Italian National Agency for New Technologies, Energy and Sustainable Economic Development

USE OF OPEN-SOURCE SOFTWARE AT ENEA FOR NUCLEAR REACTOR SAFETY SIMULATIONS

Milan — June 23rd, 2022

Antonio Cervone[1], R. Da Vià[1], S. Manservisi[2], P. Meloni[1], M. Polidori[1]
1: ENEA / FSN / SICNUC, 2: University of Bologna – DIN

Outline

Outline

A bit of history

CFD for Nuclear Reactor Safety

Code coupling

Atmospheric dispersion

Support tools

Training and Education

Challenges and future perspectives

Thermal-hydraulics for Nuclear Reactor Safety

ENEA is a public research organization in Italy devoted to research in the broad field of energy.
There is a division for Nuclear Safety, Security and Sustainability.

- two-phase flow system thermal-hydraulics is one of the core businesses at our division
- closed-source system codes such as RELAP-5/TRACE (from US-NRC) and CATHARE 2/3 (from CEA) have been is use at ENEA for a very long time
- ENEA relies on specific licensing since there is not enough man-power to build an independent capability for these type of codes
  - extensive database of validation data is required
  - experimental facilities are expensive and hard to operate
  - data from other institutions has to be licensed and its use can be limited

My history before ENEA

- PhD.: development on FeMUS (UniBO)
  `https://github.com/FemusPlatform`
  - parallel finite element Navier-Stokes solver based on PETSc
  - multi-physics
  - focus on two-phase, FSI and liquid metals for innovative nuclear reactors
- PostDoc: development on LifeV (PoliMI, EPFL, Emory, INRIA)
  `https://bitbucket.org/lifev-dev/lifev-release/wiki/Home`
  - parallel finite element PDE solver based on Trilinos
  - multi-physics
  - focus on sedimentary basins, heart, circulatory system

Outline

CFD for Nuclear Reactor Safety

My task at ENEA has been the introduction of CFD into the mix in order to

- improve fidelity of the simulations for specific components
- develop stand-alone models of facilities or components that are inherently difficult or cumbersome to simulate with system codes that mainly work with 1D and 0D components
- couple CFD and system codes to perform accurate simulations of complex systems where both set of capabilities are required
- develop new approaches and introduce innovative frameworks and model systems to extend our capabilities towards a larger set of applications
- deploy codes and application on HPC infrastructure

CFD for Nuclear Reactor Safety

Why open-source for CFD applications?

Ups:

- CFD is general purpose with many open-source frameworks available
- easier to plug in dedicated models
- no licensing problems
- more suited for HPC deployment
- easier to couple to other (closed or open) codes

Downs:

- scarse documentation and little support
- necessity to delve into low-level stuff

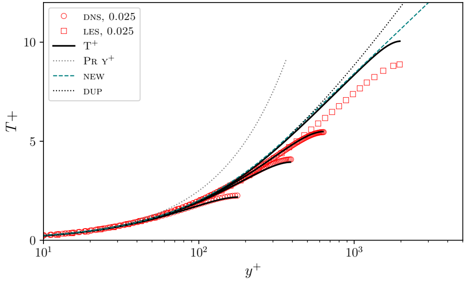
Turbulence in liquid metals

Liquid metals present unique challenges in the CFD simulation due to their very low Prandtl number: reference implementation for air and water are not suitable, dedicated models are required.

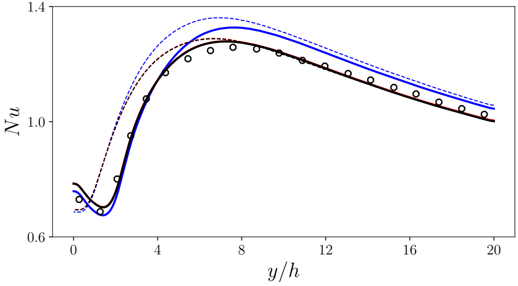
- Implementation of new models for thermal turbulence:
  - Algebraic models:
    - Kays correlation for turbulent Prandtl number
      W. Kays, Journal of Heat Transfer, vol. 116, no. 2, pp. 284–295, 1994
  - One equation models:
    - Algebraic heat flux: one transport equation for k_θ
      A. Shams et al. International Journal of Heat and Mass Transfer, vol. 79, pp. 589–601, 2014
  - Two equations models:
    - Four parameters turbulence model: one transport equation for k_θ and one for ε_θ
      R. Da Vià and S. Manservisi, International Journal of Heat and Mass Transfer, vol. 135, pp. 591–603, 2019

