Salome_CFD user meeting 2019

Salome_CFD Highlights

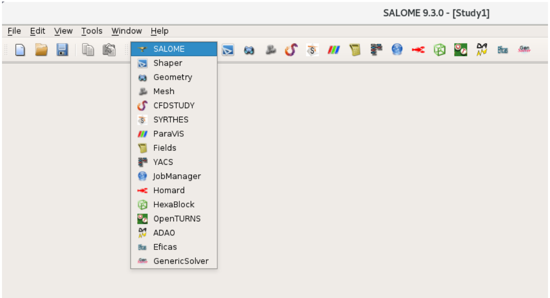
Salome_CFD development team[1]

7th May 2019

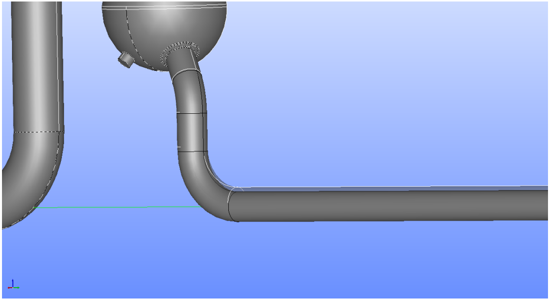
[1]Fluid Mechanics, Energy and Environment Department
EDF R&D, Chatou, France
saturne-support@edf.fr

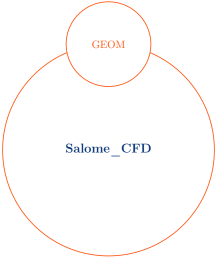
Outlines

Salome_CFD in short

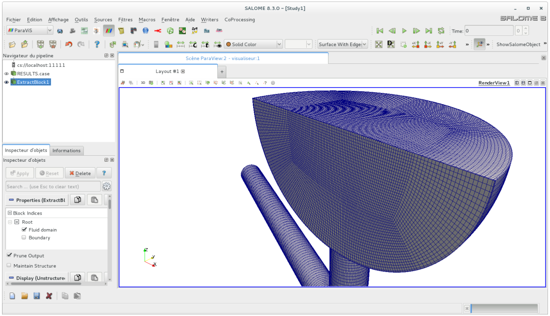


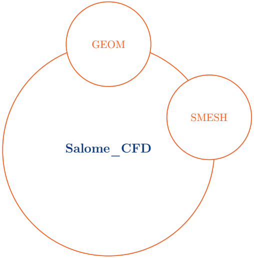
Salome_CFD in short





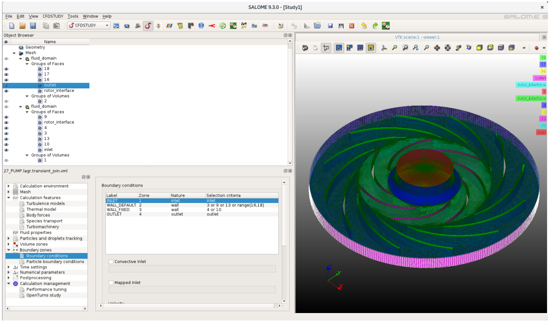
Salome_CFD in short

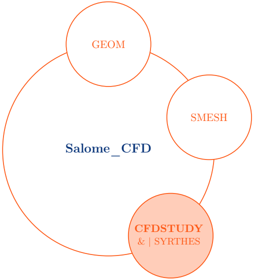




Advanced scripting capabilities

Salome_CFD in short

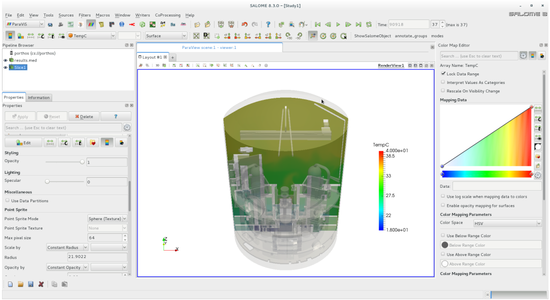




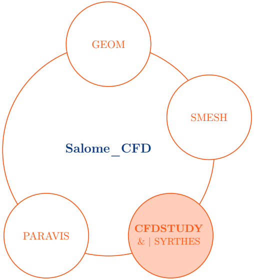
Single-phase solver *Code_Saturne*

Multi-phase solver NEPTUNE_CFD

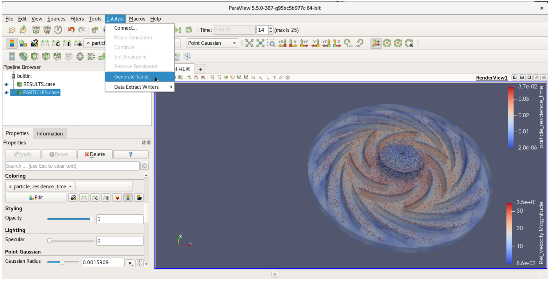
Salome_CFD in short

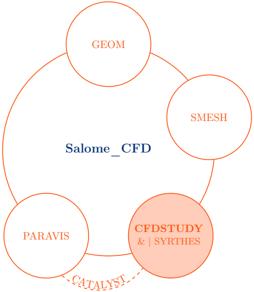


Visualisation / Remote visualisation for Big Data



Salome_CFD in short

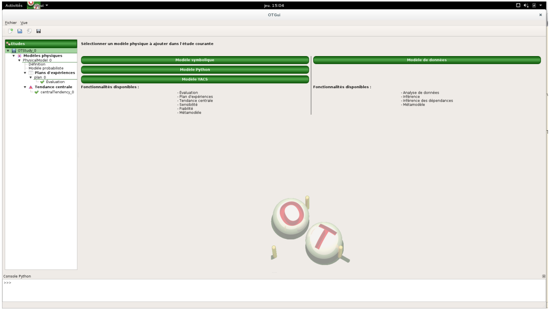


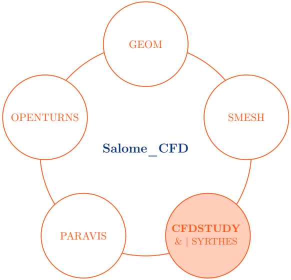


Visualisation / Remote visualisation for Big Data

In-situ and live visualization

Salome_CFD in short





UQ studies

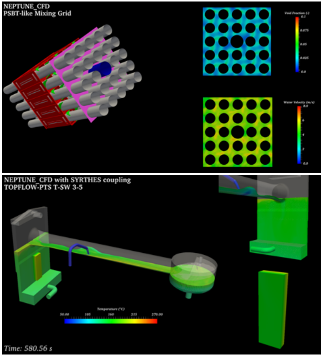
Design

Multiphase solver developed in the NEPTUNE project (EDF, Framatome, CEA, IRSN): NEPTUNE_CFD

- → Multi-fluid, Eulerian model (one-pressure, "Ishii" class of models)
- → Dedicated sets of physical models to tackle **a wide range of two-phase flows with different topologies (stratified, dispersed, etc.)**
- → Inherits **many capabilities from *Code_Saturne***: HPC, pre/post-processing, GUI, linear algebra, data structures..
- → Not open-source

Some target nuclear applications:

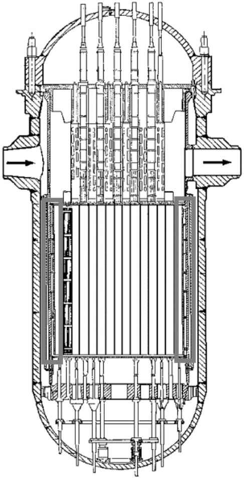
- → Departure from Nuclear Boiling (DNB)
- → Two-phase Pressurized Thermal Shock (PTS)
- → Containment flows (with aspersion and condensation)
- → Gas transport in pipes
- → Downcomer during a LOCA
- → Spent-fuel pool in case of accident
- → In-vessel corium retention

