Code_Saturne user meeting: on the road to version 6.0

Code_Saturne development team[1],

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

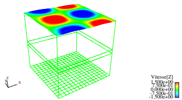
05/04/2018

Outlines

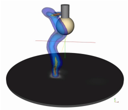
Development of *Code_Saturne* at EDF

Multiphysics modules merged into *Code_Saturne* framework

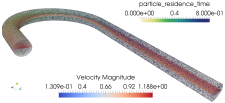
Arbitrary Lagrangian Eulerian (ALE)



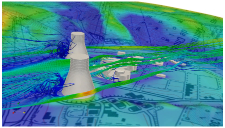
Electric Arcs



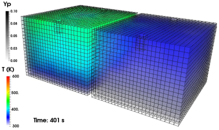
Lagrangian particle tracking



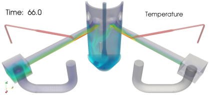
Atmospheric flows



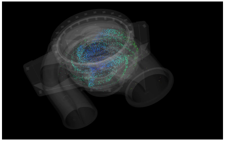
Fire modelling



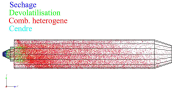
Thermohydraulics for Nuclear applications



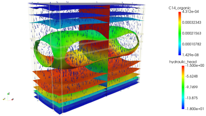
Turbomachinery



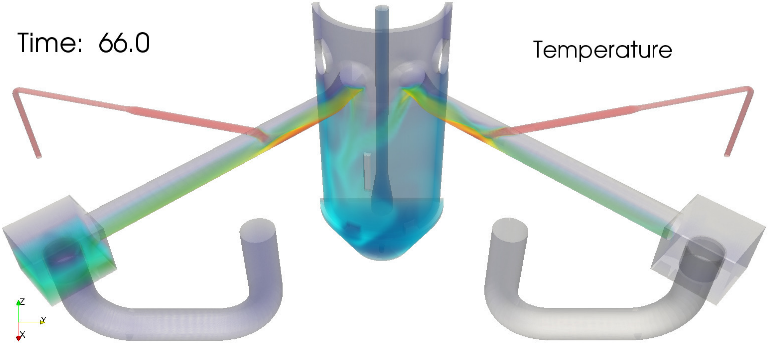
Combustion (fuel, coal, gas)



Groundwater flows



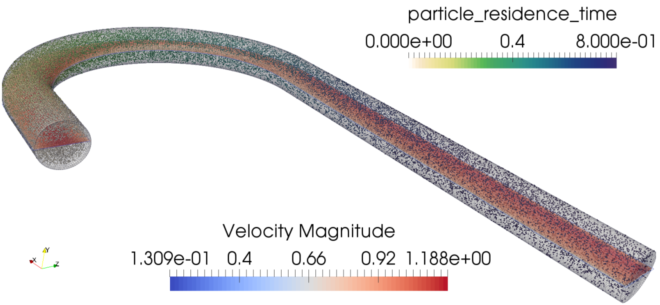
Thermohydraulics for nuclear applications



$$
\left\{
\begin{aligned}
&\frac{\partial \rho}{\partial t} + \text{div}\, \rho \underline{\overline{u}} = 0 \\
&\frac{\partial \rho \underline{\overline{u}}}{\partial t} + \underline{\text{div}}\,\left(\underline{\overline{u}} \otimes \rho \underline{\overline{u}}\right) = -\underline{\nabla}\overline{P} + \underline{\text{div}}\left(\mu\left(\underline{\underline{\nabla}}\,\underline{\overline{u}} + \underline{\underline{\nabla}}\,\underline{\overline{u}}^T\right)\right) + \rho \underline{g} - \underline{\text{div}}\left(\rho \overline{\underline{u}' \otimes \underline{u}'}\right)
\end{aligned}
\right.
$$

$$
C_p\left(\frac{\partial \rho \overline{T}}{\partial t} + \text{div}\,\left(\overline{T}\rho \underline{\overline{u}}\right)\right) = \text{div}\,\left(\lambda \underline{\nabla}\overline{T}\right) - C_p \text{div}\,\left(\rho \overline{T' \underline{u}'}\right)
$$

Lagrangian particle tracking



- Simulation of polydispersed particle-laden turbulent flow
- Moments/PDF (Euler/Lagrange) approach
- Frozen field, one-way or two-way coupling
- Dedicated models for particle heat transfer, droplets evaporation, particle deposition

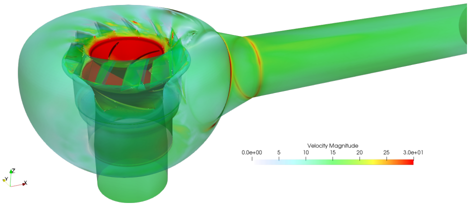
$$d\underline{X}_p = \underline{U}_p dt$$

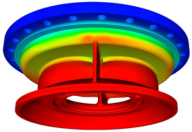
$$d\underline{U}_p = -\frac{1}{\rho_f}\underline{\nabla P}_f dt - \frac{\underline{U}_p - \langle \underline{U}_f \rangle}{T_L} dt + \sqrt{C_0 \varepsilon_f} d\underline{W}$$

where

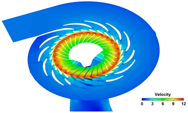
$$T_L = \frac{1}{\frac{1}{2} + \frac{3}{4}C_0} \frac{k_f}{\varepsilon_f}$$

