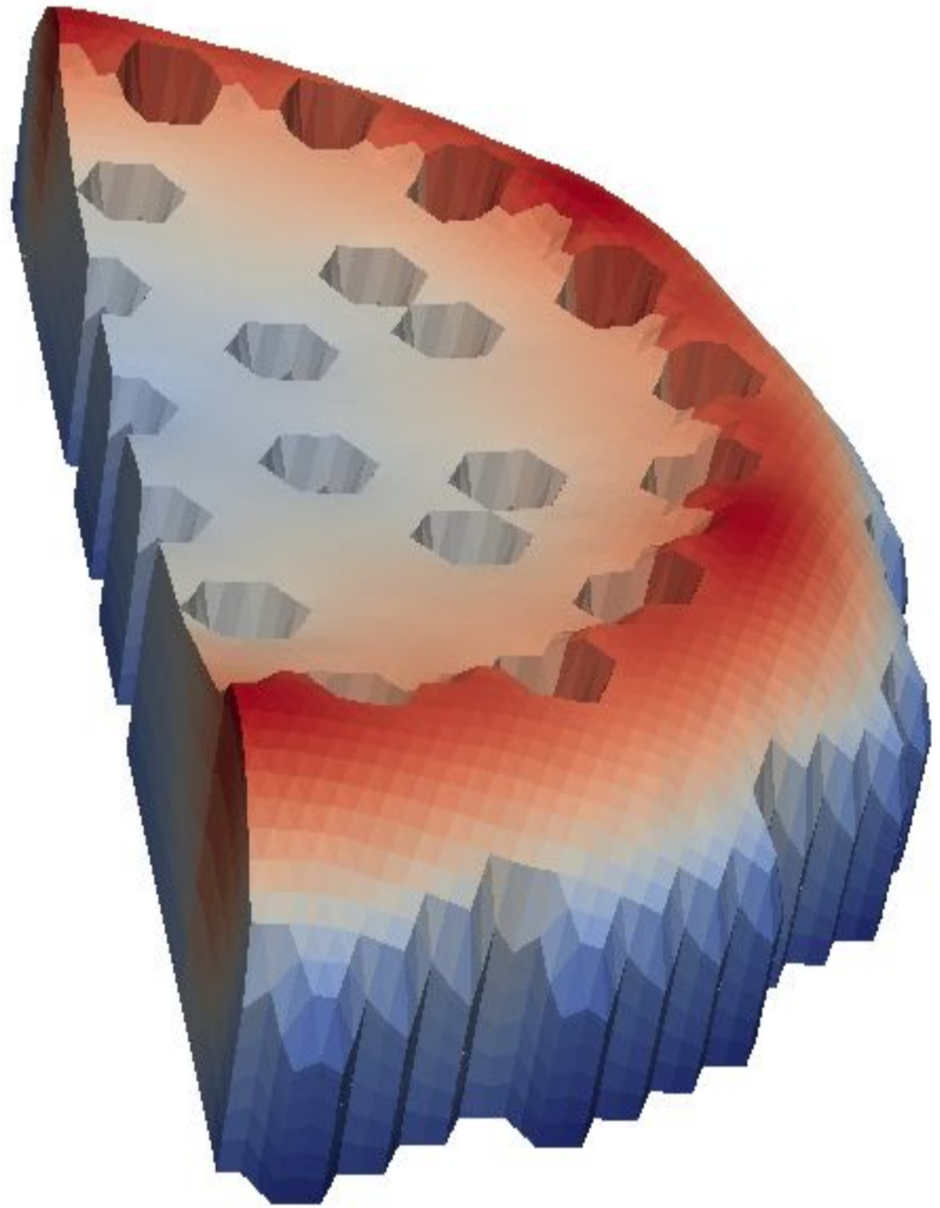
Outline

- Introduction
- Which physics?
- Multi-mesh approach
- Multi-material approach
- Coupling and coupling error
- Time stepping
- The source code



Outline

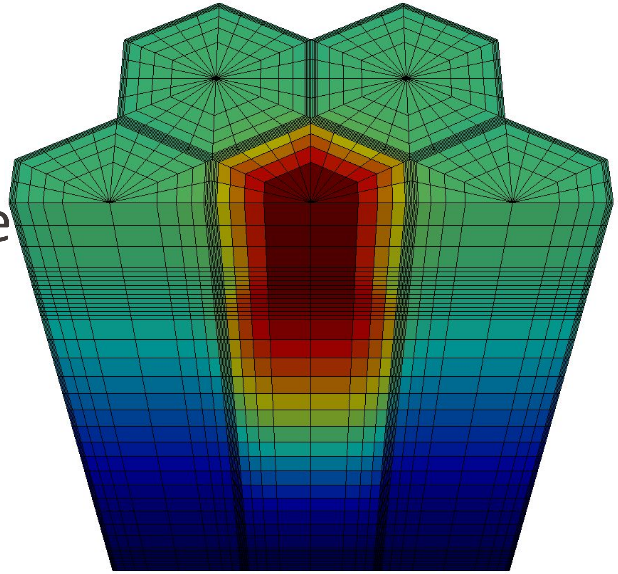
- **Introduction**
- Which physics?
- Multi-mesh approach
- Multi-material approach
- Coupling and coupling error
- Time stepping
- The source code

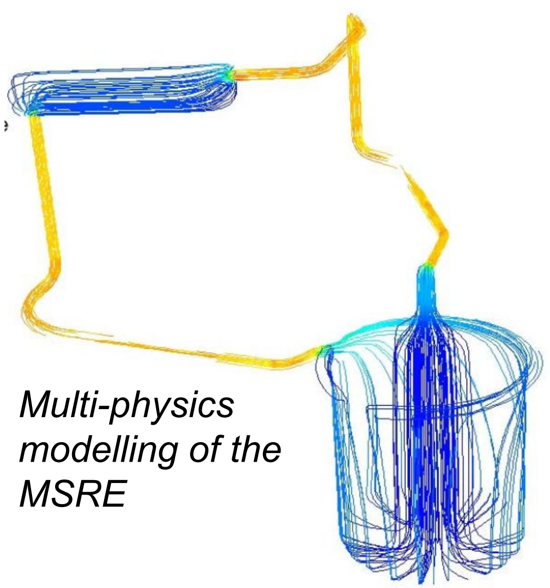


The GeN-Foam multi-physics solver:

- First attempt for a general OpenFOAM-based solver for reactor design and safety analysis
- Objective: complement legacy tool with more flexibility for novel technologies and complex situations
- Main focus on core/primary circuit. Not a system analyst tool!
- On GitLab (foam-for-nuclear/GeN-Foam)

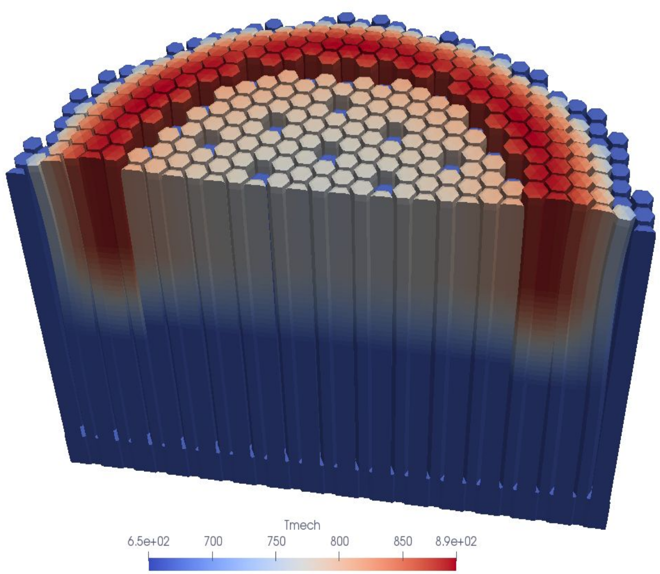
Assembly windows in a SFR



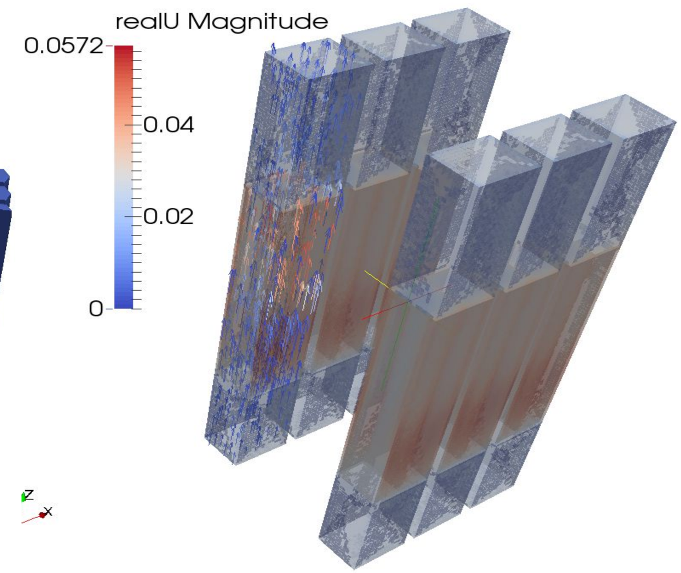


Multi-physics modelling of the MSRE

Core flowering in a SFR

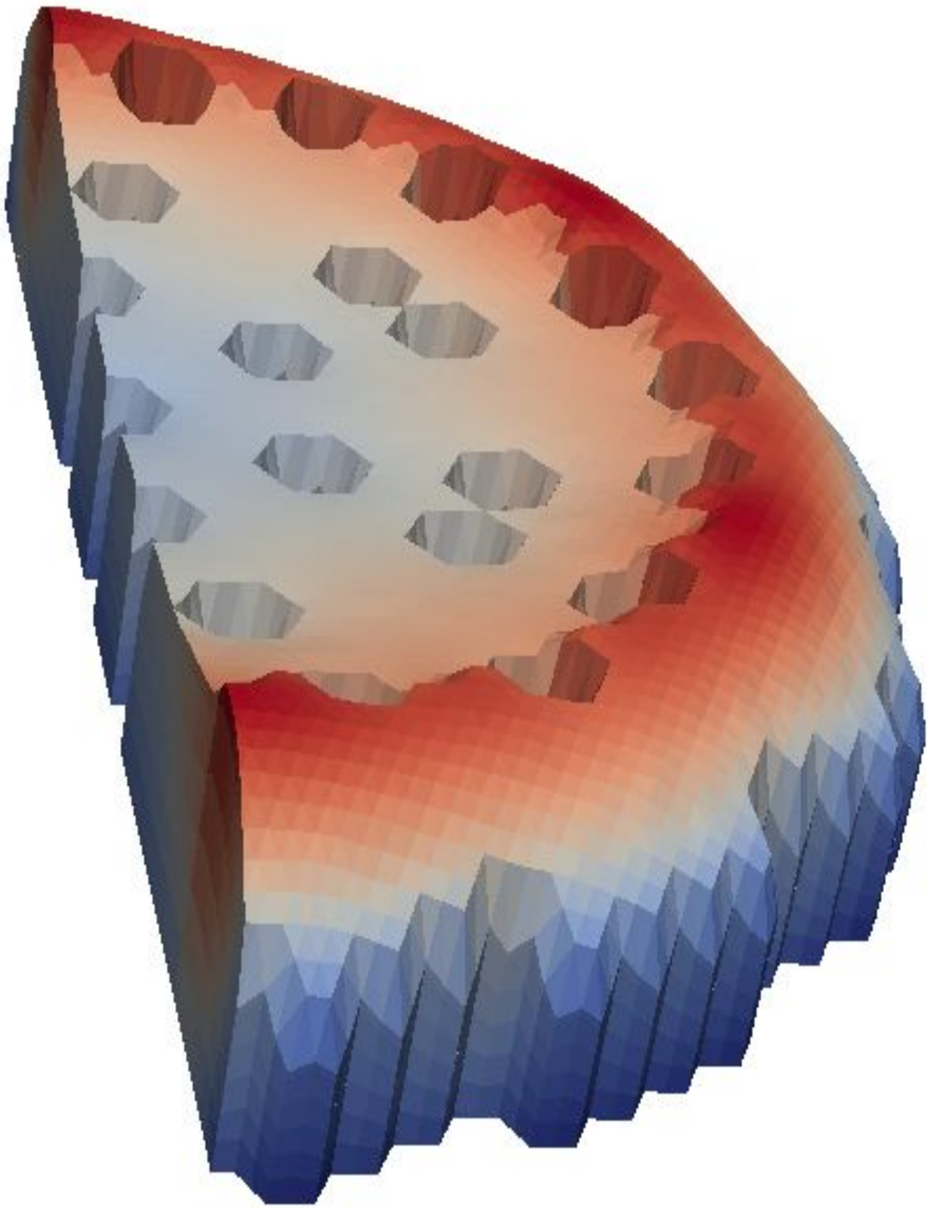


The Argonaut reactor



Outline

- Introduction
- **Which physics?**
- Multi-mesh approach
- Multi-material approach
- Coupling and coupling error
- Time stepping
- The source code

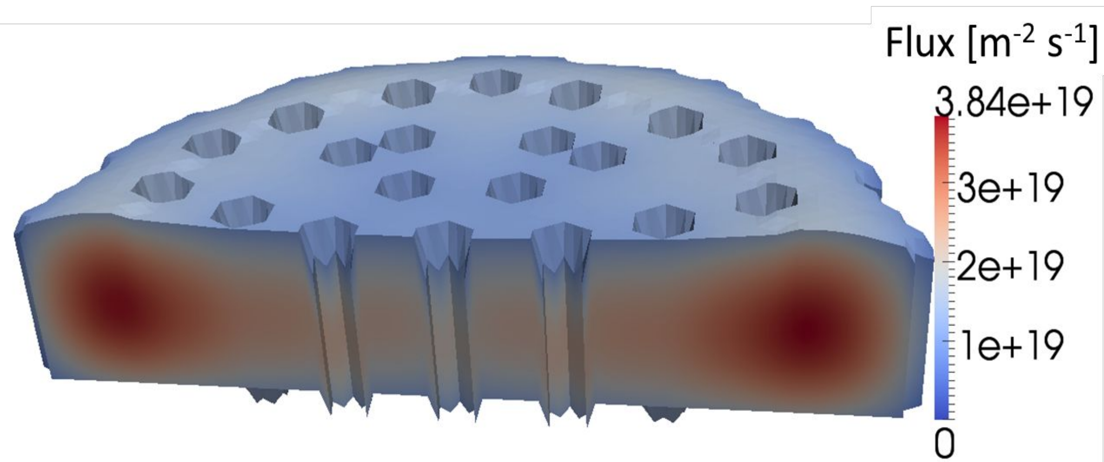


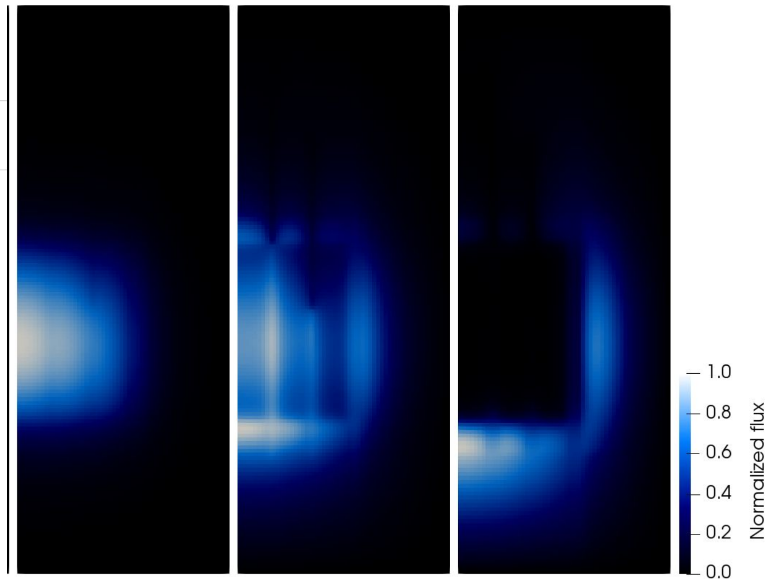
- **Which physics?**
  - ✓ Neutronics
  - ✓ Thermal-hydraulics (+fuel)
  - ✓ Core deformations (mainly for fast-reactors)

- **Which physics?**
  - ✓ Neutronics
  - ✓ Thermal-hydraulics (+fuel)
  - ✓ Core deformations (mainly for fast-reactors)

Neutronics

- **Point kinetics, multi-group diffusion / SP3, SN (preliminary)**
- **Eigenvalue** or **time dependent**
- **Parameterization** in terms of **local** temperatures, densities and deformations
- Possibility to **transport DNPs in MSRs**
- **Mesh can be deformed** according to displacement





Single-phase thermal hydraulics

- Porous-medium
  - ✓ 3-D version of a system code, or
  - ✓ More general version of a sub-channel code
  - ✓ Allows for both coarse-mesh (sub-channel like) and fine-mesh (RANS CFD) solutions on the same mesh

Volume fraction occupied by the fluid

Pressure drops correlations

$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial(\chi\rho\boldsymbol{u})}{\partial t} + \nabla \cdot (\chi\rho\boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot (\mu_t \nabla \boldsymbol{u}) - \nabla(\chi p) + \chi\boldsymbol{F}_g + \chi\boldsymbol{F}_{ss}$$