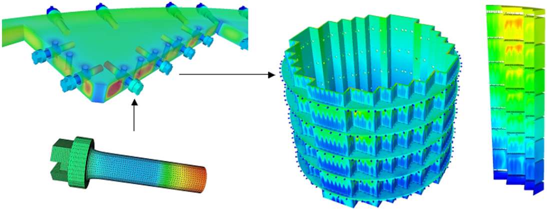


Thermal diffusion in solids and radiative transfer solver: SYRTHES

Thermal load of 1000 bolts in 900 PWR internals

1h30 on 2048 BG cores - 1 billion tet mesh

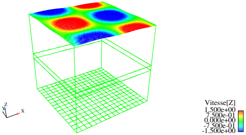




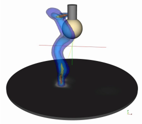
Solid coupled to fluid (*Code_Saturne*)

Multiphysics solvers gathered in *Code_Saturne*

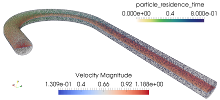
Arbitrary Lagrangian Eulerian



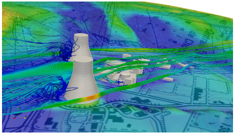
Electric Arcs



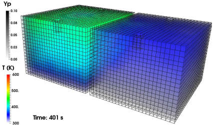
Lagrangian particle tracking



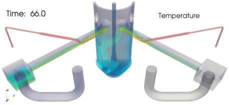
Atmospheric flows



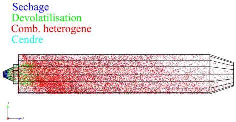
Fire modelling



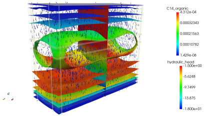
Thermohydraulics for nuclear



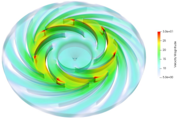
Combustion (coal, fuel, gas)



Groundwater flows

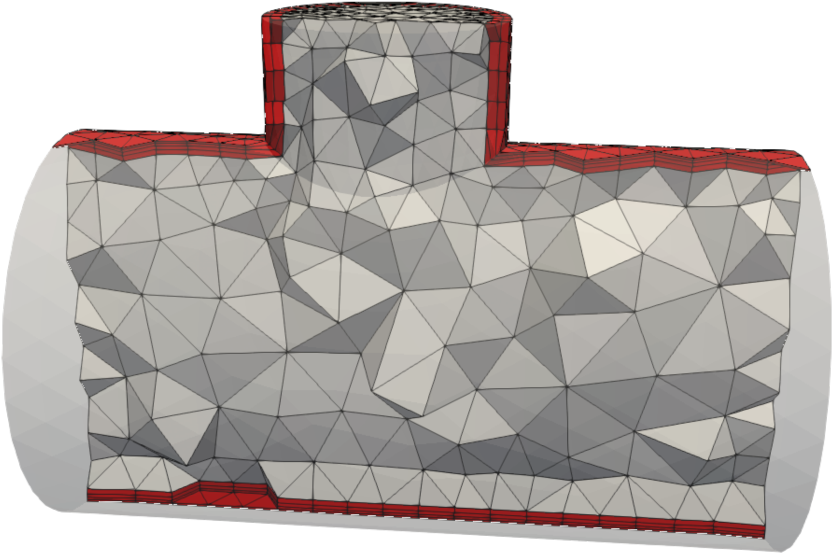


Turbomachinery



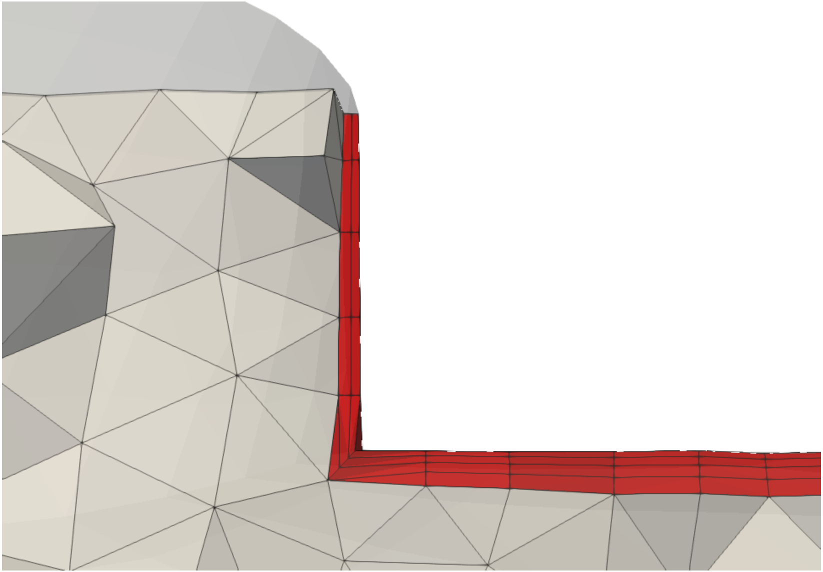
Preprocessing: automatic insertion of wall-layer cells

using CDO vertex based ALE solver



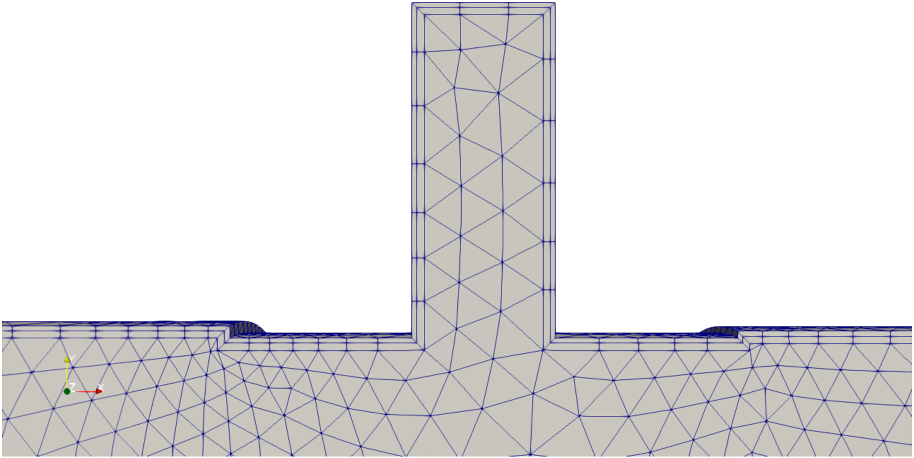
Preprocessing: automatic insertion of wall-layer cells

using CDO vertex based ALE solver



Preprocessing: automatic insertion of wall-layer cells

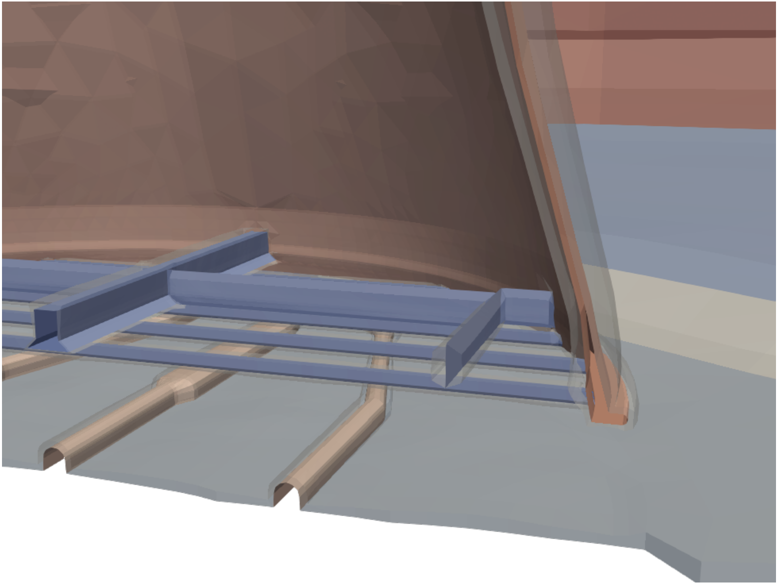
using CDO vertex based ALE solver



see user source file `cs_user_mesh-modify.c`.