Liquid metals turbulence models

Validation against DNS and LES data



fully developed flow in channel

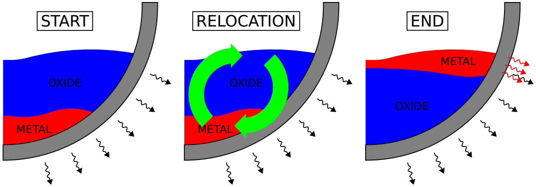


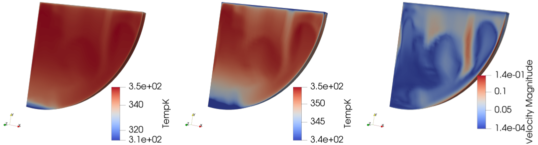
backward facing step

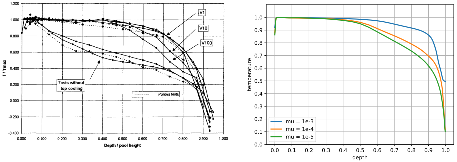
Initially developed and implemented in FEMuS, recently it has been ported to OpenFOAM as well (soon to be publicly released).

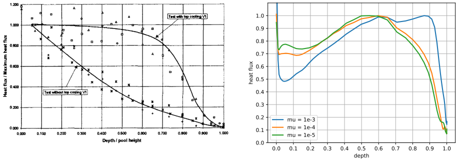
Liquid metals: BALI experimental facility

- simulation developed using `openFOAM`
- validation of the temperature and heat flux profiles
- part of a an application for the simulation of the corium behavior during a severe accident





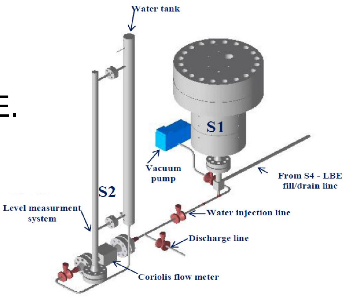


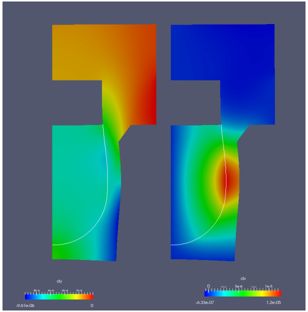


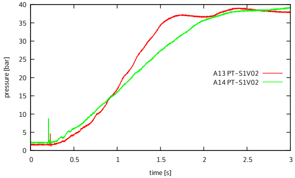
Fluid-Structure Interaction: LIFUS-5 facility

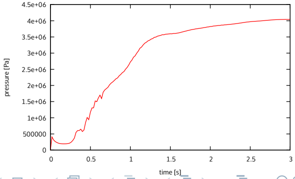
Simulation of water injection in LBE.

- Arbitrary Lagrangian- Eulerian (ALE) single fluid formulation with different constitutive laws
  - fluid: incompressible
  - solid: linear elasticity
- based of `libMesh`
- comparable results to LS-DYNA

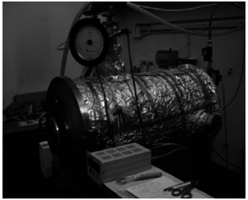


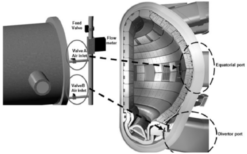






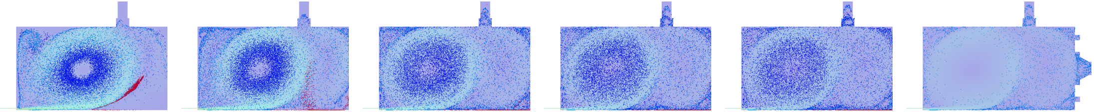
CFD with Lagrangian Particles: the STARDUST facility





Simulation of dust mobilization during a LOVA (Loss of Vacuum) incident in ITER.