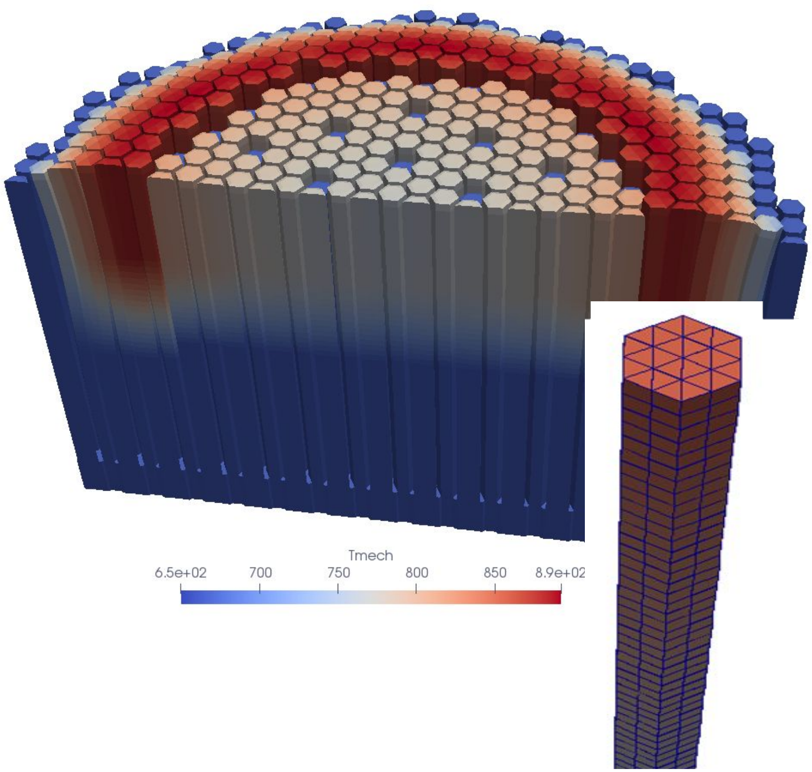
Nusselt number correlations

$$\frac{\partial(\chi\rho e)}{\partial t} + \nabla \cdot \left(\chi\rho\boldsymbol{u}\left(e + \frac{p}{\rho}\right)\right) = \nabla \cdot (\chi k_t \nabla T) + \boldsymbol{F}_{ss} \cdot \boldsymbol{u} + \chi\dot{Q}$$

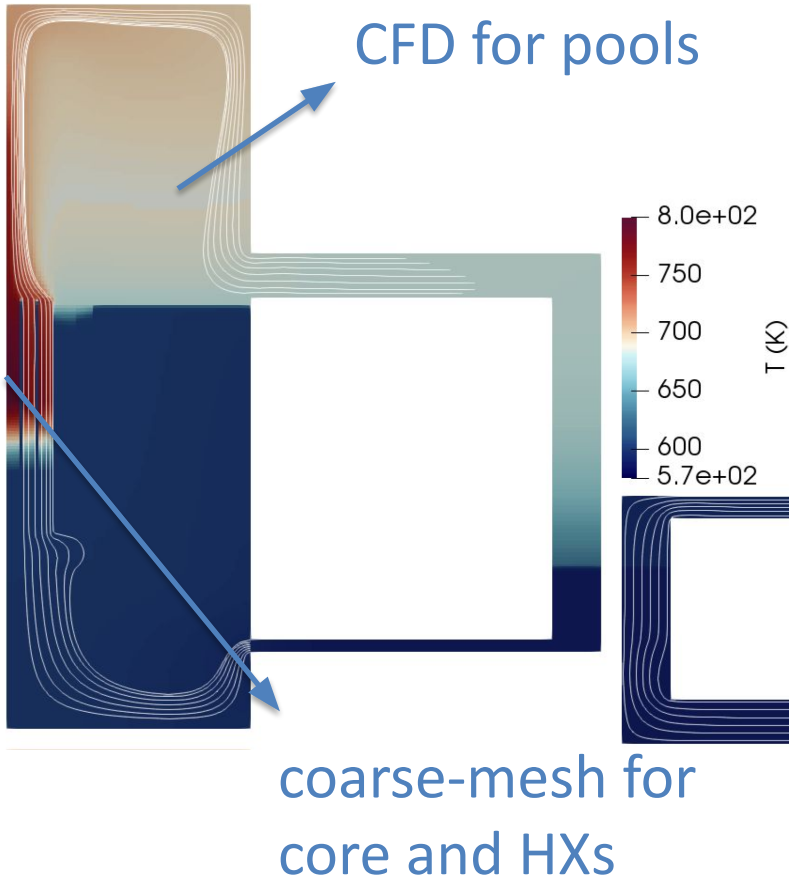
Single-phase thermal hydraulics

- Examples

3-D coarse mesh simulation of a SFR core

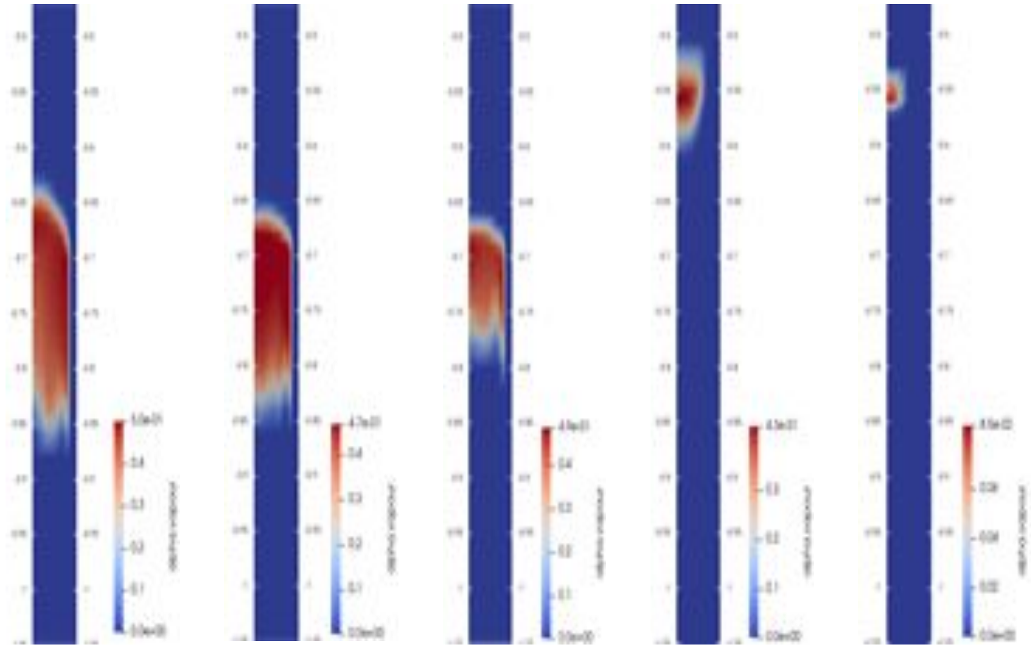


2-D combined coarse/fine mesh simulation of the Fast Flux Test Facility



Two-phase thermal hydraulics

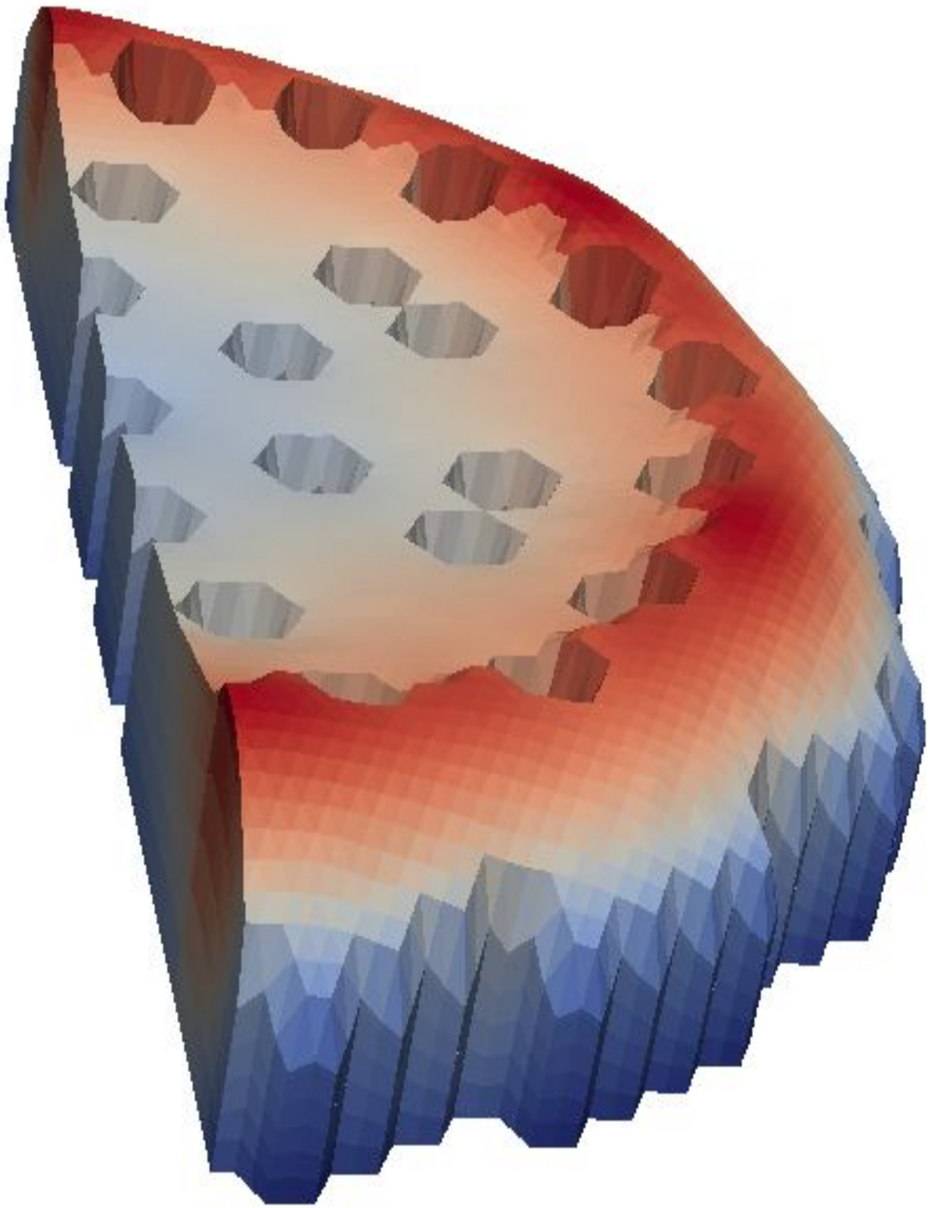
- Same approach as for single-phase thermal-hydraulics (porous-medium with sub-scale structure)
- Beyond the scope of this presentation. Further info in the EPFL PhD thesis of Stefan Radman (see reference list at the end of the presentation)



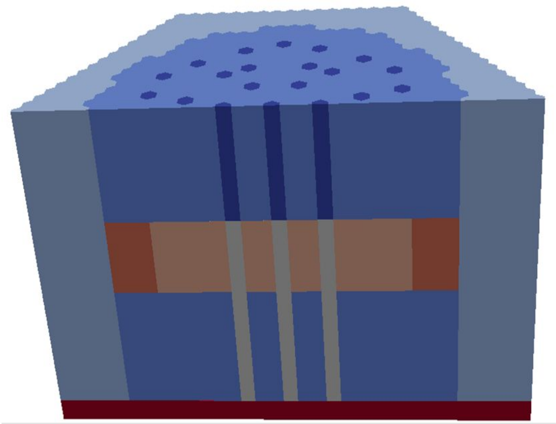
2-D coarse mesh simulation of a SFR assembly with windows

Outline

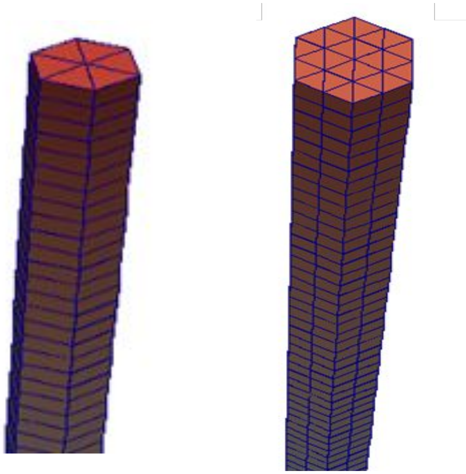
- Introduction
- Which physics?
- **Multi-mesh approach**
- Multi-material approach
- Coupling and coupling error
- Time stepping
- The source code

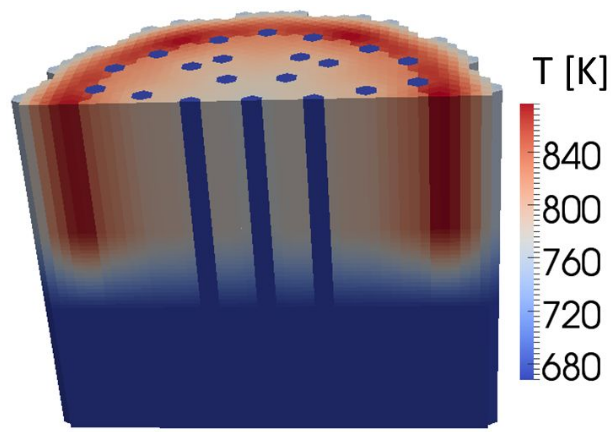


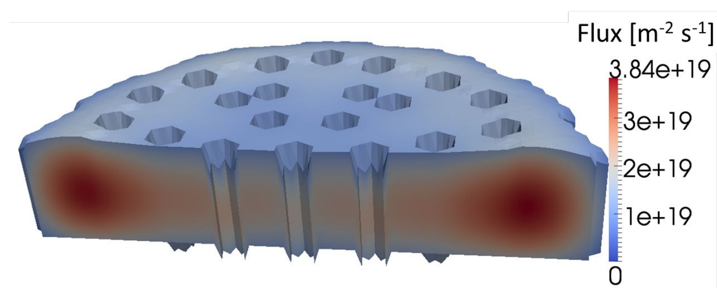
Multi-mesh



- Problem: different meshes for different "physics"
- Solution: multi-mesh (called multi-region in OpenFOAM)
- One mesh for each "physics"
- (Projection of fields from one mesh to the other for coupling)







Multi-mesh: in practice

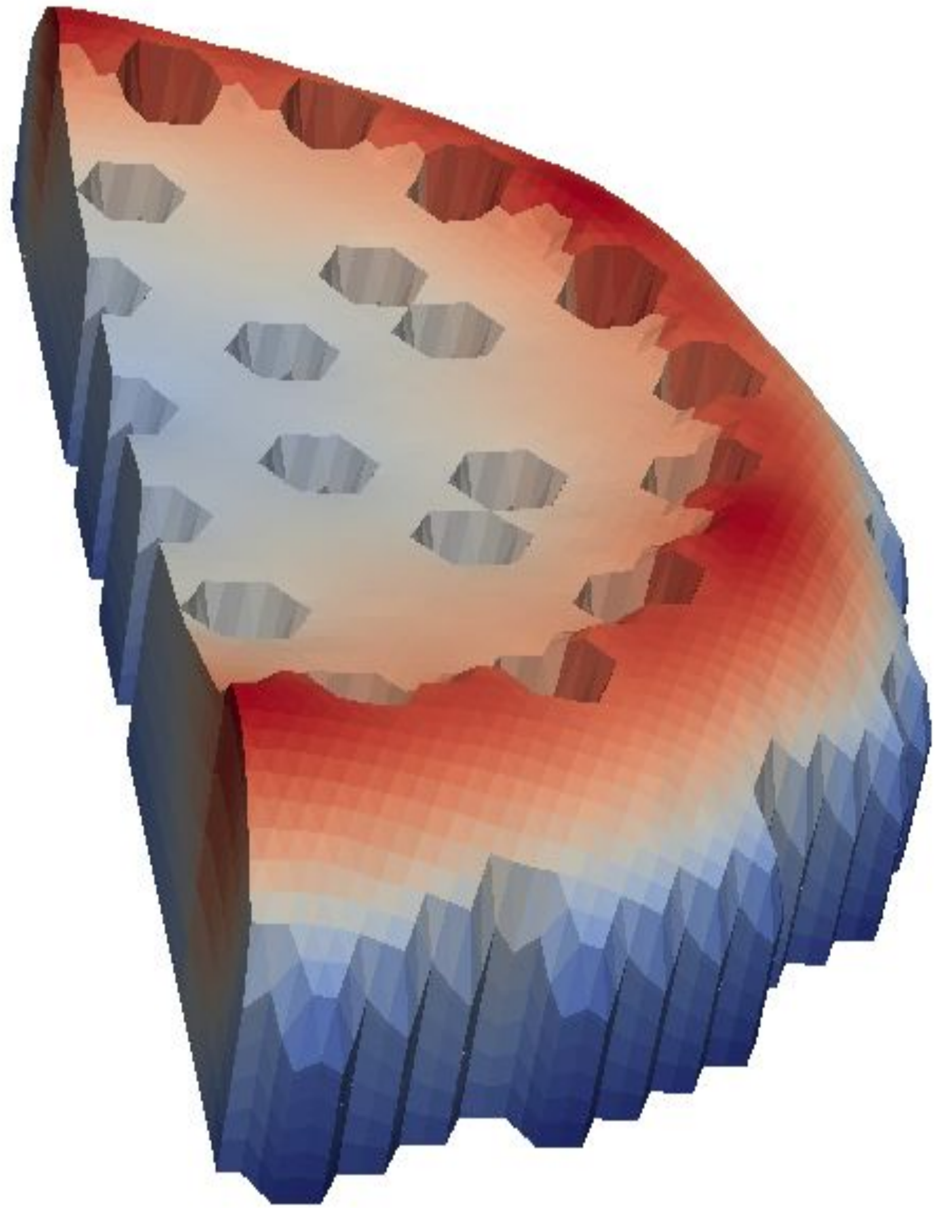
- **Case folder:**

- Case
  - 0
    - U
    - T
    - ...
  - constant
    - turbulenceProperties
    - ...
  - system
    - fvSolution
    - fvSchemes
    - controDict

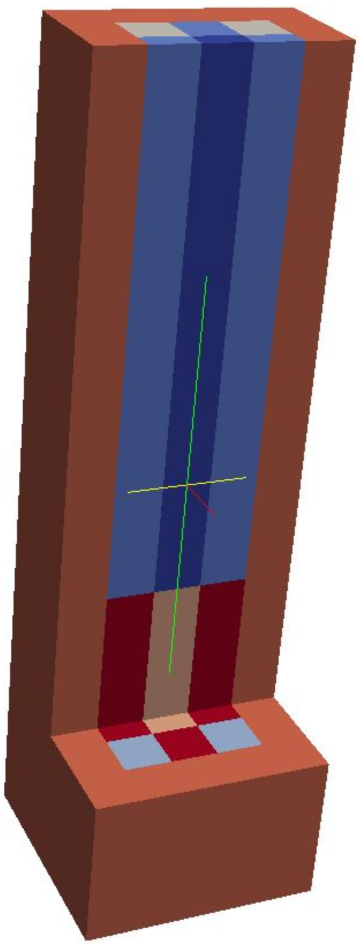
- Case
  - 0
    - neutroRegion
      - Flux
      - ...
    - fluidRegion
      - U
      - ...
    - thermoMechanicalRegion
      - ...
  - constant
    - neutroRegion
    - fluidRegion
    - thermoMechanicalRegion
    - ...
  - system
    - neutroRegion
    - fluidRegion
    - thermoMechanicalRegion
    - ...