Preprocessing: automatic insertion of wall-layer cells

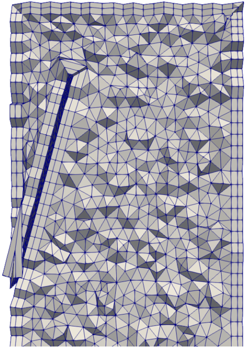
using CDO vertex based ALE solver



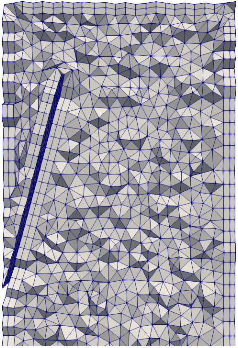
Preprocessing: automatic insertion of wall-layer cells

Fix features in sharp angles

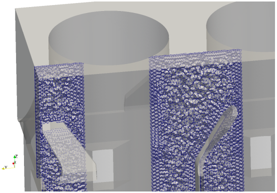
A solution is to test for negative volumes while deforming the mesh, and locally limit the extrusion on adjacent boundaries (removing one extrusion layer at vertices of those cells). This is done iteratively until no negative volume cells are produced.



before



after

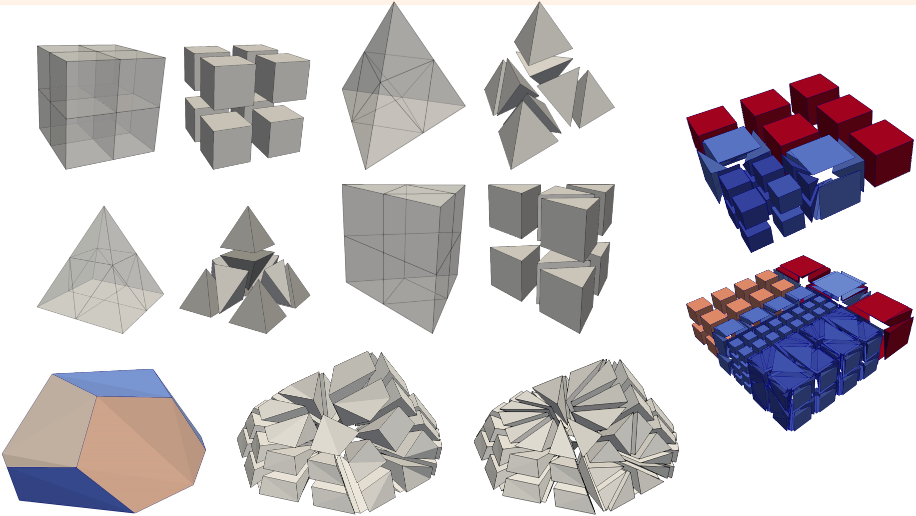


CALIFS

Also add optional cell volume ratio limiter to reduce the extrusion near cells that would be excessively flattened or entangled.

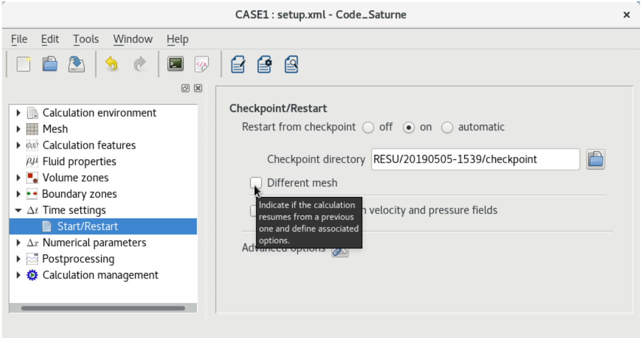
Preprocessing: Add mesh refinement engine

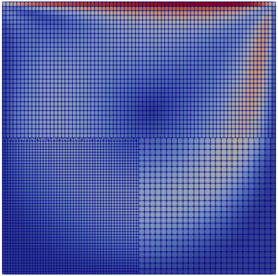
for any polyhedral, load balancing currently handled through complete repartitioning, in collaboration with STFC



see function `cs_user_mesh-modify.c`.

Add restart function to allow mapping restarts from computation using a different mesh

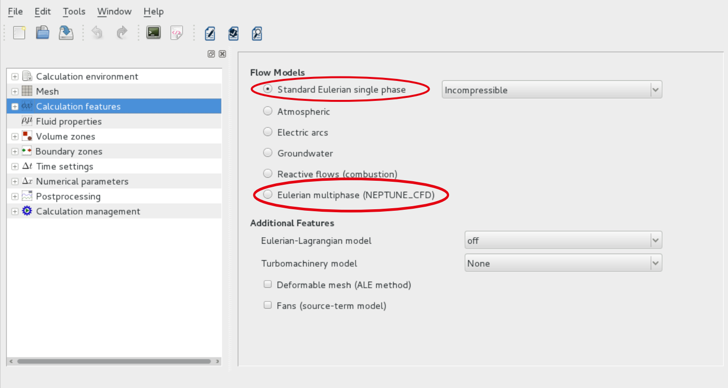




Code_Saturne & NEPTUNE_CFD now in the same GUI

Switch between *Code_Saturne* and NEPTUNE_CFD solvers

It is now possible to easily switch between the *Code_Saturne* solver (Standard Eulerian single-phase) and NEPTUNE_CFD solver (Eulerian Multiphase) using the GUI:

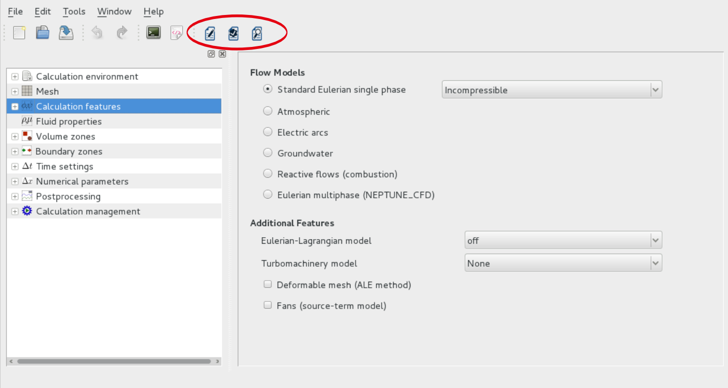


Overhaul of the Graphical User Interface (GUI)

Improving the global workflow of a study

→ Manage User functions within the GUI!