- based on `code_Saturne` from EDF
- compressible Navier-Stokes with Lagrangian particles
- one-way coupling



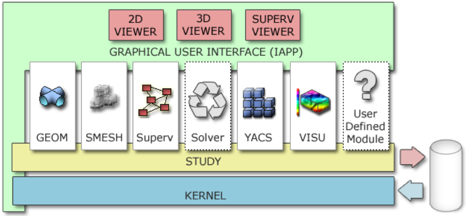
Outline

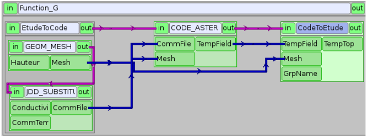
SALOME

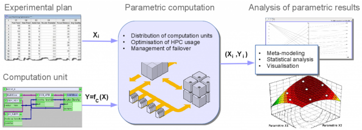
SALOME (`www.salome-platform.org`) is a open-source platoform developed by CEA and EDF that tries to manage all aspects of a simulation: CAD, meshing, coupling of phenomena, visualisation, calculation supervision, uncertainties, etc.

We adopted it as it came with capabilities to interface with all the French nuclear software ecosystem.

Based on the MED library and the HDF5 standard for intercommunication between codes.





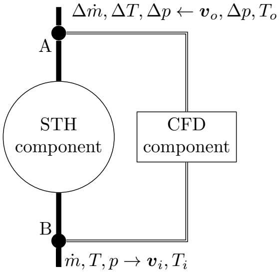


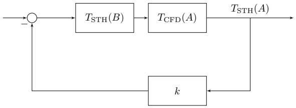
ICoCo

The control that is achivable in SALOME/YACS was not sufficient to build a customized strong (in-memory) coupling

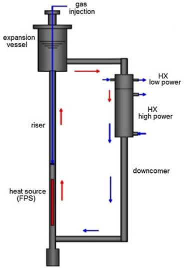
- ICoCo is a C++ interface for coupling specific codes
- embedded in CATHARE in TrioU/TRUST
- lower level than SALOME/YACS
- additional interfaces for FEMuS, OpenFOAM, Dragon/Donjon

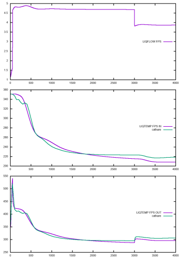
Via ICOCO we managed to operate the codes in an feedback controlled coupling with overlapping domains.

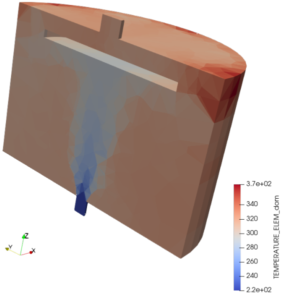


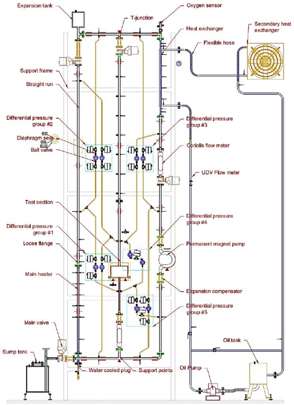


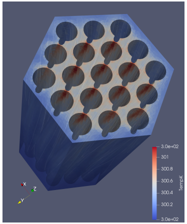
Coupled simulation of liquid metals: NACIE_UP and TALL3D