Outline

- Introduction
- Which physics?
- Multi-mesh approach
- **Multi-material approach**
- Coupling and coupling error
- Time stepping
- The source code



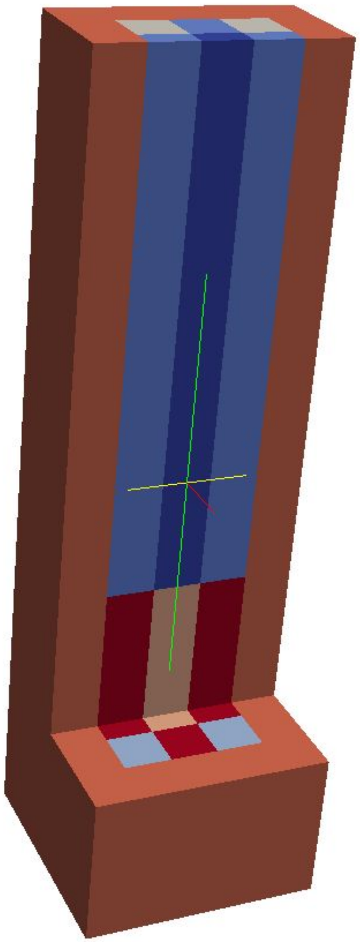
Multi-material



- Problem: one mesh, multiple material
- Solutions: cellZones
  - associate a label to each cell in polymesh/cellZones

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       regIOobject;
    location    "constant/fluid/polyMesh";
    object      cellZones;
}
// * * * * * * * * * * * * * * * * * * * * *

7
(
controlRod
{
    type cellZone;
cellLabels      List<label>
5994
(
0
1
2
```

Multi-material



- Then, for each physics, an input file (dictionary) is used that associates each of these labels with a set of properties. For instance in /constant/neutroRegion/nuclearData

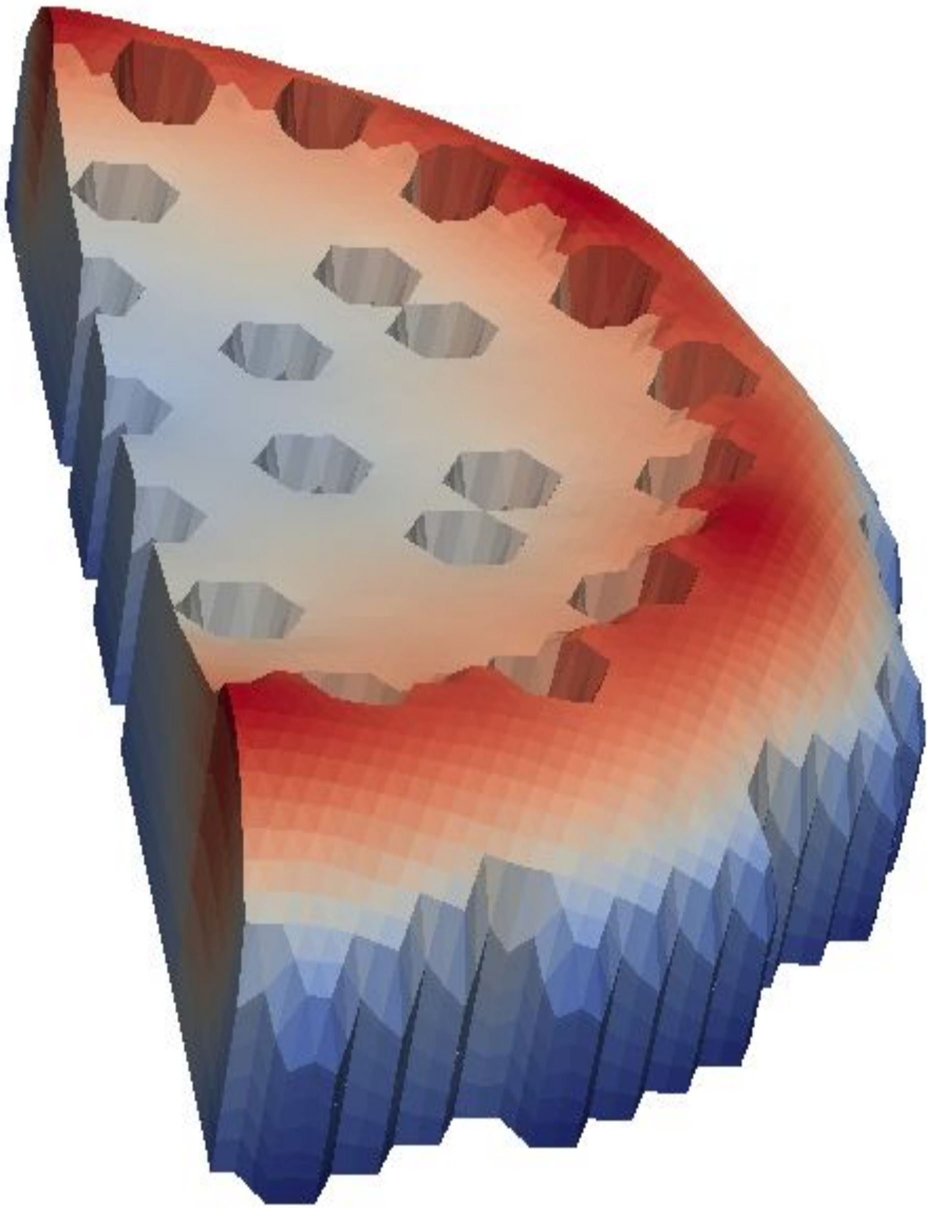
```
zones
 (