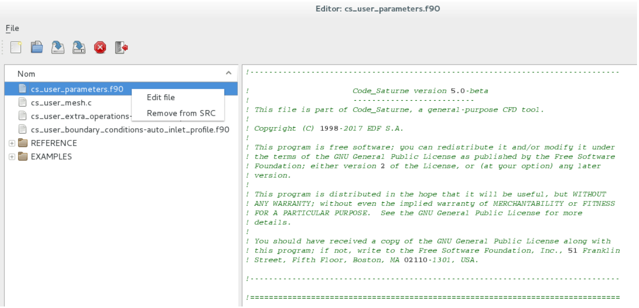
→ View log files of a run



Overhaul of the Graphical User Interface (GUI)

Improving the global workflow of a study

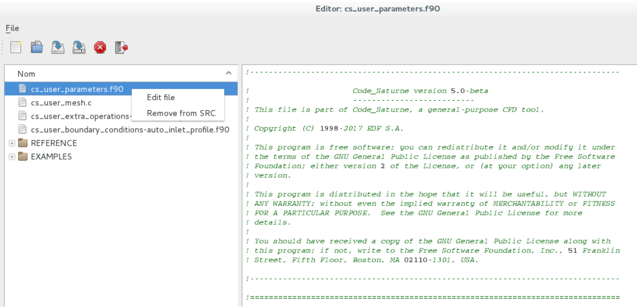
- → Manage User functions within the GUI!
- → View log files of a run

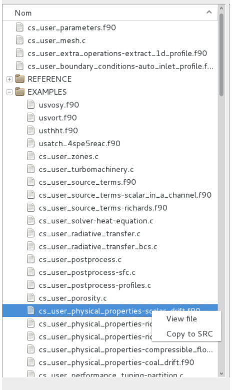


Overhaul of the Graphical User Interface (GUI)

Improving the global workflow of a study

- → Manage User functions within the GUI!
- → View log files of a run



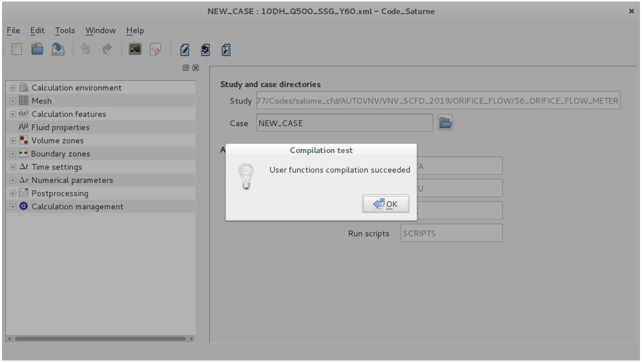


Overhaul of the Graphical User Interface (GUI)

Improving the global workflow of a study

→ Manage User functions within the GUI!

→ View log files of a run

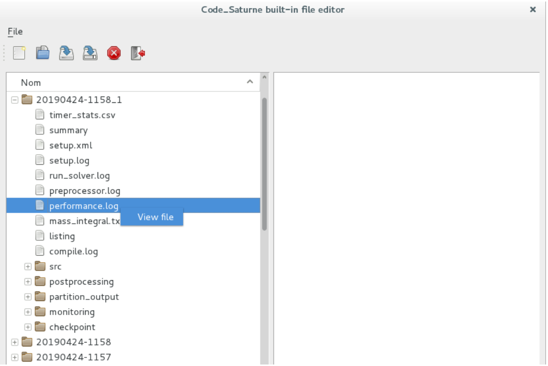


Overhaul of the Graphical User Interface (GUI)

Improving the global workflow of a study

→ Manage User functions within the GUI!

→ View log files of a run



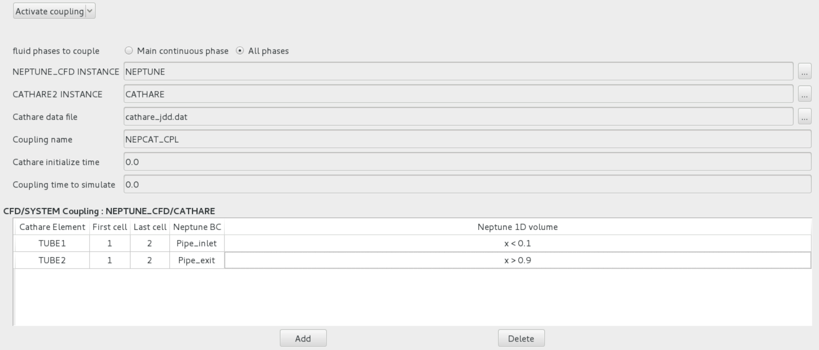
Coupled NEPTUNE_CFD/CATHARE2 cases

Creating a coupled NEPTUNE_CFD/CATHARE2 case

neptune_cfd create -c NEPTUNE –cathare CATHARE

Case parameters handling

Coupling can be handled using GUI and coupling_parameters.py file



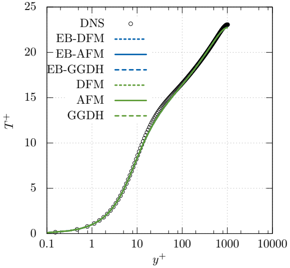
Turbulence heat flux modelling

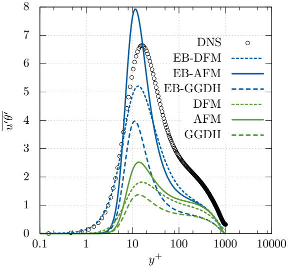
→ **Elliptic Blending Differential Flux Models for scalars** (EBDFM `ityturt = 31`)
→ Give very good results, especially wall bounded and flow with buoyancy

Similarly to EB-RSM the scrambling/dissipation terms $\Phi^*_{ij}/\varepsilon_{ij}$ are computed *via* the parameter α_θ ($\alpha_\theta \in [0,1]$). It is used to **blend the fully turbulent and the viscous regions** (DFM $\Longleftrightarrow \alpha_\theta = 1$)

$$(\Phi^*_{i\theta} - \epsilon_{i\theta}) = (1-\alpha_\theta)(\Phi^w_{i\theta} - \epsilon^w_{i\theta}) + \alpha_\theta(\Phi^h_{i\theta} - \epsilon^h_{i\theta})$$

→ **Elliptic Blending version of GGDH and AFM** (EBGGDH `ityturt = 11` and EBAFM `ityturt = 21`)





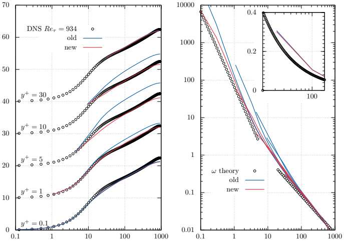
→ 1D channel flow (Dirichlet on T)

Major **improvement** for the prediction of **fluctuations** and **temperature variance** compared to **standard models**

Changes and new turbulence models

Changes in the k-ω-SST model

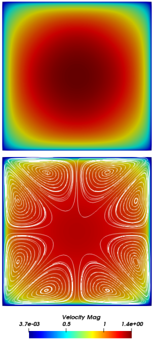
- → In the near wall region (see Menter): Dirichlet on $\omega_f = 10 \times \dfrac{6\nu}{\beta d}$, with d the cell center to cell face distance.
- → Now, to be coherent with other models and Menter: $\omega_f = \dfrac{5u_k^2}{\sqrt{C_\mu}\kappa\nu y^+}$ back to the previous BC with `ikwcln = 0`.



New quadratic k-ε model from Baglietto et al.

Theory of invariants → generalized the idea of constitutive relation for the Reynolds stresses
It becomes $\underline{\underline{\mathbf{R}}} = \mathbf{f}(\underline{\underline{\mathbf{S}}}, \underline{\underline{\mathbf{\Omega}}})$ where $\underline{\underline{R}}$ is the Reynolds stress tensor. When going to order 2 → quadratic models

- → Reproduction of secondary flows due to anisotropy
- → *E.g.* secondary flows in the square duct channel
- → Can be activated by setting `iturb = 23`



All y^+ treatment

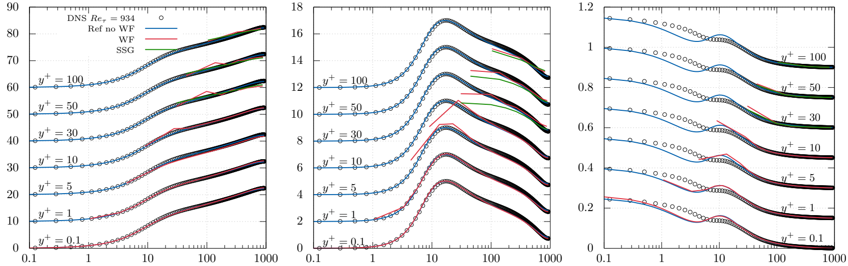
Wall functions for the EB-RSM are available with `iwallf = 7` and activated as default.