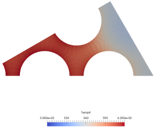


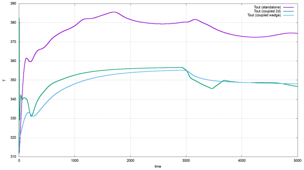












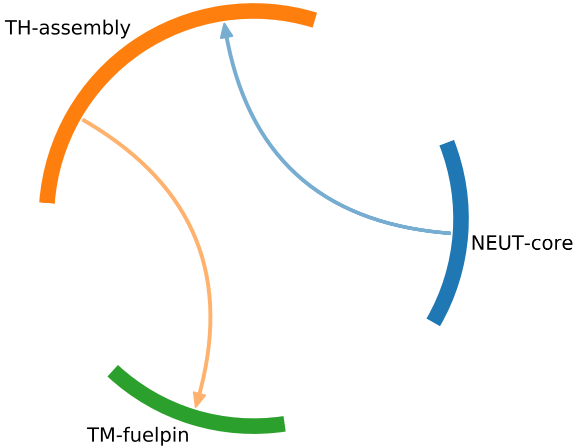
MEDUSA

- MEDUSA is a platform designed to facilitate the simulation of multi-physics and multi-scale problems through code coupling
- MED and MEDCoupling library (from SALOME) as common format for storing solutions
  - Coupling benefits from a great availability of interpolation methods
  - Visualization using ParaVIS (PARAVIEW in SALOME)

		Physical phenomena		
		Neutronics	Thermo-Mechanics	Thermo-Hydraulic
Physical scale	Pin		X	
	Assembly	(X)	X	X
	Core, Shielding	X	X	X

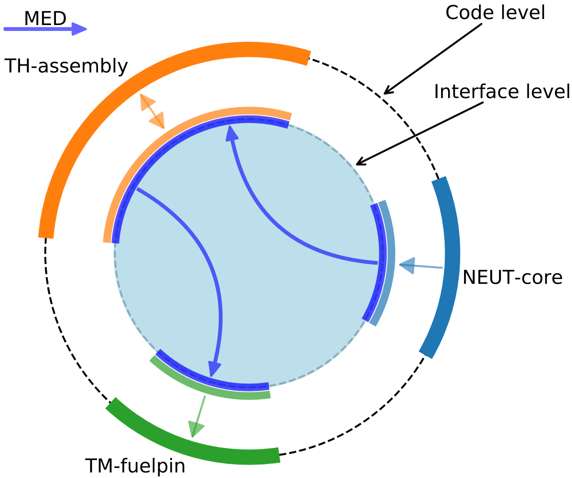
MEDUSA: a coupling chain case

- Neutronic code: 3D power distribution in core
  - Need to interpolate on TH code mesh
- Thermo-Hydraulic code: temperature distribution in assembly
  - Need to interpolate on TM code mesh
- Thermo-Mechanic code: temperature distribution in fuel pin

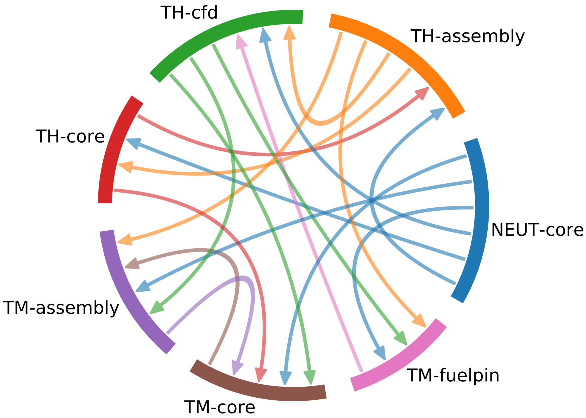


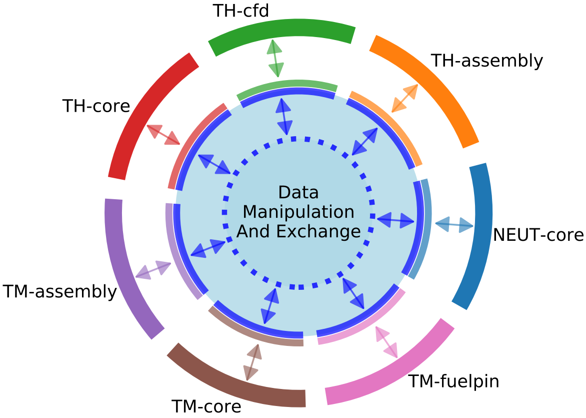
MEDUSA: using a common solution format

- One interface application must be provided for each code
- At interface level, solution is converted from/to code format to/from common (MED) format
- Interpolation (data manipulation in general) is handled with set of functions written for MED format (more general, re-usable)