controlRod
{
 fuelFraction 1.000000e+00 ;
 IV nonuniform List<scalar> 1 (8.477550e-07  );
 D nonuniform List<scalar> 1 (1.562700e-02  );
 nuSigmaEff nonuniform List<scalar> 1 (0.000000e+00  );
 sigmaPow nonuniform List<scalar> 1 (0.000000e+00  );
 scatteringMatrix  1  1 (
 ( 2.509070e+01 )
 );
```

Multi-material: in practice

- **How to create a multi-zone mesh:**
  - ✓ All mesh generators allows for the option to generate "cellZones"
  - ✓ NB: cellZones are called in different ways (physical volumes in gmsh, groups in Salome, etc)
  - ✓ The mesh conversion tool (e.g., gmshToFoam) takes care of converting the format
- **Case folder:**
  - ✓ Polymesh folder including cellZones
  - ✓ Dictionaries that associates a cellZone to some value of a field or property

Outline

- Introduction
- Which physics?
- Multi-mesh approach
- Multi-material approach
- **Coupling and coupling error**
- Time stepping
- The source code



Coupling and coupling error

- **Two main options**
  - ✓ **Matrix-coupled solution:**
    - All equations in the same matrix
    - Straightforward, no iterations needed, can be faster
    - Often difficult to precondition
  - ✓ **Segregated solution (operator splitting):**
    - One matrix for each equation + iteration
    - Easier preconditioning and optimal choice of solution method
    - No need to solve all physics at each coupling/time step
  - ✓ **Possibility to combine the two**
    - e.g., segregated for each mesh, coupled for some physics (multi-group diffusion)

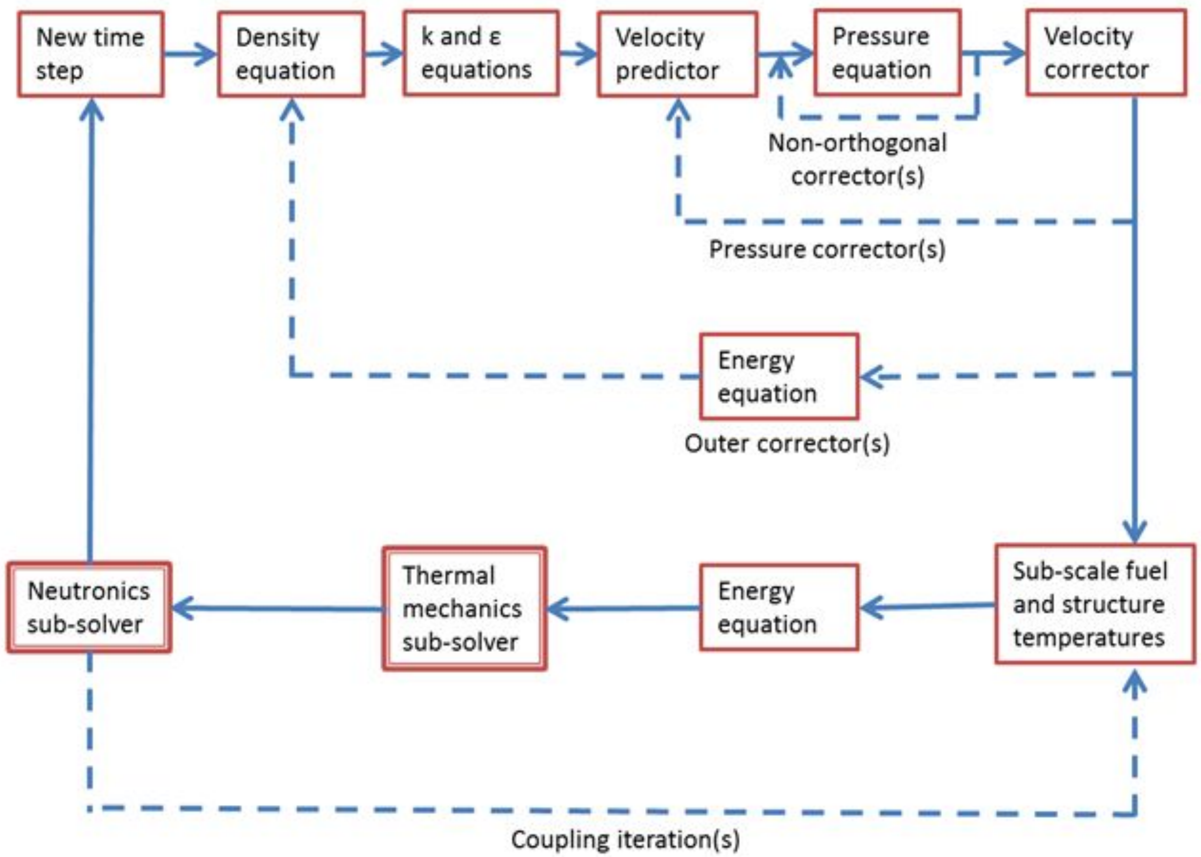
Coupling and coupling error

- Coupled solution not (fully) available in OpenFOAM

- Segregated solution employed for GeN-Foam

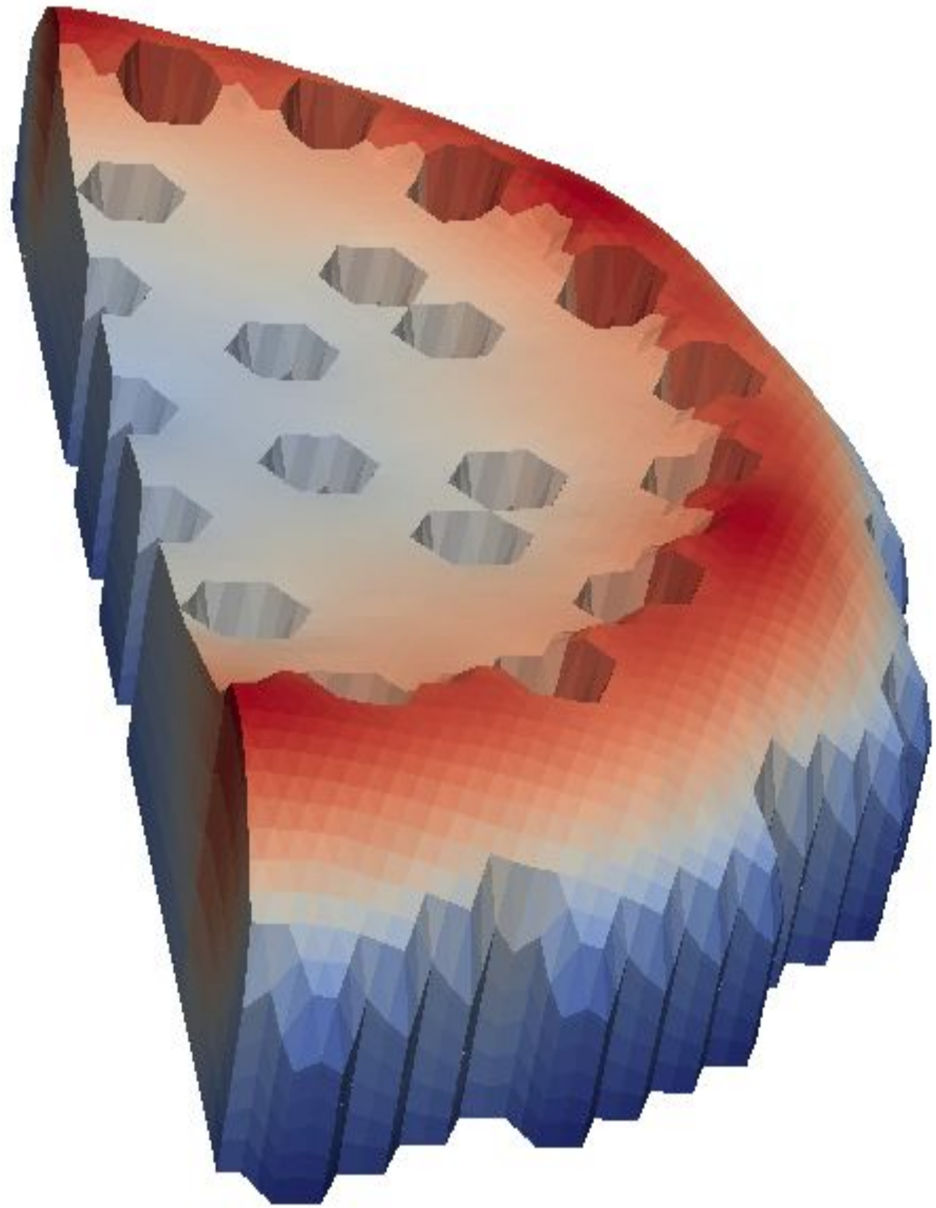
Coupling and coupling error: coupling strategy (single phase)

- Initial residuals evaluated before solving each iteration
- Solve only if initial residuals higher than desired residuals for each physics
- Keep iterating till initial residual for each physics is lower than the desired coupling residual



Outline

- Introduction
- Which physics?
- Multi-mesh approach
- Multi-material approach
- Coupling and coupling error
- **Time stepping**
- The source code



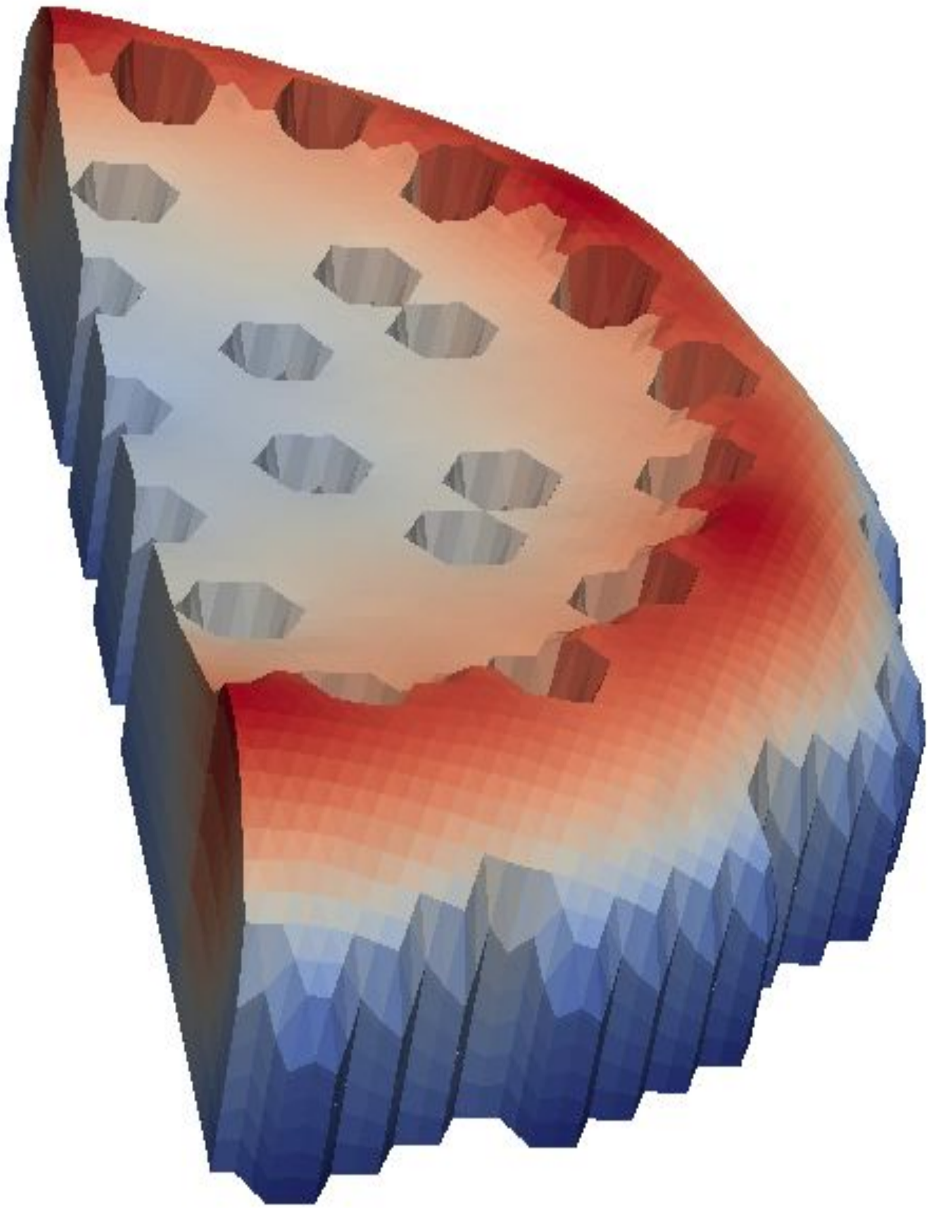
Time stepping: choice of time step

- **Fixed or adaptive**
- **Two possibilities for adaptive time step:**
  - ✓ Based on **previous iteration**
  - ✓ Based on **current iteration,** more stable, more time consuming
- **Criteria for adaptive time step:**
  - ✓ Based on **stability criteria**: e.g. CFL condition
  - ✓ Based on **"physics based"** criteria: e.g., max power variation
  - ✓ Based on **rigorous mathematical approaches**: e.g., in high-order time discretization schemes, based on the difference between solutions for two different orders

- **In GeN-Foam:**
  - ✓ Fixed or adaptive; adaptive based on CFL + max power variation, using values at previous time step; necessary to guess first one

Outline

- Introduction
- Which physics?
- Multi-mesh approach
- Multi-material approach
- Coupling and coupling error
- Time stepping
- **The source code**



The GeN-Foam multi-physics solver

- All sub-solvers are organized into C++ classes
  - Easier to understand its coding
  - Possible to easily extract sub-solvers for use in other solvers
    - ✓ You have complete freedom to freely use and modify
    - ✓ (Does not mean that copyright does not exists: acknowledgment of previous the work of other authors is always good practice and consistent with ethics in open-source development)

What is a C++ class

- C++ is object oriented
- Object-oriented roughly means that you can organize you code into classes
- Classes are a set of data, and functions that operate on those data
- For instance, in GeN-Foam, classes for:
  - neutronics
  - cross-sections
  - thermal-hydraulics
  - thermal-mechanics
  - other “functional classes” e.g. for handling multi-physics simulations
- For instance, the neutronics class contains:
  - neutronics quantities, such as keff, power field, etc.
  - functions that manipulate these quantities

What is a C++ class

- Classes can have *derived classes*, i.e., classes that can "see" everything in the original class, but that contains additional data and functions
- In GeN-Foam, this is used to "specialize" solver classes into sub-solvers
- For instance, from the neutronic class, we derive classes for:
    - diffusion
    - Sp3
    - SN
    - point-kinetics
- For instance, the "diffusion" derived class contains:
    - all data and functions from the neutronics class
    - additional data (e.g., multi-group fluxes)
    - additional functions, the most important being the function that solves for the fluxes at every time step

Thank you for your attention

Carlo Fiorina carlo.fiorina@epfl.ch

Useful literature / documentation on GeN-Foam

- General
  - C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, “GeN-Foam: A novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors,” Nuclear Engineering and Design, vol. 294, 2015, doi: 10.1016/j.nucengdes.2015.05.035.
  - Intro (theory and practice): https://gitlab.com/foam-for-nuclear/GeN-Foam/-/tree/master/Documentation
  - Wiki documentation: https://gitlab.com/foam-for-nuclear/GeN-Foam/-/wikis/home
  - Commented tutorials: https://gitlab.com/foam-for-nuclear/GeN-Foam/-/tree/master/Tutorials

Useful literature / documentation on GeN-Foam

- **Neutronics**
  - Diffusion
    - ✓ C. Fiorina, N. Kerkar, K. Mikityuk, P. Rubiolo, and A. Pautz, "Development and verification of the neutron diffusion solver for the GeN-Foam multi-physics platform," Annals of Nuclear Energy, vol. 96, 2016, doi: 10.1016/j.anucene.2016.05.023.
  - SP3
    - ✓ C. Fiorina, M. Hursin, and A. Pautz, "Extension of the GeN-Foam neutronic solver to SP3 analysis and application to the CROCUS experimental reactor," Annals of Nuclear Energy, vol. 101, pp. 419–428, Mar. 2017, doi: 10.1016/j.anucene.2016.11.042.