- → Simplified version of J.F Wald PhD (2016)
- → Ensure convergence towards standard EB-RSM when mesh is refined
- → Degenerate in a SSG-like model on high Reynolds meshes

→ New continuous wall function on the velocity and the velocity gradient

→ Similarly to $k-\omega$ SST model, in high Reynolds zones, we use $\varepsilon_f = \varepsilon_{I'} + d \left.\dfrac{\mathrm{d}\varepsilon}{\mathrm{dy}}\right|_{d/2}$ to impose apropriate BCs on ε. This is blended to the wall behaviour through a blending function $f_\varepsilon(y^+)$.

→ Homogeneous Neumann BC on Reynolds stresses, except $R_{12} = C_\alpha \sqrt{R_{11}R_{22}}$, $C_\alpha = 0.47$



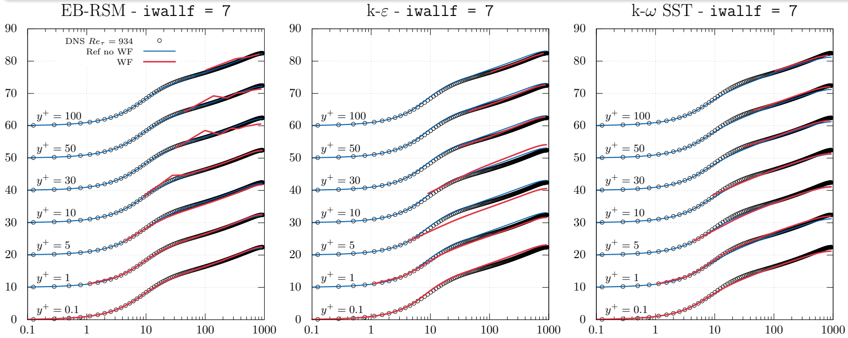
All y^+ treatment

New `iwallf = 7` can be activated with the $k-\omega$ SST.

- $\longrightarrow$ Default wall functions (`iwallf = 3`), can also be used as an all y^+ treatment
- $\longrightarrow$ Be careful to avoid buffer layer $5 < y^+ < 20$

A low Reynolds $k-\varepsilon$ model (Launder and Sharma, 1974) is available (`iturb = 22`).

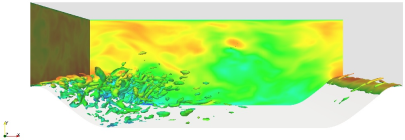
- $\longrightarrow$ Default wall functions (`iwallf = 3`), can be used as an all y^+ treatment
- $\longrightarrow$ New `iwallf = 7` can be activated

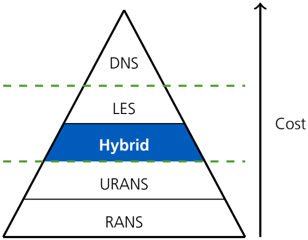


Delayed Detached Eddy Simulation

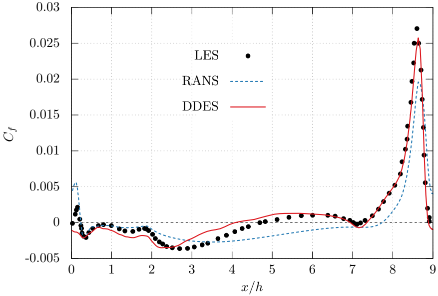
→ Hybrid RANS/LES model (Spalart 2006) for k-ω-SST model

→ **Mesh requirements between U-RANS and LES**





Mean skin friction coefficient



The turbulent length scale

$$\widetilde{L} = f_d L_{LES} + (1 - f_d)\, L_{RANS}$$

is blended with $f_d \in [0, 1]$

→ Improvement in velocity predictions

→ Better level of turbulent kinetic energy

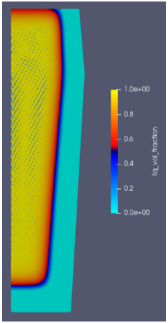
→ Reattachement point in agreement with LES

Solidification modelling within *Code_Saturne*

See dedicated presentation by C. Demay (EDF R&D/PRISME)

Objectives: study of carbon segregations during ingots casting

Modeling: fluid / solid mixing model with motionless solid phase



$$\begin{cases} \text{div}\,(\rho \underline{u}) = 0 \\ \dfrac{\partial \rho \underline{u}}{\partial t} + \underline{\text{div}}\,(\underline{u} \otimes \rho \underline{u}) = -\underline{\nabla} P + \underline{\text{div}}\, \mu_l \underline{\underline{\nabla} u} \\ \qquad - \dfrac{\mu_l}{K} \underline{u} + \rho^b (T, C_l) \underline{g} \\ C_p \left(\dfrac{\partial \rho T}{\partial t} + \text{div}\,(T \rho \underline{u}) \right) = \text{div}\, \lambda \underline{\nabla} T - \dfrac{\partial \rho g_l L}{\partial t} \\ \dfrac{\partial \rho C^\alpha}{\partial t} + \text{div}\,(C^\alpha \rho (1 + \eta^\alpha) \underline{u}) = \text{div}\, K \underline{\nabla} C^\alpha \end{cases}$$

Microsegregation model, binary phase diagram ($g_l(T, C)$) and Boussinesq Hyp.

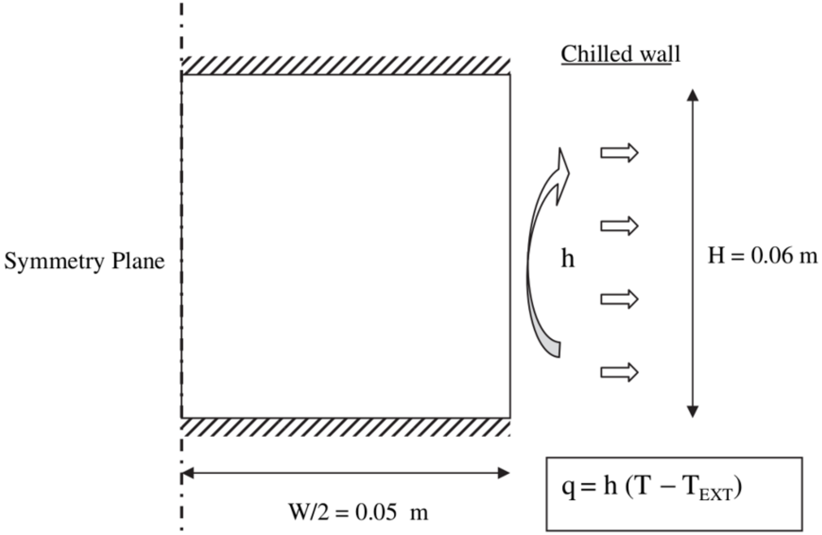
Work in progress combining several features:

- → PISO-like sub iterations (10)
- → Buoyancy: C and T are solved in the sub-iterations (`is_buoyant` field key word)
- → Uses "improved hydrostatic treatment"
- → Uses "porous" modelling (to cancel mass fluxes in the solid zones)
- → Conjugated heat transfer with internal coupling

Work done within M.-A. Rasendra internship and continued with EDF China and a thesis.