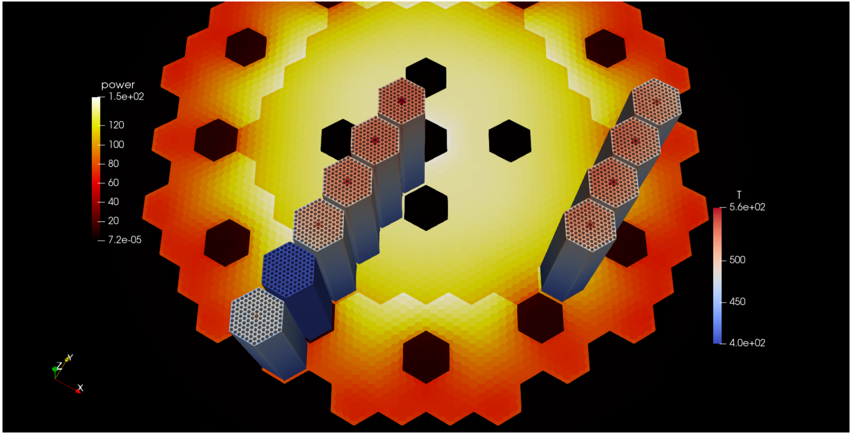
MEDUSA: allow the platform to scale





- With a common format for solutions, coupling can be easily handled and the platform can scale (add new codes and possible coupling chains)

MEDUSA: example of neutronics and TH coupling



Outline

Atmospheric dispersion for NRS

FLEXPART (`www.flexpart.eu`) is an open-source software for meso-scale dispersion of (radio-) pollutants in the atmosphere.

- heavily relies on the availability of meteorological data (hard to get, up until recently)
- poor documentation and open-source governance
- improving!
- spurred the developement of a support tool `flextools` to manage input/output, visualization, interfacing with ECMWF MetView

Outline

General purpose open-source tools

A lot of general purpose open-source tools that are not specific to the nuclear field are indeed necessary to the realization of a CFD simulation

- `spack` for dependency management/installation
- GMSH and pyGMSH for mesh generation
- ParaView for postprocessing
- etc.

Many of these tools have their own unique way of deploying open-source software.

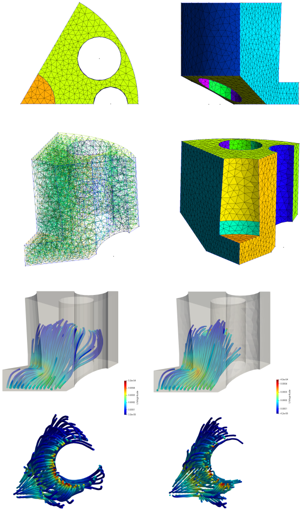
Outline

Training and Education

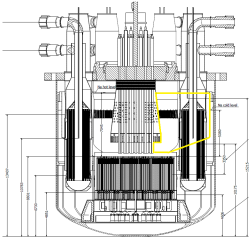
Open-source enables easier and powerful collaboration with universities

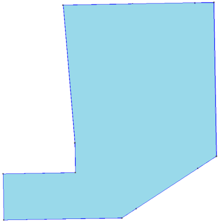
- undergraduate/master students from university come for their dissertation
- learning open-source software have several advantages
  - engaging – become part of a community
  - sharable – easier to deliver publications and open or closed applications as well
  - limited scope – easier to find small self-contained tasks for green developers
  - portable – insurance that the skill can be used in their future

Liquid metals: ESFR-SMART Redan meshing



Development of a mesh to simulate the REDAN component of the ESFR-SMART reactor





Outline

Challenges

For Nuclear Safety applications two domains are still poorly covered in open-source

- system level codes with robust validation
- two-phase flow with two-fluid model capable of simulating nuclear reactor regimes of interest (high pressure, boiling, DNB, etc.)

Another aspect is the modernization and maintainence of legacy codes.

Future perspectives

The EU is pushing harder and harder on opening the research that is financed with public money.

FAIR Findable Accessible Interoperable Reusable - applies to data and software

Open Science as the combination of Open Publication, Open Data, Open Code, that makes the product *reproducible*

Some aspects are already mandatory in the newest line of EURATOM and Horizon EU calls.

Open-source can play a key role in this transition and leverage it to move new and old projects towards the open-source paradigm.

Thanks for the attention!
Questions?

Antonio Cervone

antonio.cervone@enea.it