  - SN
    - ✓ C. Fiorina, S. Radman, M.-Z. Koc, and A. Pautz, "Detailed modelling of the expansion reactivity feedback in fast reactors using OpenFoam," M&C International Conference, Portland (US), 2019.
  - Point kinetics
    - ✓ S. Radman, "A coarse-mesh methodology for the analysis of one and two-phase nuclear reactor thermal-hydraulics in a multi-physics context," 2021.
    - ✓ A.S. Mattioli et al., "Derivation and implementation in OpenFOAM of a point-kinetics model for Molten Salt Reactors". Submitted to the ANS winter meeting 2021.

Useful literature / documentation on GeN-Foam

- Single-phase thermal-hydraulics
  - C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, “GeN-Foam: A novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors,” Nuclear Engineering and Design, vol. 294, 2015, doi: 10.1016/j.nucengdes.2015.05.035.
  - S. Radman, “A coarse-mesh methodology for the analysis of one and two-phase nuclear reactor thermal-hydraulics in a multi-physics context,” PhD Thesis, EPFL, Switzerland 2021, https://memento.epfl.ch/event/a-coarse-mesh-methodology-for-the-analysis-of-on-2/ .
  - S. Radman et al., “A coarse-mesh methodology for modelling of single-phase thermal-hydraulics of ESFR innovative assembly design,” Nuclear Engineering and Design 355, 2019.

Useful literature / documentation on GeN-Foam

- Two-phase thermal-hydraulics
  - S. Radman, "A coarse-mesh methodology for the analysis of one and two-phase nuclear reactor thermal-hydraulics in a multi-physics context," PhD Thesis, EPFL, Switzerland 2021, https://memento.epfl.ch/event/a-coarse-mesh-methodology-for-the-analysis-of-on-2/ .
  - S. Radman, C. Fiorina, and A. Pautz, "Development of a Novel Two-Phase Flow Solver for Nuclear Reactor Analysis: Algorithms, Verification and Implementation in OpenFOAM," Submitted to Nuclear Engineering and Design, 2021.
  - S. Radman, C. Fiorina, and A. Pautz, "Development of a Novel Two-Phase Flow Solver for NuclearReactor Analysis: Validation against Sodium Boiling Experiments," Nuclear Engineering and Design, Submitted, 2021.

Useful literature / documentation on GeN-Foam

- Mesh deformation
  - C. Fiorina, I. Clifford, M. Aufiero, and K. Mikityuk, “GeN-Foam: A novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors,” Nuclear Engineering and Design, vol. 294, 2015, doi: 10.1016/j.nucengdes.2015.05.035.
  - C. Fiorina, S. Radman, M.-Z. Koc, and A. Pautz, “Detailed modelling of the expansion reactivity feedback in fast reactors using OpenFoam,” 2019.

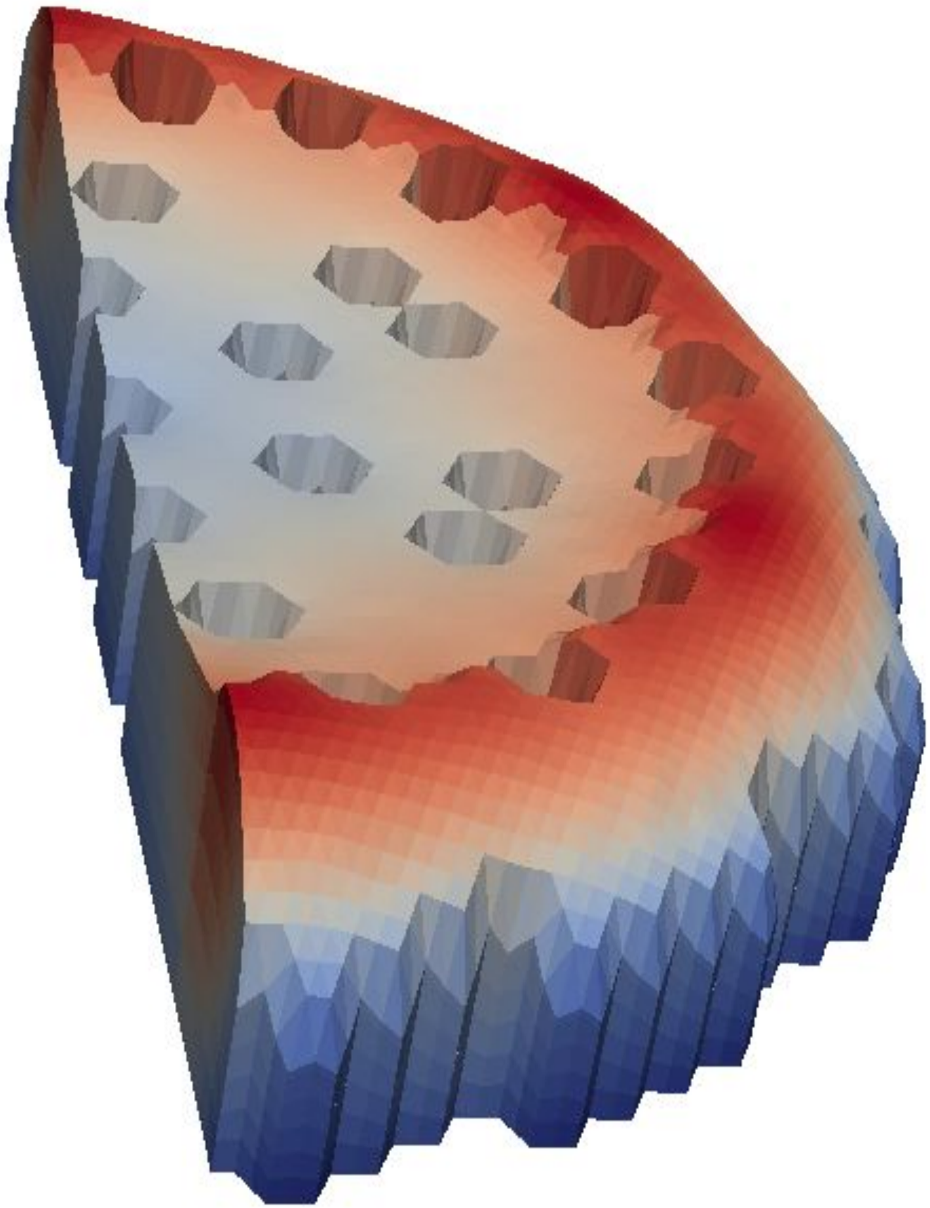
Ecole Polytechnique
Fédérale de Lausanne
EPFL

A very short introduction to using GeN-Foam

Carlo Fiorina – carlo.fiorina@epfl.ch

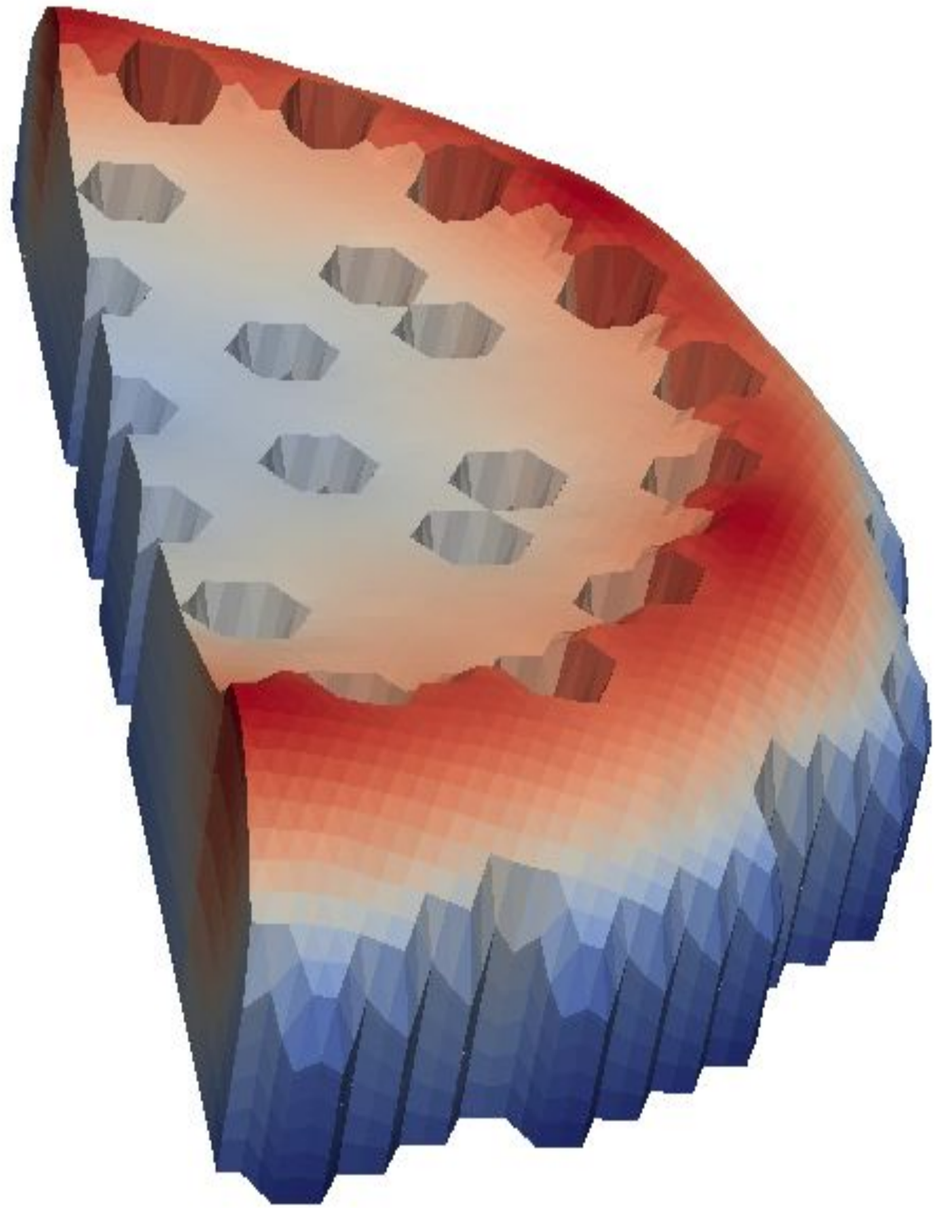
Outline

- How to get it?
- How to install it?
- What's inside?
- An example
- Some other tutorials



Outline

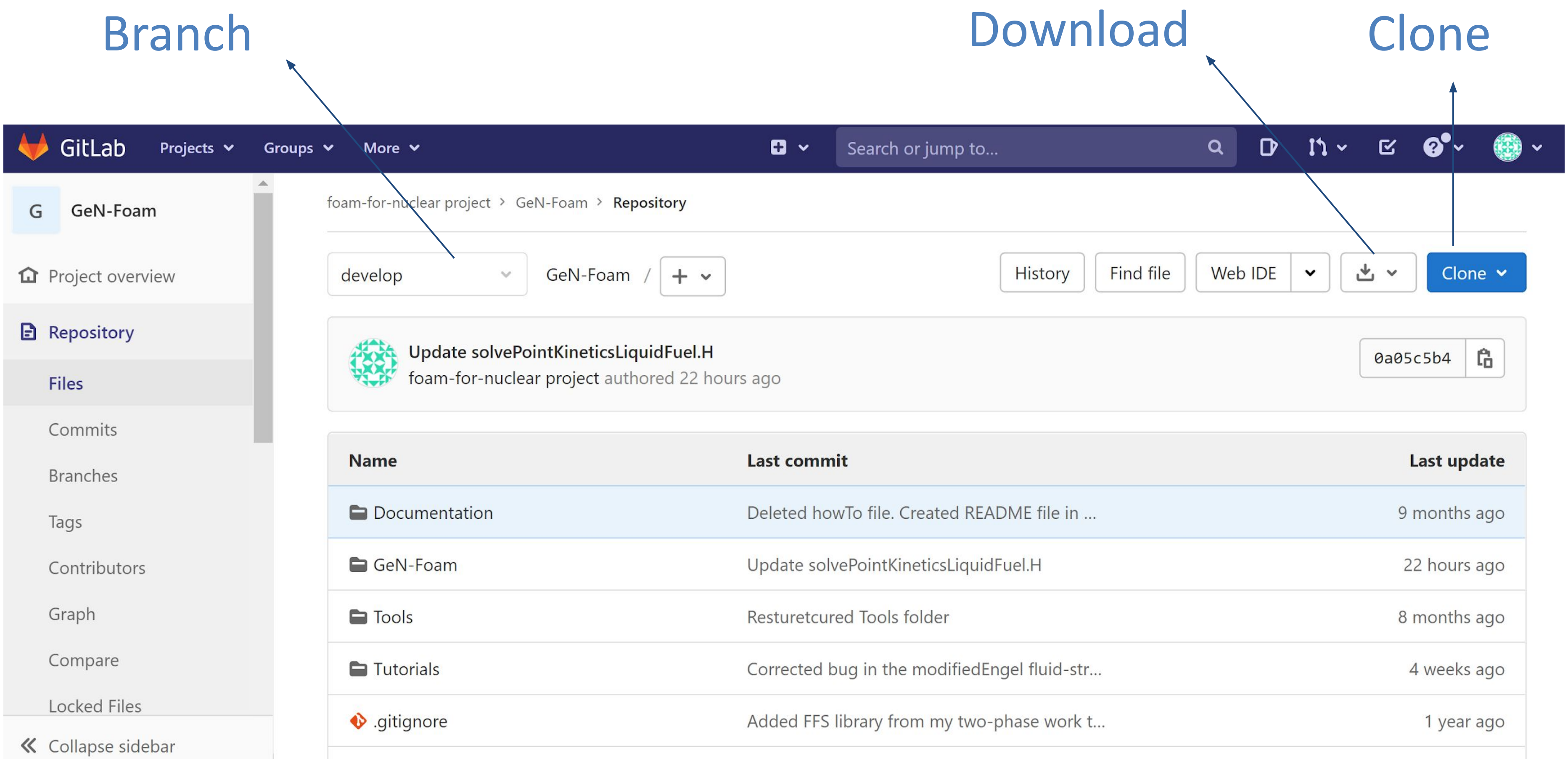
- **How to get it?**
- How to install it?
- What’s inside?
- An example
- Some other tutorials



How to get it?

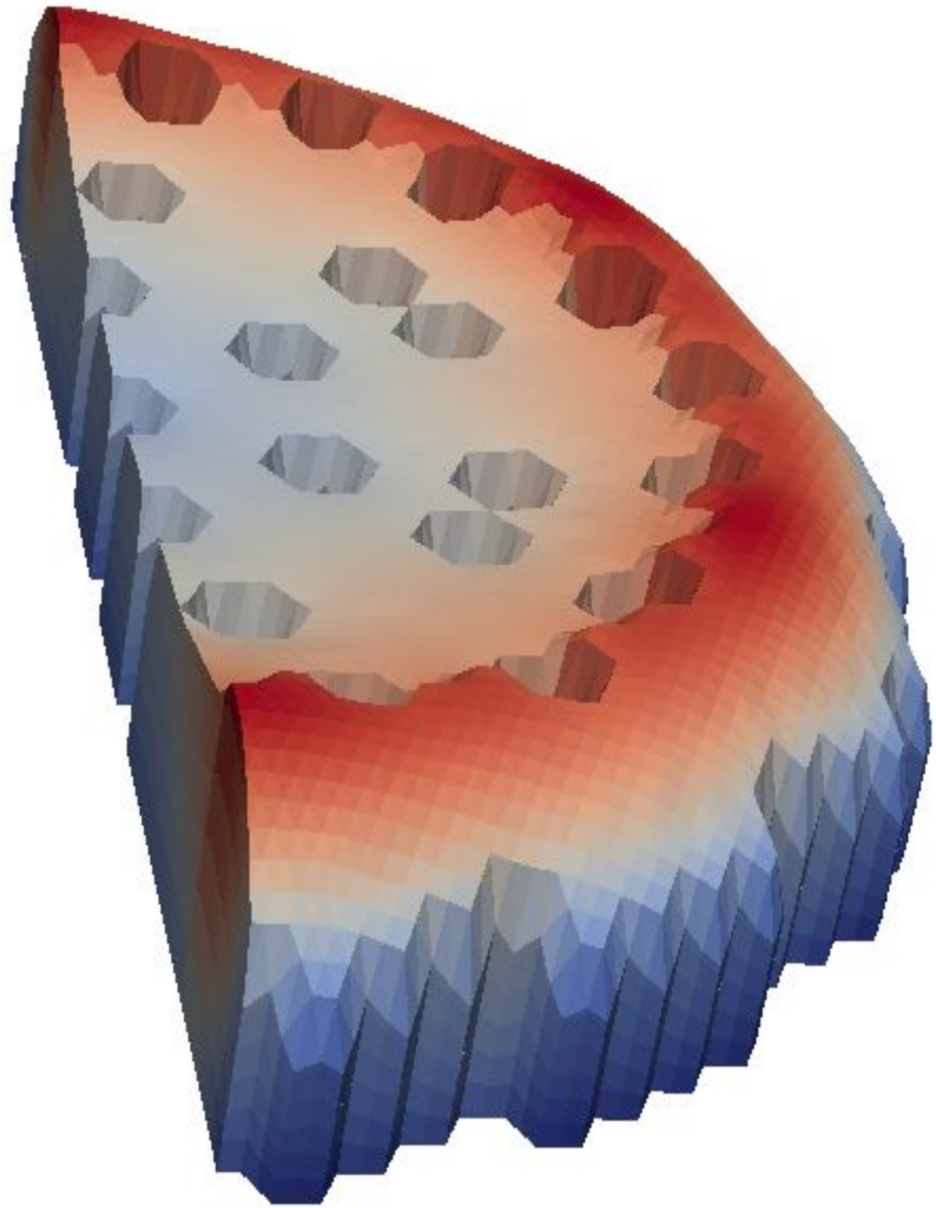
- **Free, online at** https://gitlab.com/foam-for-nuclear/GeN-Foam/-/tree/develop
  - "Develop" branch or "Master" branch
  - Either
    - git clone https://gitlab.com/foam-for-nuclear/GeN-Foam.git"
    - or, simply download

How to get it?


Branch
Download
Clone
GitLab
Projects
Groups
More
Search or jump to...
GeN-Foam
Project overview
Repository
Files
Commits
Branches
Tags
Contributors
Graph
Compare
Locked Files
Collapse sidebar
foam-for-nuclear project › GeN-Foam › Repository
develop
GeN-Foam /
History
Find file
Web IDE
Clone
Update solvePointKineticsLiquidFuel.H
foam-for-nuclear project authored 22 hours ago
0a05c5b4
Name
Last commit
Last update
Documentation
Deleted howTo file. Created README file in ...
9 months ago
GeN-Foam
Update solvePointKineticsLiquidFuel.H
22 hours ago
Tools
Resturetcured Tools folder
8 months ago
Tutorials
Corrected bug in the modifiedEngel fluid-str...
4 weeks ago
.gitignore
Added FFS library from my two-phase work t...
1 year ago

Outline

- How to get it?
- **How to install it?**
- What's inside?
- An example
- Some other tutorials

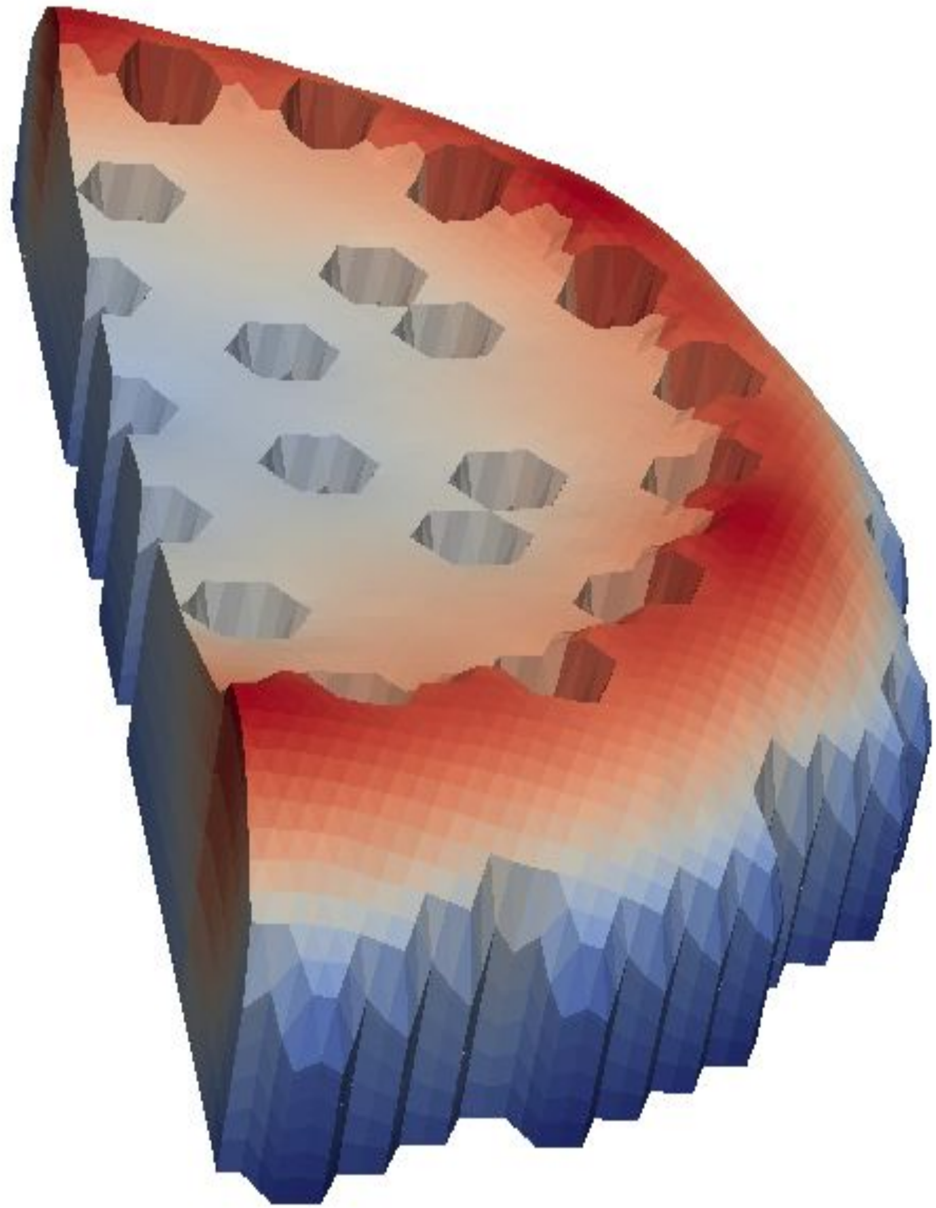


How to install it?

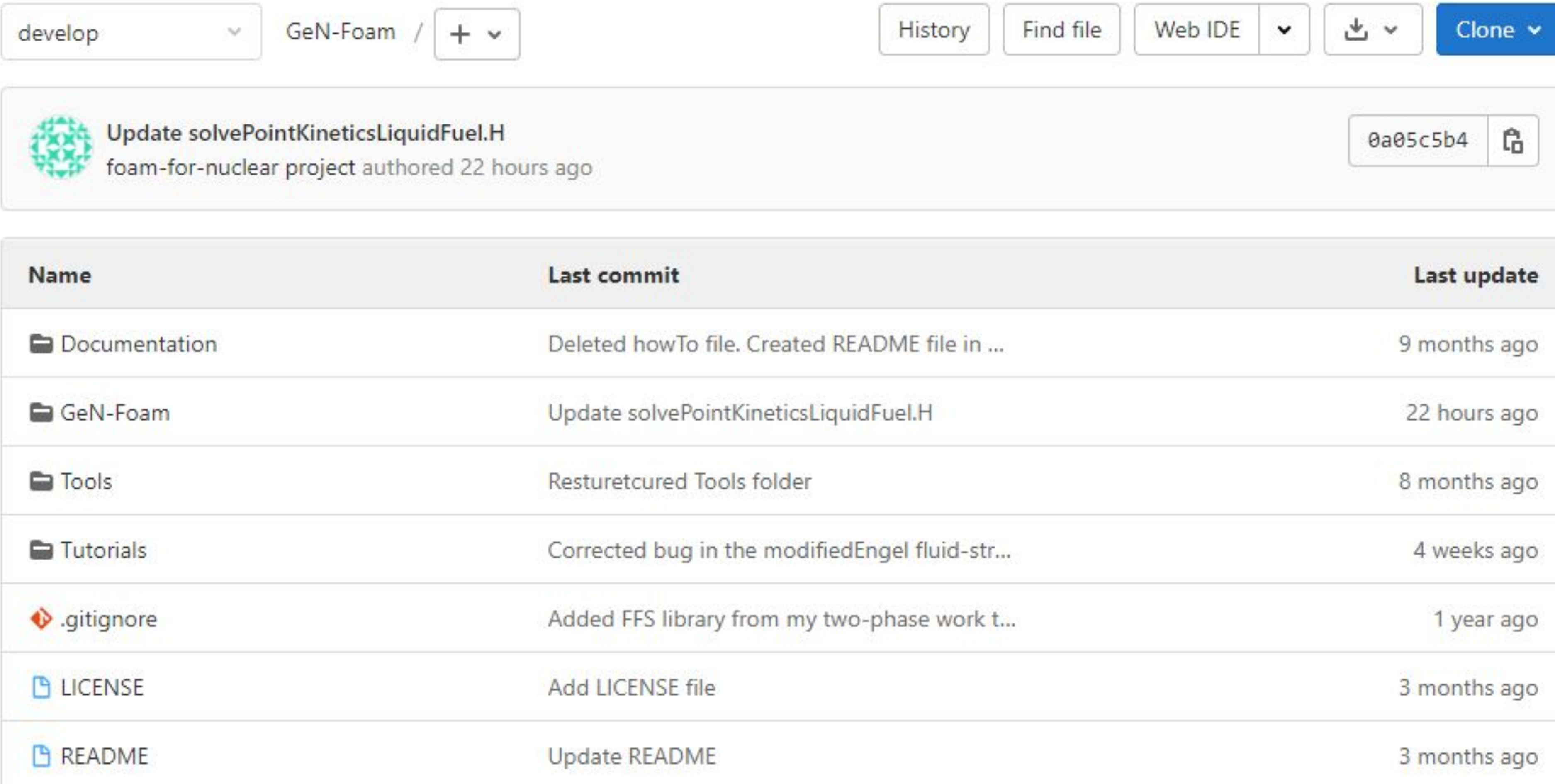
- Download OpenFOAM at
  - https://www.openfoam.com/download/
  - (Typically the latest release, but it may take us some few weeks to update to a new release each time)
- Install OpenFOAM and prepare the environment
  - https://www.openfoam.com/download/installation.php
- Download GeN-Foam
- Enter the GeN-Foam/GeN-Foam folder and run:
  - *Allwclean*
  - *Allwmake* (or *Allwmake -j*, to compile in parallel)
- Testing - enter any tutorial and run:
  - *Allrun*

Outline

- How to get it?
- How to install it?
- **What's inside?**
- An example
- Some other tutorials

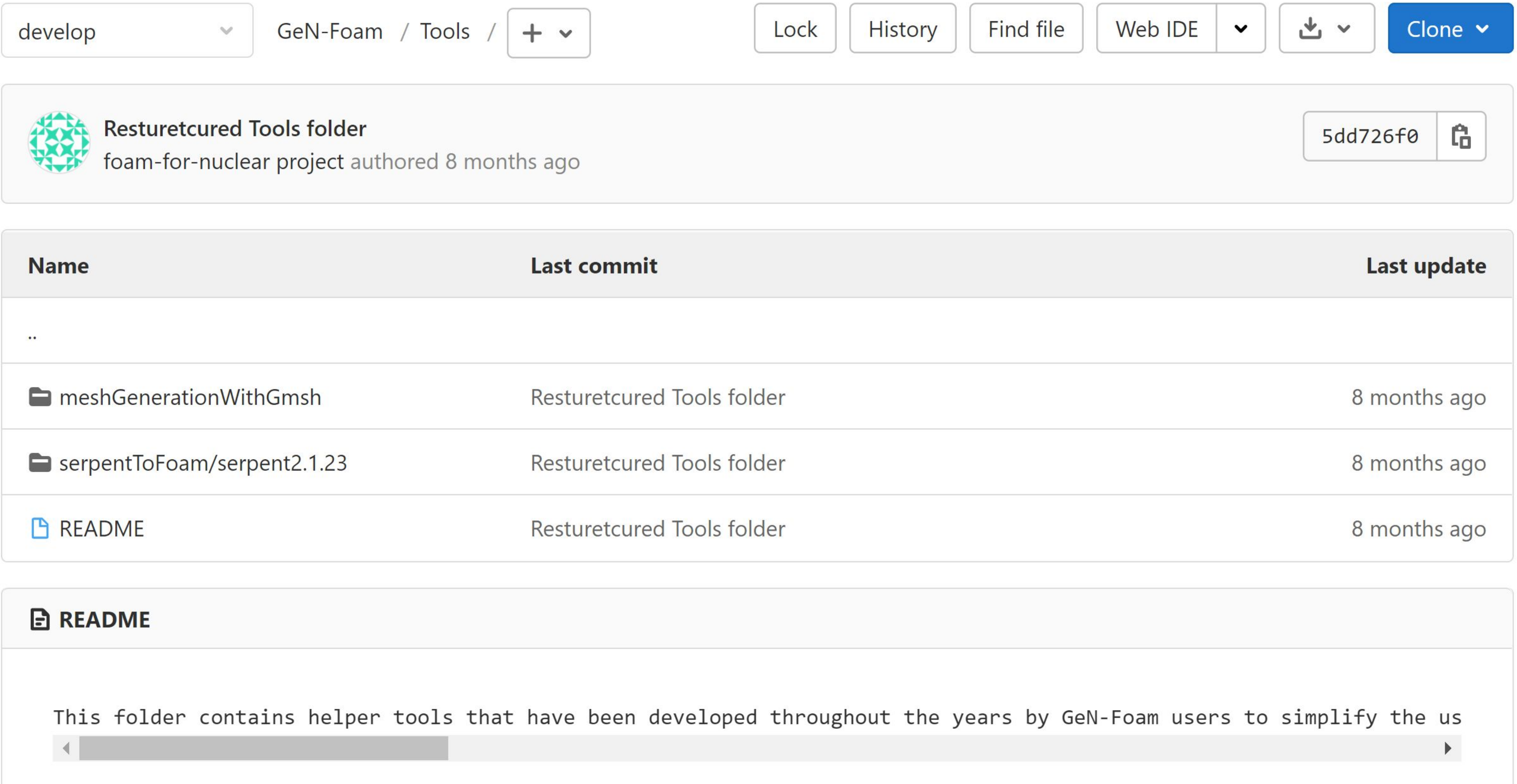


What's inside?



- README file often present to describe what's in a subfolder

What's inside? Tools



- Helper tools to make life of a user easier
  - Example of a mesh creation with gmsh
  - Script to convert an output of Serpent into an input for GeN-Foam

What's inside? Documentation

develop | GeN-Foam / Documentation / | + | Lock | History | Find file | Web IDE | Clone

Update README in GeN-Foam/Documentation
foam-for-nuclear project authored 5 months ago

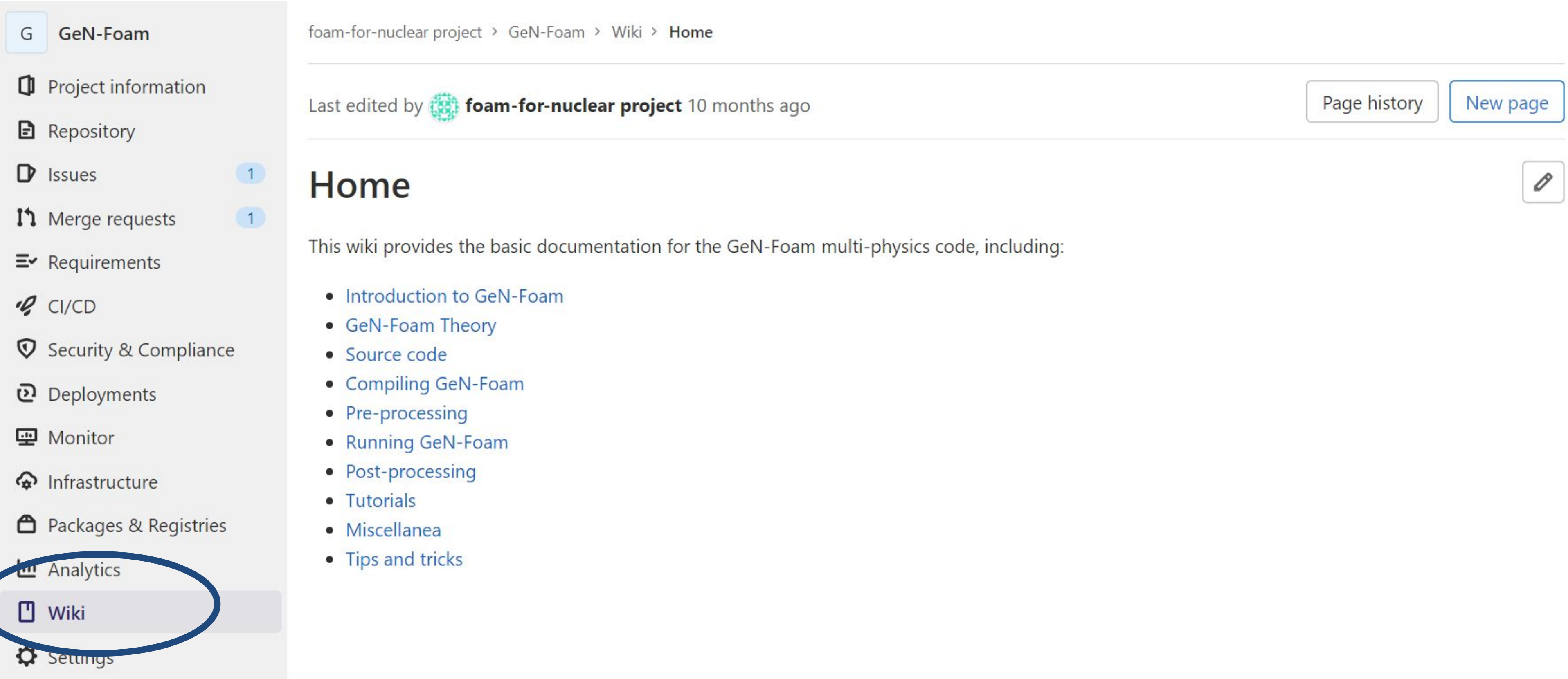
e1c958a9

Name	Last commit	Last update
..		
theory_papers	Corrected bug in pressure solution (it was n...	1 year ago
IntroToGeN-Foam_practice.pdf	Upload New File	5 months ago
IntroToGeN-Foam_theory.pdf	Upload New File	5 months ago
OpenFOAMUserGuide-A4.pdf	improved documentation	1 year ago
README	Update README in GeN-Foam/Documentati...	5 months ago

README