Solidification modelling within *Code_Saturne*

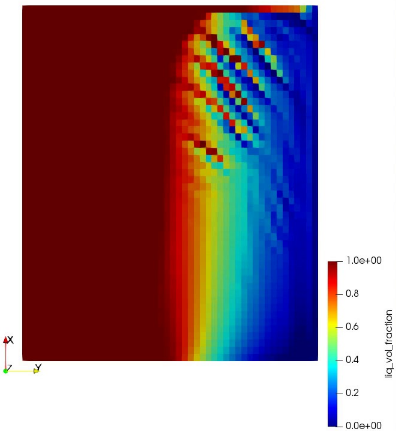
Validation testcase preview (versus SOLID)...



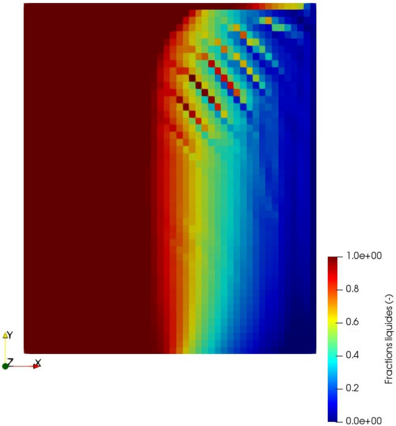
Validation testcase preview (versus SOLID)...

Liquid fraction g_l

Time: 204.010000



Code_Saturne

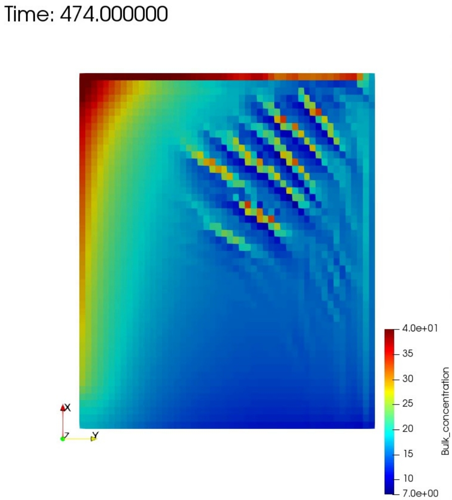


SOLID

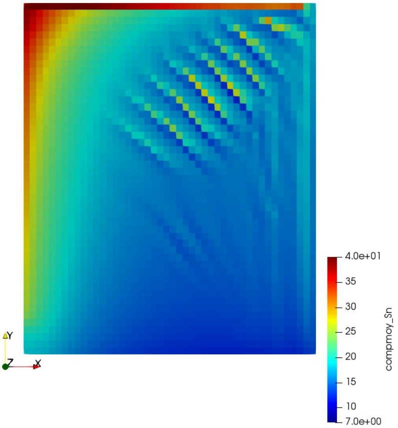
Solidification modelling within *Code_Saturne*

Validation testcase preview (versus SOLID)...

Carbon *C*



Code_Saturne

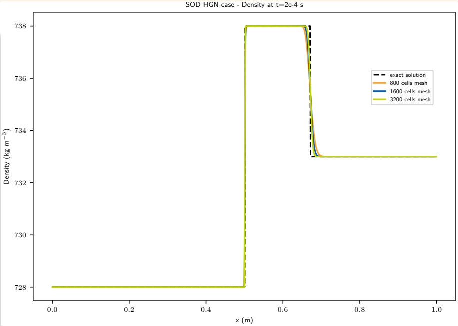


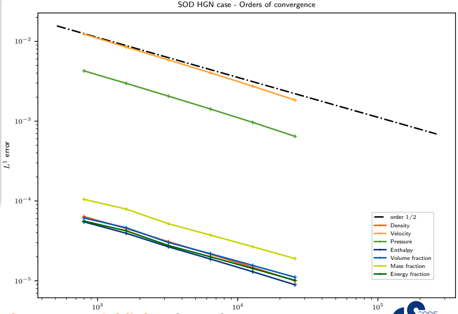
SOLID

Add a compressible 2-phase homogeneous model

Work of O. Hurisse (EDF R&D/MFEE)

- → fractional step method sharing mass, momentum, energy balance steps with single phase compressible algorithm
- → convection and source terms (relaxation towards equilibrium) step for each fraction (volume, mass, energy) follow
- → thermodynamic of the mixture is generic
- → each phase thermodynamics follows a stiffened gas EOS (parameters are at hand for the user)
- → relaxation time scale of return to equilibrium also at hand for the user
- → the model can be activated by setting to 2 the physical model parameter `cs_glob_physical_model[CS_COMPRESSIBLE]`

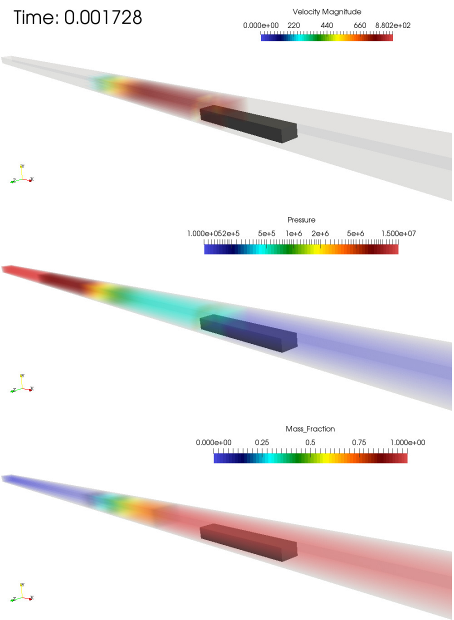




Add a compressible 2-phase homogeneous model

Work of O. Hurisse (EDF R&D/MFEE)

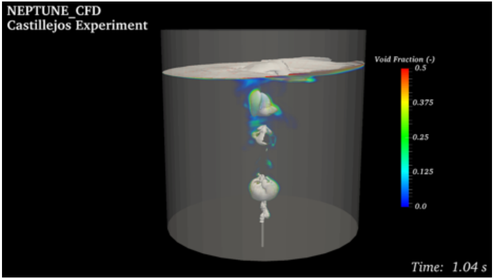
- → fractional step method sharing mass, momentum, energy balance steps with single phase compressible algorithm
- → convection and source terms (relaxation towards equilibrium) step for each fraction (volume, mass, energy) follow
- → thermodynamic of the mixture is generic
- → each phase thermodynamics follows a stiffened gas EOS (parameters are at hand for the user)
- → relaxation time scale of return to equilibrium also at hand for the user
- → the model can be activated by setting to 2 the physical model parameter `cs_glob_physical_model[CS_COMPRESSIBLE]`



Two major modelling advances in NEPTUNE_CFD

Towards the modelling of two-phase flow regime transition

Generalized Large Interface Method (GLIM) available and validated on a collection of adiabatic two-phase flows with different regimes (see technical presentation)

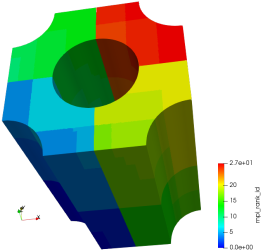


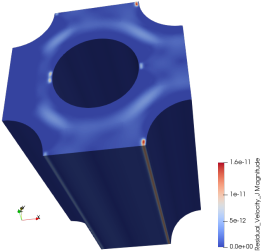
Modelling the motion of solids in two-phase flows

An innovative method to simulate the motion of solid structures (user- or flow-imposed) is now available (see dedicated presentation + demo)

Changes in default option of linear solvers

- → Default solver for convection-diffusion changed to symmetric Gauss-Seidel ⇒ **Performance gain** slightly less than 2 expected
- → On BUNDLE benchmark case (12M cell variant), Pressure solution is unchanged at 202 seconds, velocity solution goes from 298 to 95 seconds, k from 115 to 43, epsilon from 31 to 12, total elapsed time from 893 to 606 seconds.
- → BiCGStab alternatives are usually slower (except in cases where Jacobi convergence is slow, especially in the first iterations or for internal coupling).
- → For internal coupling, Jacobi or Gauss-Seidel is not well adapted to resolution of systems with purely diffusive zones
  - → BiCGStab or BiCGStab2 recommended in these zones
  - → since occasional divergence of these solvers is observed (1 every 500 to 1500 time iterations), temporary fallback to slower, but more robust GMRES is applied automatically.



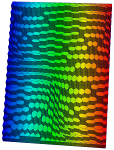


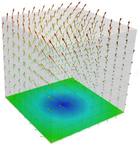
Jacobi

CDO roadmap (See Groundwater flow pres. by J. Bonelle & R. Lamouroux)

Bonelle's PhD
Diffusion and Stokes in
$\underline{\mathrm{curl}}\,\underline{\mathrm{curl}}$ formulation

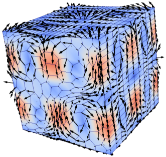
2011–2014





Cantin's PhD
Transport equations

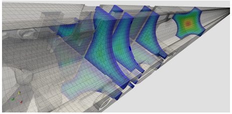
2013–2016



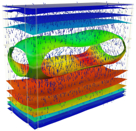
Milani's PhD
Stokes & Navier–Stokes equations

2017–2020

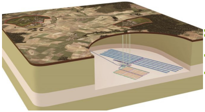
Diffusion | **Ground-water flows** | **Mesh deformation** | **Laminar flows** | **Turbulent flows**



Wall distance



First industrial study



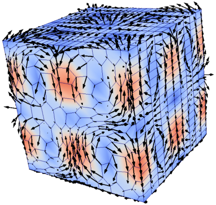
CIGEO industrial study

Grand challenge

CDO News

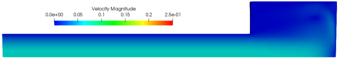
↦ **Stokes** for Face-based schemes

- Towards (Navier–)Stokes
- Fully coupled (Velocity-Pressure) algorithm
- Steady-state algorithm available



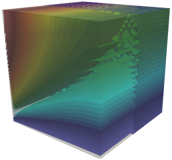
↦ **Vector-valued Laplacian** in Vertex-based schemes

- ALE module - See dedicated presentation by J. Harris (ENPC)



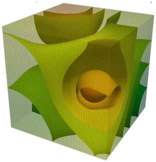
↦ Adding more **general boundary conditions**

- Robin and sliding boundary conditions



↦ **Volumetric interpolation**

- Enforcement of values inside the domain

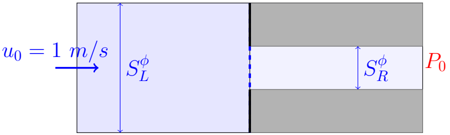


Porous modelling

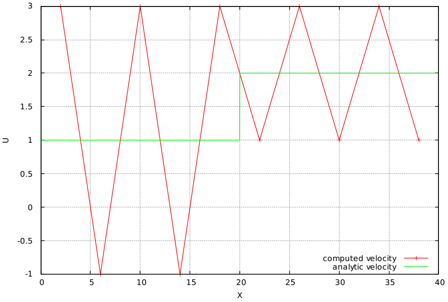
Work done in C. Colas PhD - see his poster for more on the integral formulation

- → Model a fluid section jump in an incompressible flow.
- → Tools: integral formulation `cs_glob_porous_model = 3` with discontinuous porosity.

- → Situation before: uncorrect result due to linear interpolation of cell values to compute interface values (valid for C^1 fields).



Channel with constriction

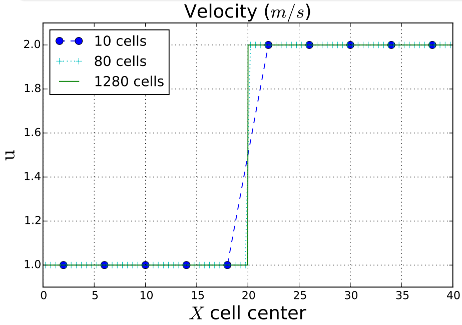


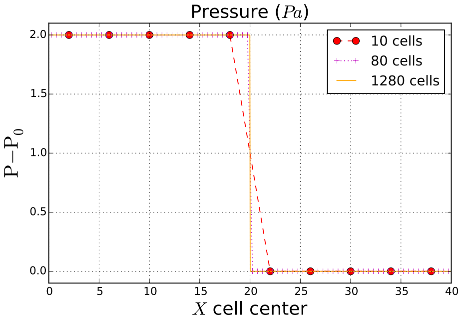
Porous modelling

Work done in C. Colas PhD - see his poster for more on the integral formulation

→ Solution: modified interpolation for interface values (field not differentiable) using a local steady balance on the dual sub-cells.

→ Situation after: the piecewise constant velocity and pressure fields are recovered with the solver precision.





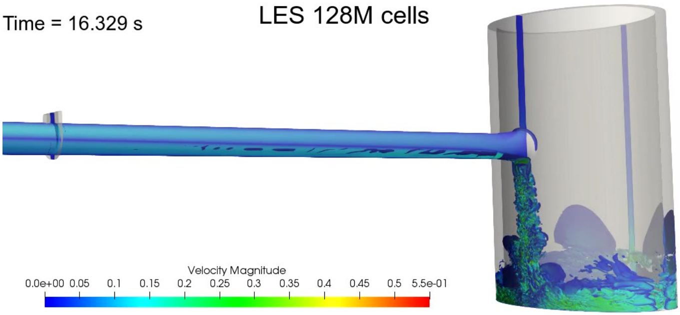
N.B. Rhie & Chow filter should be used and adapted when solving the pressure to avoid spatial oscillations.

2^{nd} order time scheme for variable density flow

→ Major changes in time stepping to ensure 2^{nd} order in time for variable density flow if 2^{nd} order time scheme is enabled. Scheme is similar to (C. Pierce, 2000).

- Buoyant scalars and density update can be included in velocity-pressure loop. To enable it, use `is_buoyant` field key word.
- The momentum equation is staggered in time, i.e. when 2^{nd} order is enabled, velocity is solved from time $n - \frac{1}{2}$ to $n + \frac{1}{2}$.

Work done in collaboration with C. Flageul.

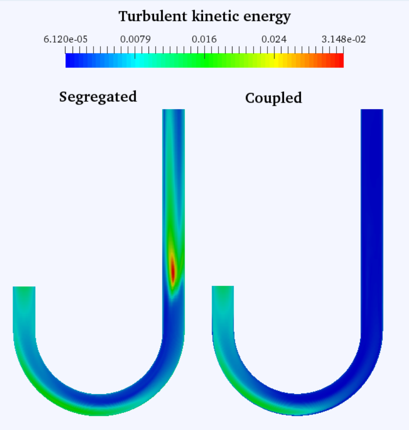


OECD Cold Leg Mixing CFD-UQ benchmark - R. Camy (EDF/DT).

Coupled solver for DRSM turbulence models (default)

→ Goal: increase linear solver robustness, and ensure realisability of $\underline{\underline{R}}$.

Comparison between segregated and coupled versions



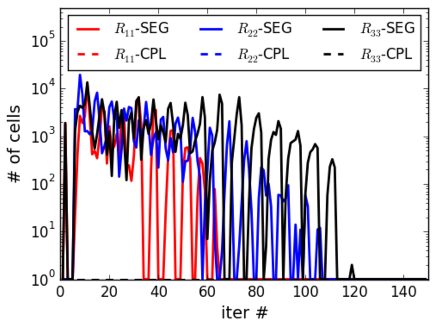
→ New time scheme allowed by the coupled solver in which more terms are made **implicit**.