```
This folder contains reference papers, the standard OpenFOAM user guide, an introductory presentation about the logi
please refer to the wiki of the GitLab repository.
```

Wiki



- https://gitlab.com/foam-for-nuclear/GeN-Foam/-/wikis/home

What's inside? Source code

develop | GeN-Foam / GeN-Foam / + | Lock | History | Find file | Web IDE | Clone

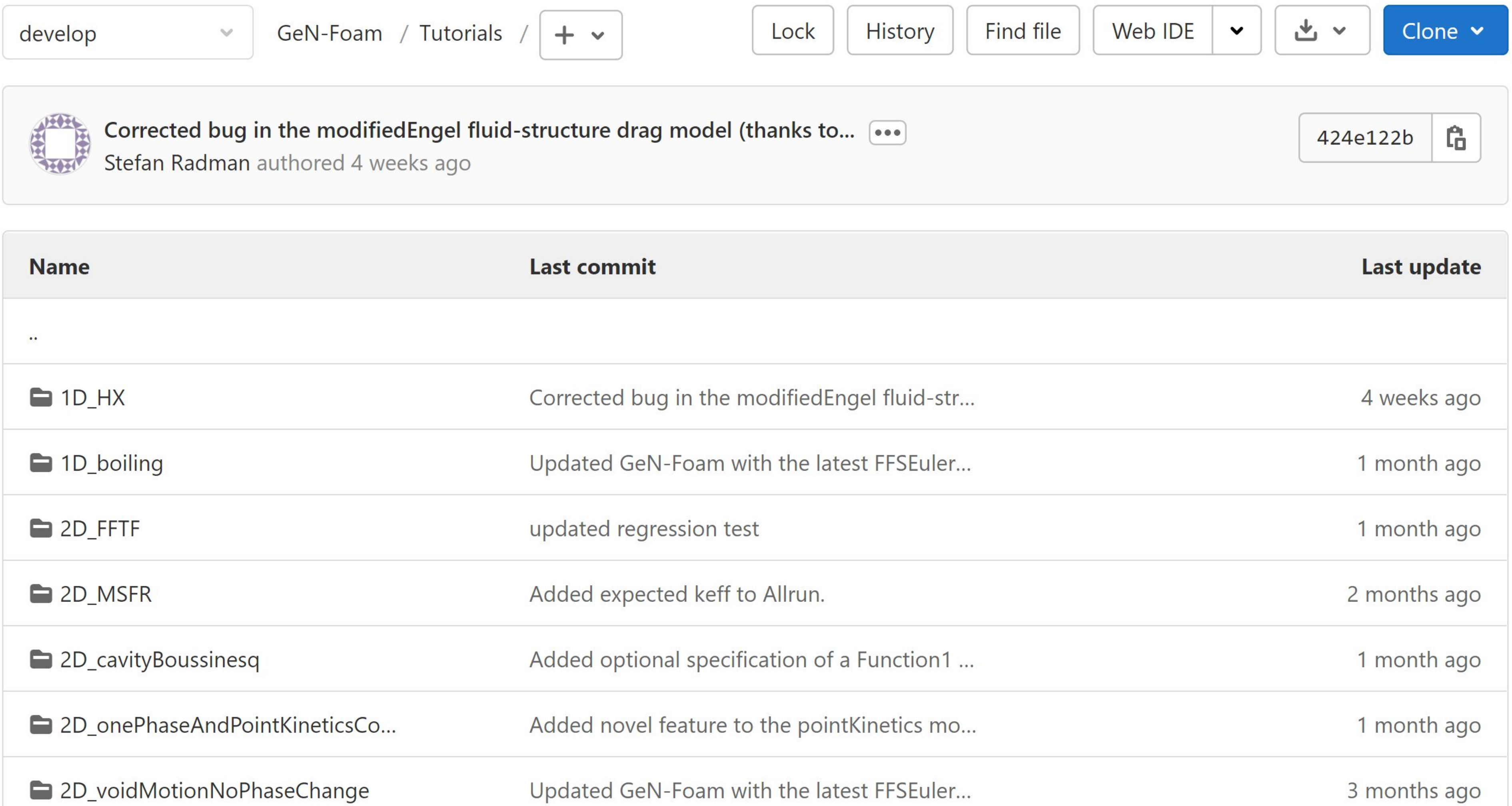
Update solvePointKineticsLiquidFuel.H
foam-for-nuclear project authored 23 hours ago — 0a05c5b4

Name	Last commit	Last update
..		
Make	Updated GeN-Faom to OpenFOAM v2006, w...	6 months ago
classes	Update solvePointKineticsLiquidFuel.H	23 hours ago
include	Updated GeN-Faom to OpenFOAM v2006, w...	6 months ago
main	Added optional specification of a Function1 ...	1 month ago
Allwclean	Added 1D tutorial case on boiling, uncouple...	9 months ago
Allwmake	Updated GeN-Foam with the latest FFSEuler...	1 month ago

- "Classes" contains all the physics
- "main" contains what glues them together
- "include" are folders that mainly contain chunks of code that perform specific tasks and that are included (#include) in the code

https://gitlab.com/foam-for-nuclear/GeN-Foam/-/tree/develop/GeN-Foam

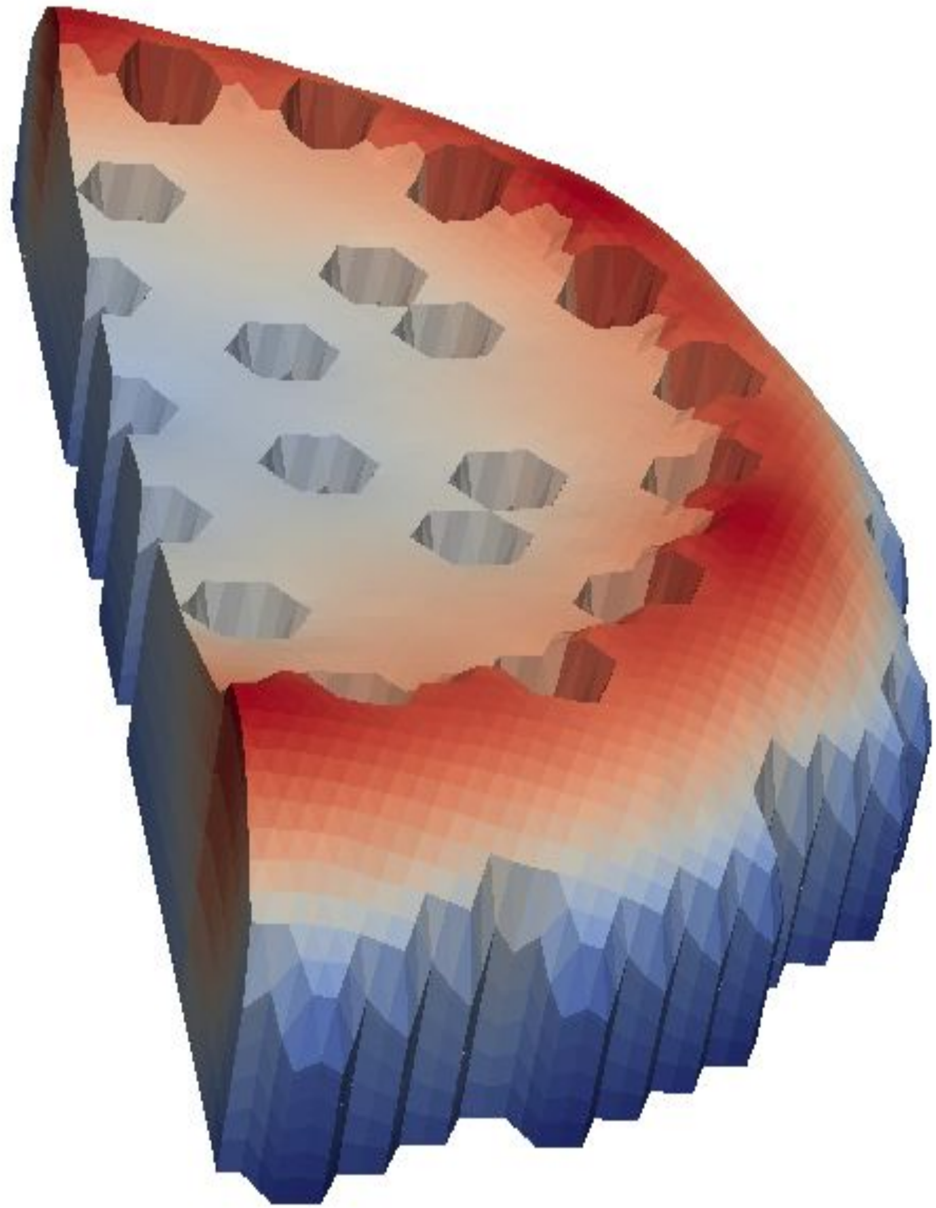
What's inside? Tutorials



- Cover essentially all functionalities of GeN-Foam
- They include a README file, an Allrun file (sometimes Allrun_parallel), an Allclean file, and some extensively commented inputs

Outline

- How to get it?
- How to install it?
- What's inside?
- **An example**
- Some other tutorials



An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

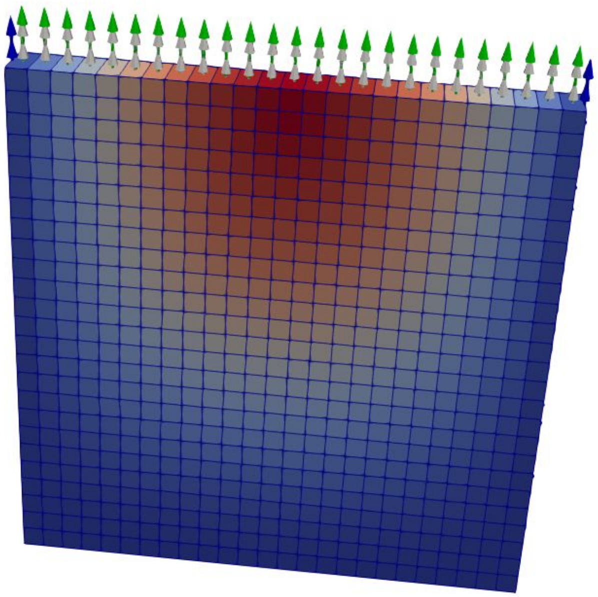
- https://gitlab.com/foam-for-nuclear/GeN-Foam/-/tree/develop/Tutorials/2D_onePhaseAndPointKineticsCoupling
- Understanding the tutorial:
  - wiki or README file
  - Case folder and Allrun file
  - Run it and use paraview to see what happens

An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

- Start from the README file (https://gitlab.com/foam-for-nuclear/GeN-Foam/-/blob/develop/Tutorials/2D_onePhaseAndPointKineticsCoupling/README)

```
DESCRIPTION
simplified test case for the pointKinetics neutronics model … 2-D
square domain … single cellZone … coolant flows from bottom to
top. Cross-sections arbitrarily chosen in 2 groups.
They are used to obtain the power
shape.
Two transients:
- 0.2$ reactivity insertion
- like the transient above, but
including driveline expansion
```

- Do an Allrun and plot some results



An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

- Look at the case folder
  - *0* folder with three subfolder containing the fields for each physics
  - *constant* folder with 3 subfolders
    - 3 meshes (*polyMesh* folders)
    - 3 sets of dictionaries
  - *system* folder with:
    - 3 subfolders with dedicated *fvScheme* and *fvSolution* for each physics
    - 1 *controlDict*
    - 1 common *fvSolution* with some multi-physics controls

```
├── 0
│   ├── fluidRegion
│   │   ├── T
│   │   ├── U
│   │   ├── ...
│   ├── neutroRegion
│   │   ├── defaultFlux
│   │   ├── ...
│   └── thermoMechanicalRegion
│       ├── CRDisp
│       ├── ...
├── constant
│   ├── fluidRegion
│   │   ├── g
│   │   ├── phaseProperties
│   │   ├── thermophysicalProperties
│   │   ├── turbulenceProperties
│   │   ├── polyMesh
│   │   │   ├── boundary
│   │   │   ├── faces
│   │   │   ├── neighbour
│   │   │   ├── owner
│   │   │   └── points
│   ├── neutroRegion
│   │   ├── CRmove
│   │   ├── neutronicsProperties
│   │   ├── nuclearData
│   │   ├── nuclearData*
│   │   ├── polyMesh
│   │   │   ├── ...
│   │   └── reactorState
│   └── thermoMechanicalRegion
│       ├── polyMesh
│       │   ├── ...
│       └── thermoMechanicalProperties
└── system
    ├── blockMeshDict
    ├── controlDict
    ├── decomposeParDict
    ├── fvSchemes
    ├── fvSolution
    ├── fluidRegion
    │   ├── decomposeParDict
    │   ├── fvSchemes
    │   └── fvSolution
    ├── neutroRegion
    │   ├── decomposeParDict
    │   ├── fvSchemes
    │   └── fvSolution
    └── thermoMechanicalRegion
        ├── fvSchemes
        └── fvSolution
```

An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

- Look at the dictionaries
  - All of the dictionaries are extensively commented in at least one of the tutorials
  - Which tutorial to look at for every dictionary? Look in the wiki/pre-preprocessing https://gitlab.com/foam-for-nuclear/GeN-Foam/-/wikis/Pre-processing
  - In our case, the tutorial is mainly dedicated to the point kinetics model. Look at constant/neutroRegion/nuclearData https://gitlab.com/foam-for-nuclear/GeN-Foam/-/blob/master/Tutorials/2D_onePhaseAndPointKineticsCoupling/rootCase/constant/neutroRegion/nuclearData