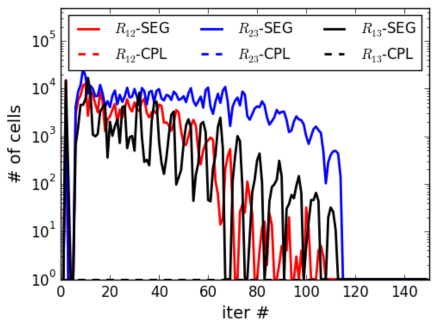
Example: 100 iterations for each case from same initial state:

→ Faster convergence for the coupled version.

Coupled solver for DRSM turbulence models (default)

Improvement of the linear solver efficiency: **no clipping** at all for any of the components (dashed lines).



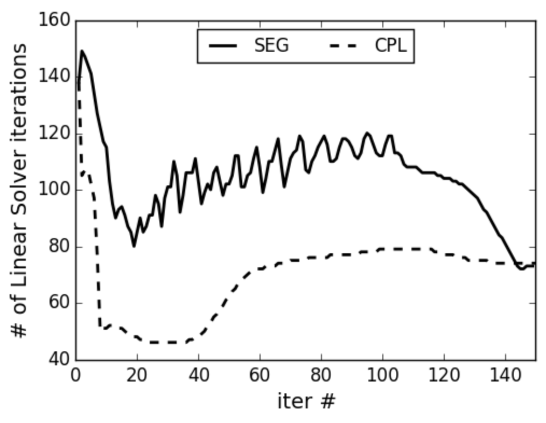


Number of clipped cells for each component of $\underline{\underline{R}}$.

Total number of cells ≈ 110 000.

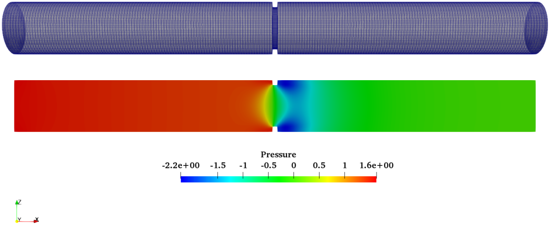
Coupled solver for DRSM turbulence models (default)

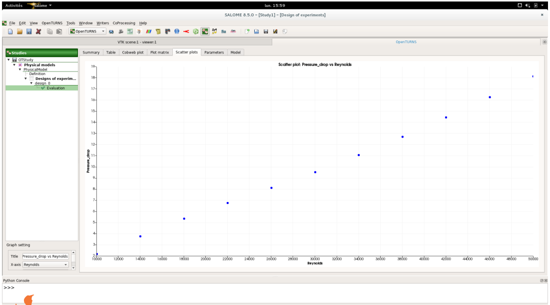
Improvement of the linear solver efficiency: a better convergence of the linear solver



→ Number of required iterations divided by 2

→ Smooth evolution of the number of iterations workflow example: compute $C_d = \dfrac{\Delta P}{\rho U^2}$ on an orifice plate model.





1. define input ($\boldsymbol{Re}$) and output ($\Delta \boldsymbol{P}$) inside *Code_Saturne*'s GUI
2. switch to OpenTURNS mode available in *Code_Saturne*'s GUI in Salome_CFD
3. define computational resources (desktop, **distant HPC cluster**) for a unitary run
4. switch to OpenTURNS module and for example, define a DOE, here a set of $\boldsymbol{Re}$ values
5. run all evaluations
6. postprocess results inside OpenTURNS, here C_d.

MEDCoupling support

MEDCoupling:

A library which is co-developed by EDF and CEA within the SALOME project. It is centered around mesh and fields manipulation, including interpolation.
Offers powerful functionnalities such as 3D/3D, 2D/2D or 3D/1D interpolation tools.

What is implemented:

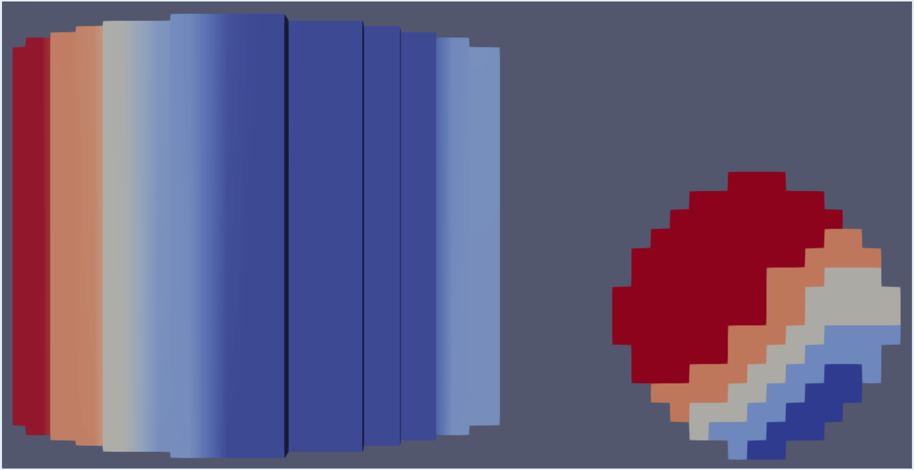
$\longrightarrow$ Sequential and Parallel remapping

$\longrightarrow$ Parallel (MPI-based) code coupling

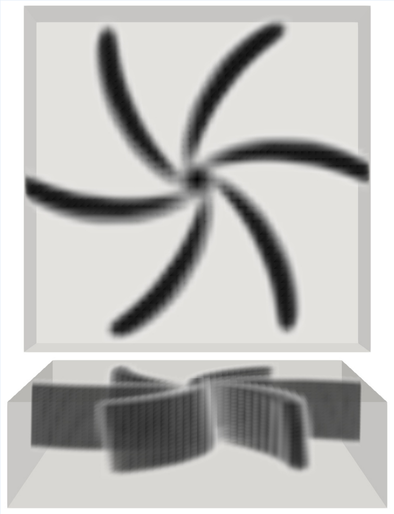
MEDCoupling support

Example 1: 2D fields

Imposing a temperature field at an inlet



Time and space dependent porosity field



→ Configuration: a rotor in a fluid domain

→ On the fly computation of the fluid volume during the simulation.

Get sources or binaries fitting your needs!

Git repository

→ Internal repository on PAM GitLab forge:
https://gitlab.pleiade.edf.fr/salome_cfd and subprojects.

→ development tab on code-saturne.org now links to
https://github.com/code-saturne/code_saturne public git mirror.

- Bug reports and feature requests on matching issues section.
- for confidential data or issues, use of the PAM GitLab issue tracker or saturne-support e-mail is possible.

→ GitHub also provides a Wiki (to be populated).

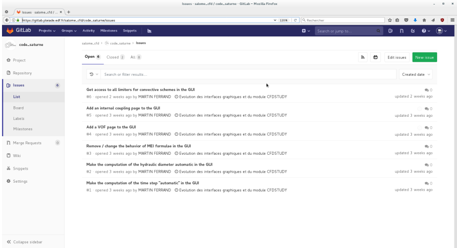
Get Salome_CFD binaries

→ on *Code_Saturne* website
https://www.code-saturne.org/cms/salome-cfd

Conclusion messages

Code_Saturne is NOT a black-box!
Ask what you want! Merci pour toutes ces années d'utilisation !

internal issue tracker (EDF users)
https://gitlab.pleiade.edf.fr/salome_cfd/code_saturne/issues



external issue tracker (public)
https://github.com/code-saturne/code_saturne/issues