```
├── 0
│   ├── fluidRegion
│   │   ├── T
│   │   ├── U
│   │   ├── ...
│   ├── neutroRegion
│   │   ├── defaultFlux
│   │   ├── ...
│   └── thermoMechanicalRegion
│       ├── CRDisp
│       ├── ...
├── constant
│   ├── fluidRegion
│   │   ├── g
│   │   ├── phaseProperties
│   │   ├── thermophysicalProperties
│   │   ├── turbulenceProperties
│   │   ├── polyMesh
│   │   │   ├── boundary
│   │   │   ├── faces
│   │   │   ├── neighbour
│   │   │   ├── owner
│   │   │   └── points
│   ├── neutroRegion
│   │   ├── CRmove
│   │   ├── neutronicsProperties
│   │   ├── nuclearData
│   │   ├── nuclearData*
│   │   ├── polyMesh
│   │   │   ├── ...
│   │   └── reactorState
│   └── thermoMechanicalRegion
│       ├── polyMesh
│       │   ├── ...
│       └── thermoMechanicalProperties
└── system
    ├── blockMeshDict
    ├── controlDict
    ├── decomposeParDict
    ├── fvSchemes
    ├── fvSolution
    ├── fluidRegion
    │   ├── decomposeParDict
    │   ├── fvSchemes
    │   └── fvSolution
    ├── neutroRegion
    │   ├── decomposeParDict
    │   ├── fvSchemes
    │   └── fvSolution
    └── thermoMechanicalRegion
        ├── fvSchemes
        └── fvSolution
```

An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

- Look at the Allrun file

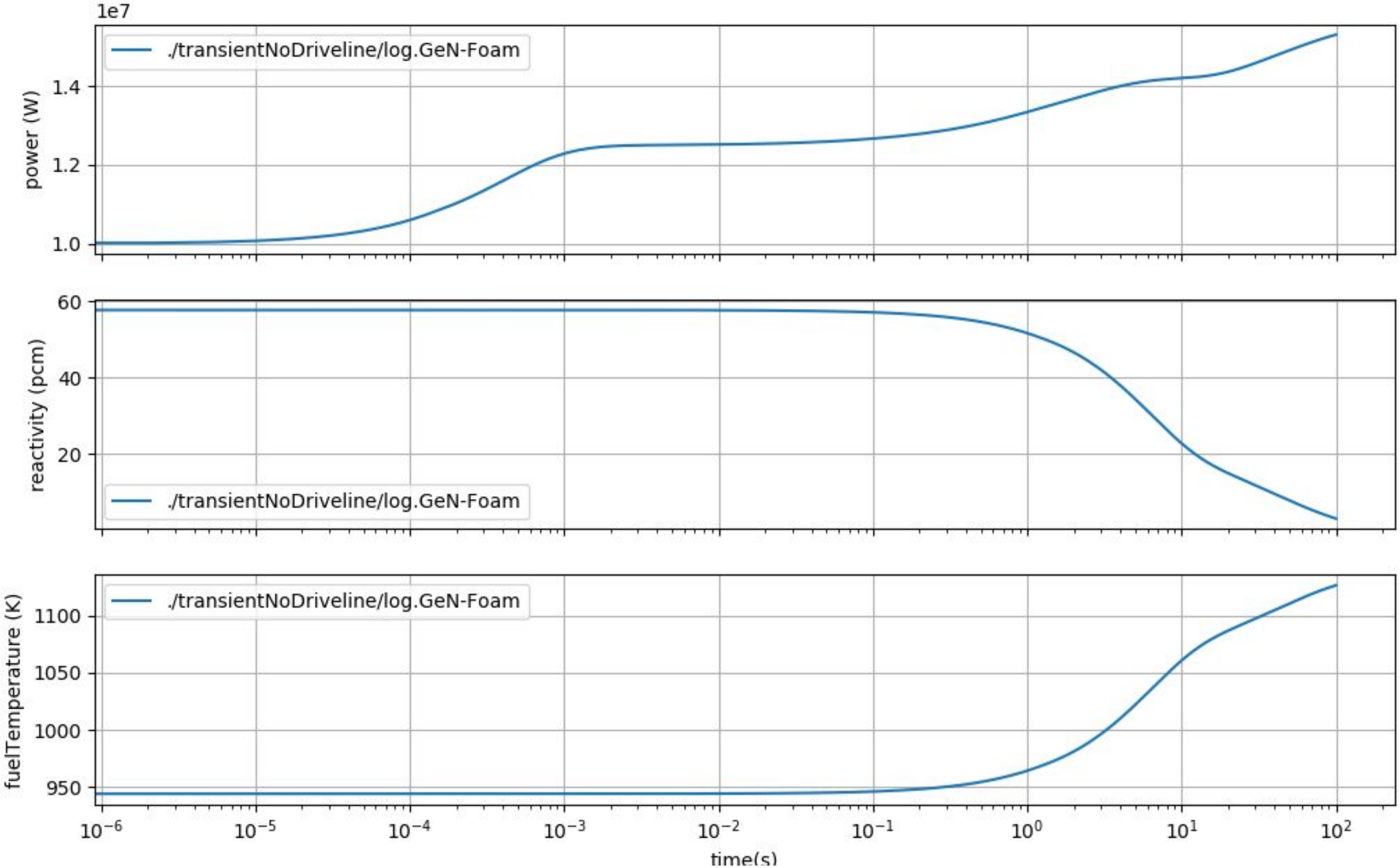
```
cases="steadyState transientNoDriveline transientWithDriveline"
...
setSteadyState()
{
    foamDictionary $2/constant/neutroRegion/neutronicsProperties -entry model -set diffusionNeutronics
    foamDictionary $2/constant/neutroRegion/neutronicsProperties -entry eigenvalueNeutronics -set true
    foamDictionary $2/system/controlDict -entry startTime -set 0
    foamDictionary $2/system/controlDict -entry endTime -set 100
    foamDictionary $2/system/controlDict -entry deltaT -set 1
...
setTransientNoDriveline()
{
    foamDictionary $2/constant/neutroRegion/neutronicsProperties -entry model -set pointKinetics
    foamDictionary $2/constant/neutroRegion/neutronicsProperties -entry eigenvalueNeutronics -set
false
    foamDictionary $2/system/controlDict -entry startTime -set 100
    foamDictionary $2/system/controlDict -entry endTime -set 200
    foamDictionary $2/system/controlDict -entry deltaT -set 1e-6
    foamDictionary $2/constant/neutroRegion/nuclearData -entry absoluteDrivelineExpansionCoeff -set
0.0
...
```

An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

- Run the tutorial -> ./Allrun
- Check the results:
  - Choose a folder: steadyState, transientNoDriveline, transientWithDriveline
  - Use:
    - ./log.GeN-Foam: standard OpenFOAM log
    - ./GeN-Foam.dat: quick overview of time behavior of main quantities (power, keff, min/max/average fuel and clad temp.)
    - ./constant/neutroRegion/reactorState for keff
    - in some tutorials, a python script to extract info from log file
    - **paraFoam -region regionName (where regionName is set to fluidRegion, neutroRegion, or thermoMechanicalRegion)**

An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

- python script (extract data from log)

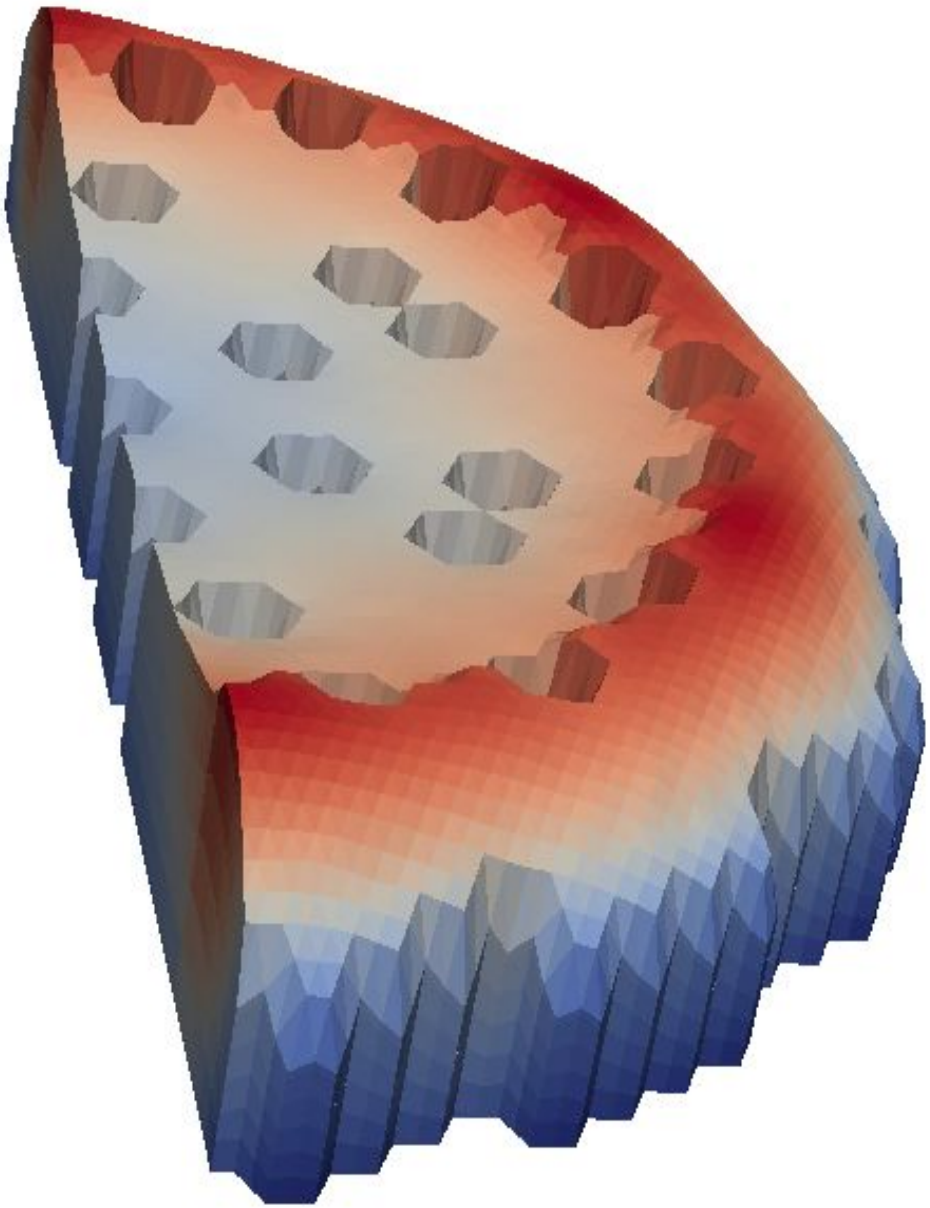


An example: the tutorial 2D_onePhaseAndPointKineticsCoupling

- paraview (export animation + ffmpeg)

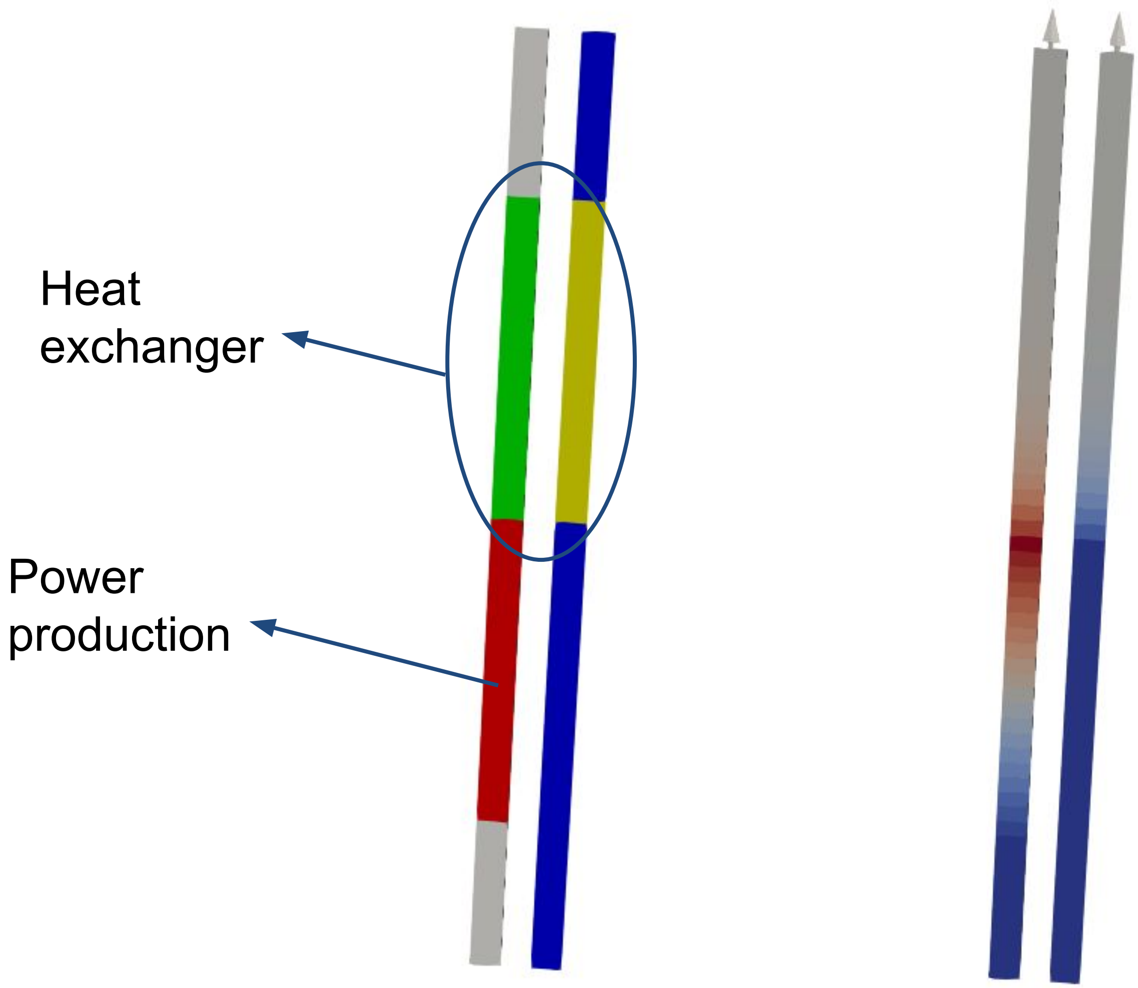
Outline

- How to get it?
- How to install it?
- What's inside?
- An example
- **Some other tutorials**



1D_HX

- Example on how to set up a heat exchanger

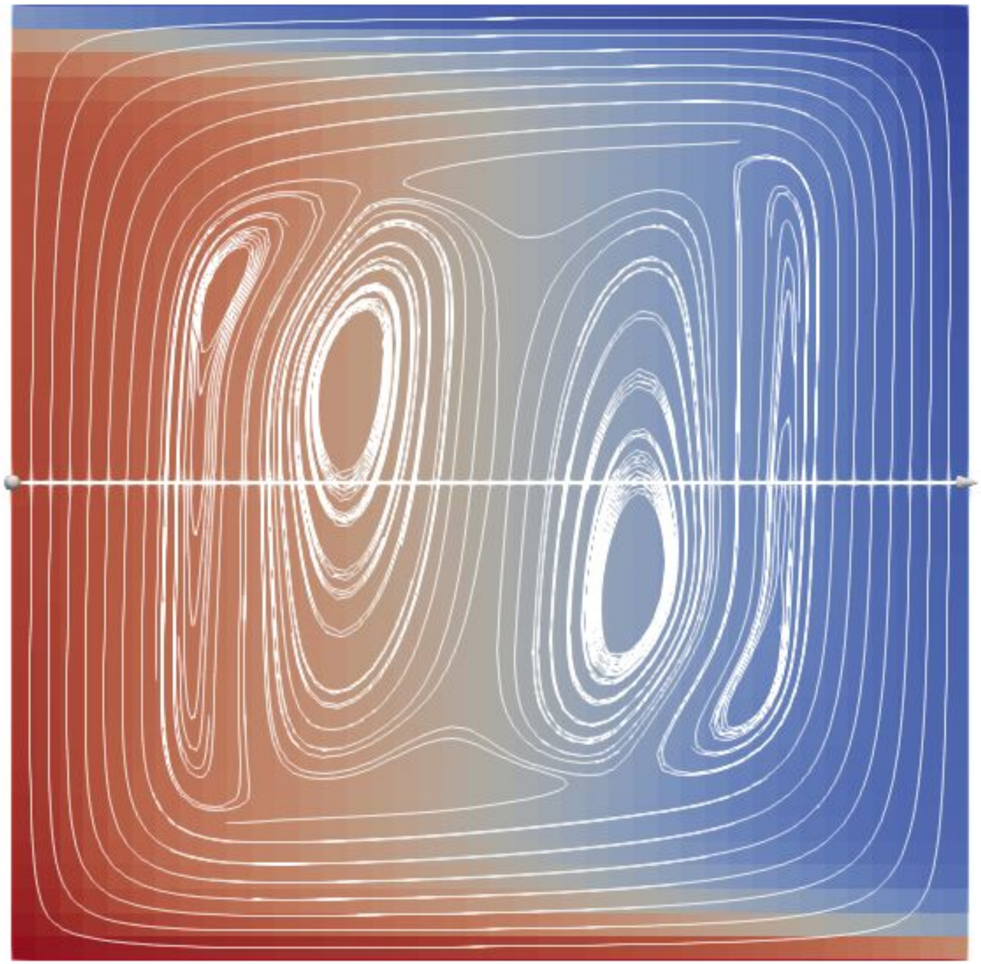


1D_boiling

- Example of two-phase simulation. 1D channel with a pressure-driven flow of liquid sodium, with power source turned on at time 0, eventually leading to boiling. After a certain time the power is turned off

2D_cavityBoussinesq

- Example of how to use of the Boussinesq approximation for buoyancy based on the standard buoyancy-driven cavity

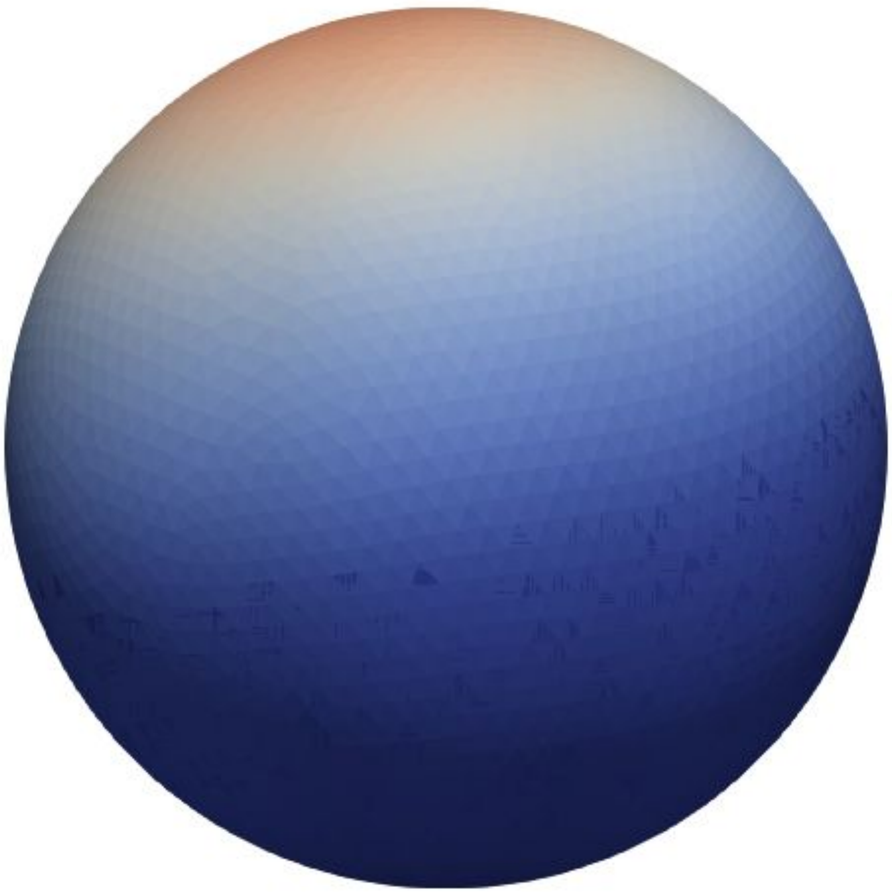


2D_voidMotionNoPhaseChange

- Simple two-phase case without mass transfer between phases

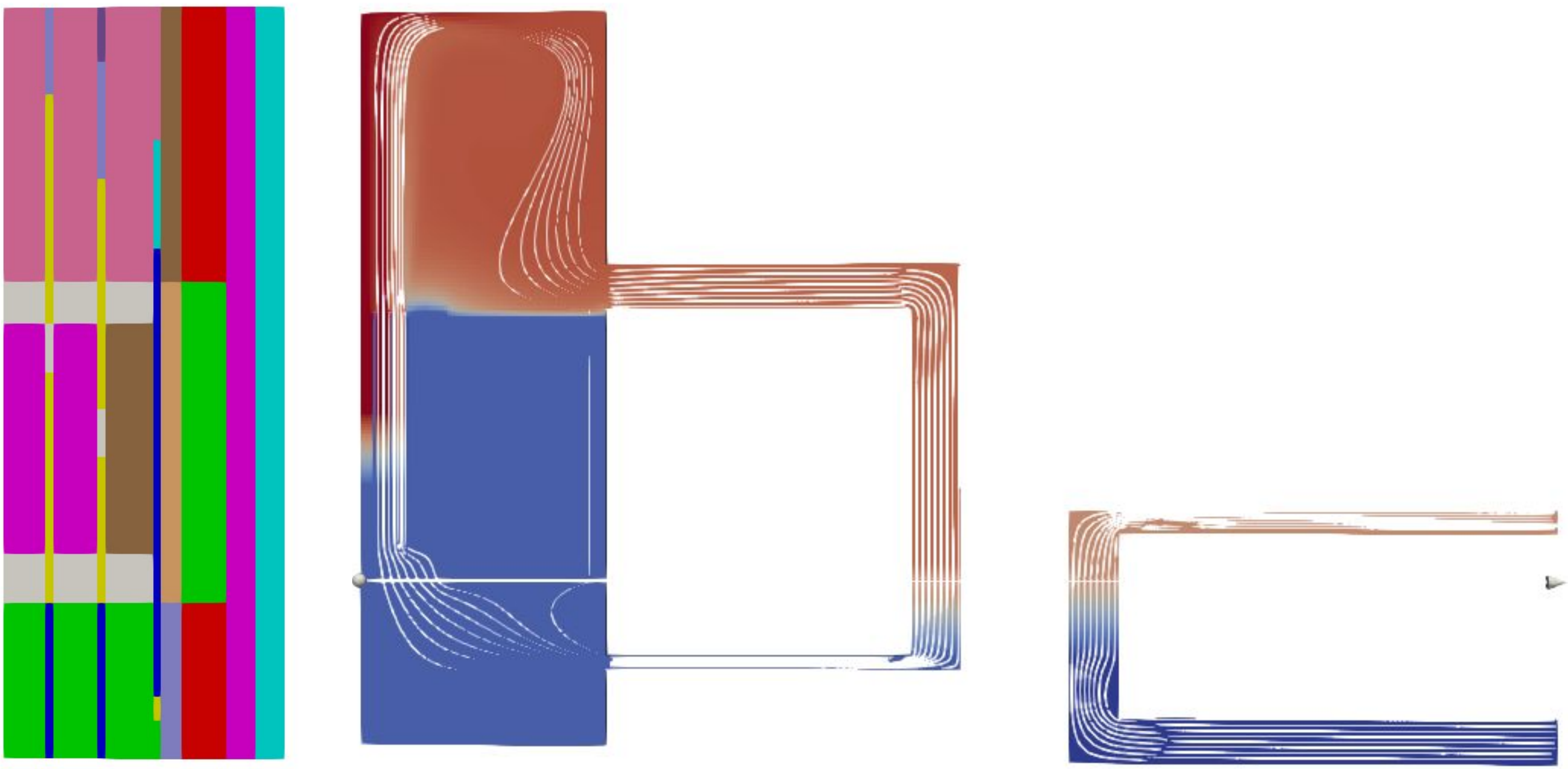
Godiva_SN

- Example of a discrete ordinate calculation of Godiva



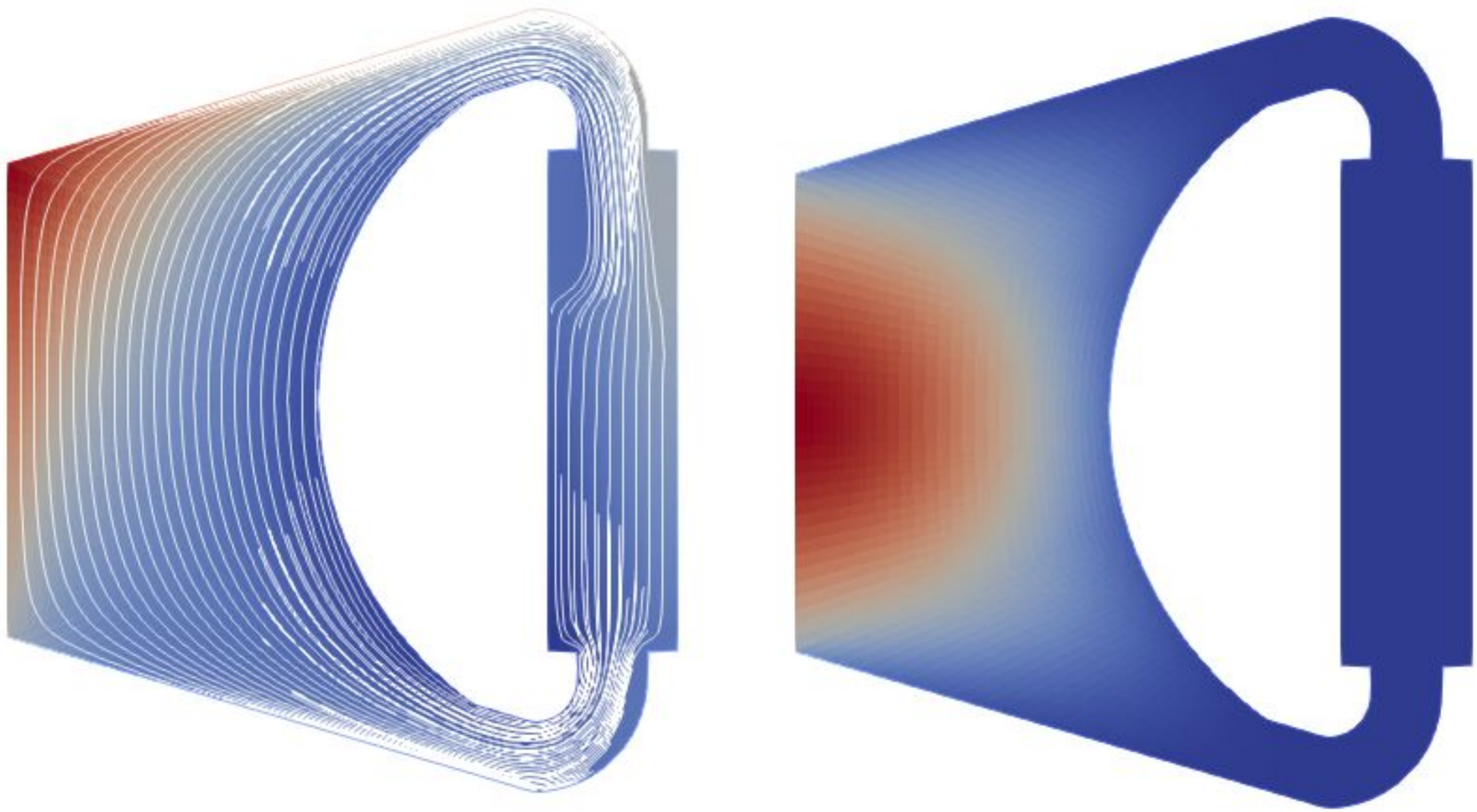
2D_FFTF

- Simplified, 2-D model of the FFTF. Simulation of a ULOF



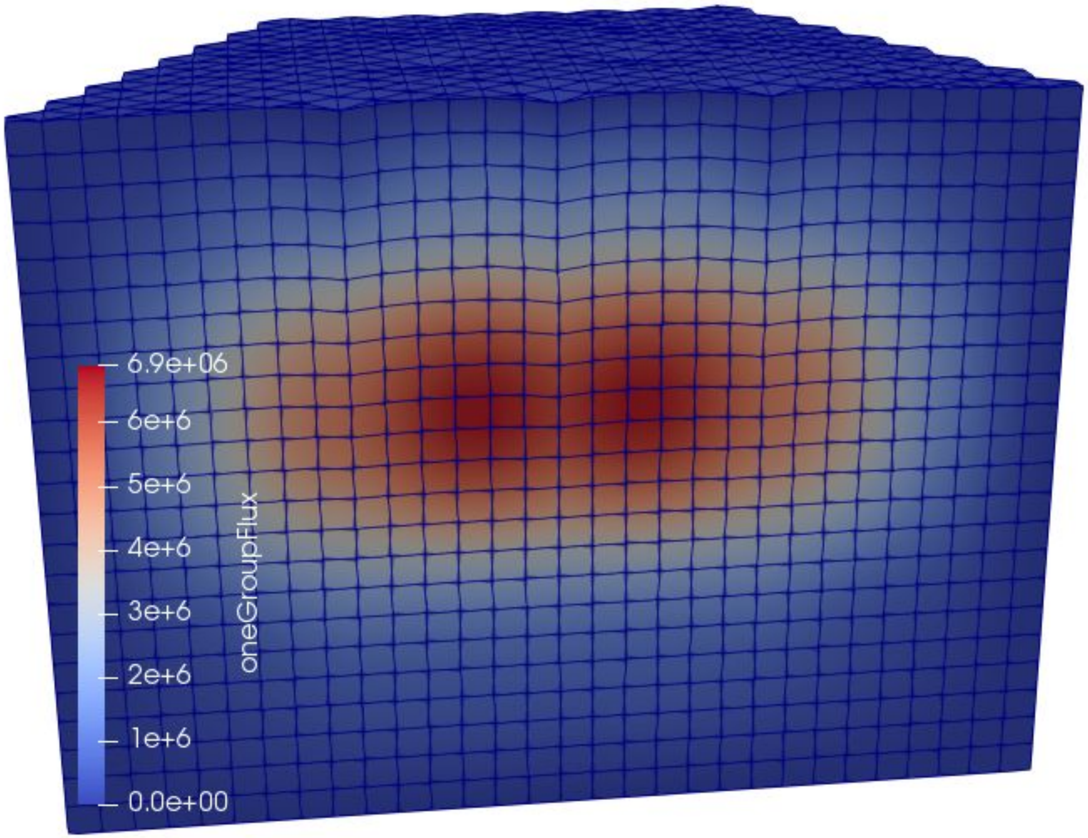
2D_MSFR

- Simplified, 2-D model of the MSFR



3D_SmallESFR

- Slightly smaller version of the European Sodium Fast Reactor
- Example of a 3D full multi-physics simulation, including core deformation



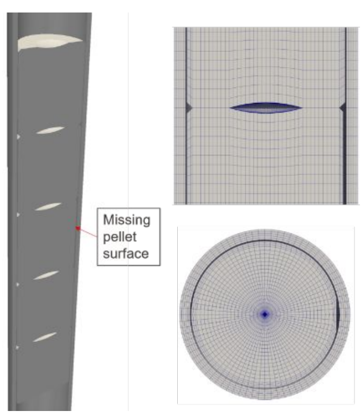
Thank you for your attention

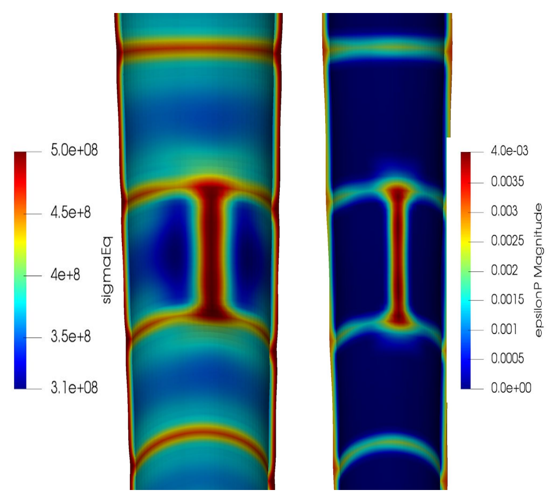
Carlo Fiorina carlo.fiorina@epfl.ch

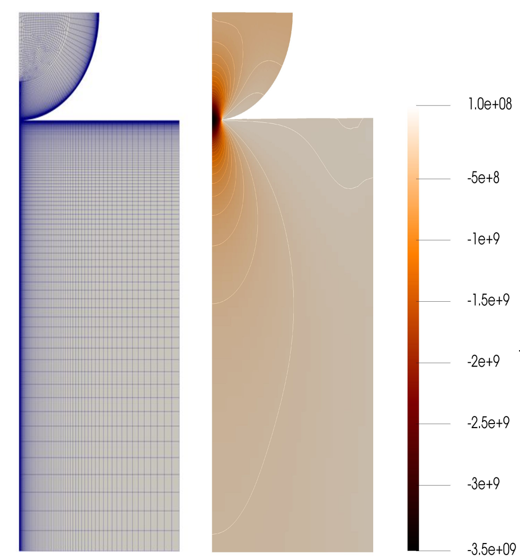
C. Fiorina

Stay tuned: more from the EPFL

- OFFBEAT: OpenFOAM Fuel BEhavior Analysis Tool
  - Advanced multi-dimensional tool co-developed by the EPFL and the PSI
  - Currently under development
  - First release planned for 2022







Stay tuned

- IAEA Technical Meeting on the Development and Application of Open-Source Modelling and Simulation Tools for Nuclear Reactors (27-29 October 2021)
  - https://conferences.iaea.org/event/247/
  - Promote and facilitate the exchange of information
  - Present and discuss the current status of research and development
  - Discuss the available open-source tools and their state of development;
  - Discuss the value and drawbacks of the open-source model
  - Provide forum for sharing user and developer experiences
  - Discuss open-source code project best practices,
  - Discuss the opportunities provided by the use of open-source codes for education and training
  - Discuss and identify R&D needs and gaps

Stay tuned

- PHYSOR 2022
    - https://www.ans.org/meetings/physor2022/
    - May 15–20, 2022
    - Pittsburgh, PA
    - IAEA special session of use and development of open-source code
    - Full day workshop on the use OpenFOAM for nuclear reactor analysis