Turbomachinery

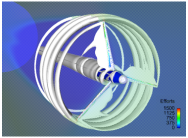




Temperature at a safety pump suction



Flow field in a Francis99 turbine



Q-criteria around the MEXICO wind turbine

Application examples

- **Reactor Coolant Pump:** performance studies, hydraulic loads as input of mechanical calculations (*code_aster*)
- **Safety pumps:** thermal transient, lagrangian particle tracking toward guiding and sealing systems
- **Renewable energy:** hydraulic turbines, wind turbines

Turbomachinery capabilities of *Code_Saturne*

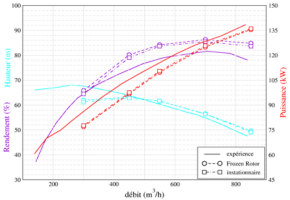
- Rotor/stator interaction modelling
  - Frozen rotor
    Frozen geometry, rotor rotation by boundary conditions and source terms

    Often sufficient for performance studies
  - Sliding mesh
    Effective rotation of the rotor mesh

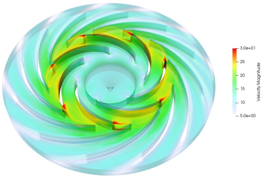
    Useful for finer unsteady analysis
- Implementation in *Code_Saturne*
  - Mesh joining
    Conservative, requires sufficiently clean mesh interfaces.

    Robustness improved by retries with perturbed position (V5.2)
  - Coupling of boundary conditions
    Slower initialization, but good steady-state results, more tolerant of dirty meshes.

    Now defined in a single case, allowing easy setup (V5.1-5.2)
- Dedicated post-processing macros
  Cylindrical coordinates, manometric head and torque available
- On-line documentation
  www.code-saturne.org/doxygen/src/turbomachinery.html



Performance characteristic of Gourdin pump

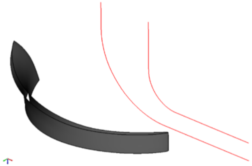


ERCOFTAC validation test case: Genova pump

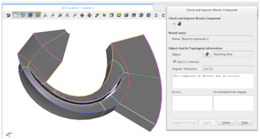
Turbomachinery meshing facility

Complex bloc-structured mesh in turbomachinery:

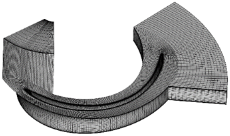
⇒ Dedicated mesh processor script example in the SALOME plateform (*coming soon*)



CAD input files

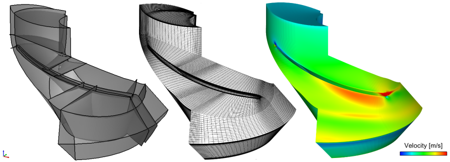


Specific block-structured partition of the flow domain (*SALOME GEOM*)



Meshing of the flow domain (*SALOME SMESH*)

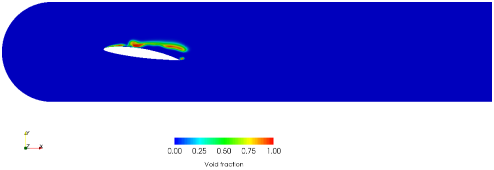
Toward an integrated workbench for turbomachinery into SALOME_CFD...



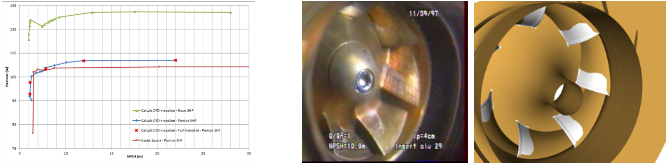
Cavitation modelling

Homogeneous mixture model with void fraction transport equation and vaporisation/condensation source term

- $\dfrac{\partial \alpha}{\partial t} + \text{div}\,(\alpha \underline{u}) = \dfrac{\Gamma_V}{\rho_V}$
- $\rho = \alpha \rho_V + (1-\alpha)\rho_l$
  $\mu = \alpha \mu_V + (1-\alpha)\mu_l$
  constant $\rho_V, \rho_l, \mu_V, \mu_l$
- $\Gamma_V(\alpha, P)$ source term: Merkle model

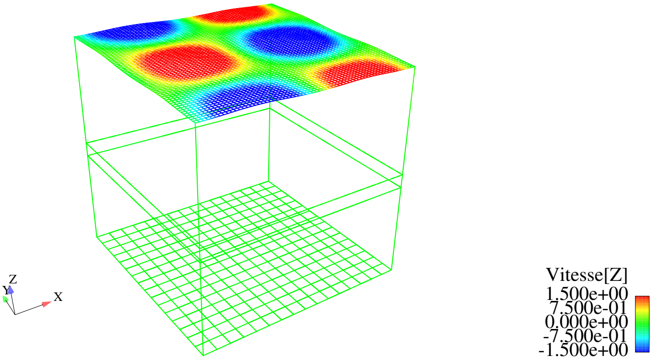


Cavitating flow around a hydrofoil



Application to turbomachinery (SHF pump): computation/experiment comparison of the NPSH curve (left) and cavitation sheets visualization (right)

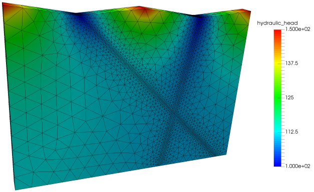
Arbitrary Lagrangian Eulerian (ALE)



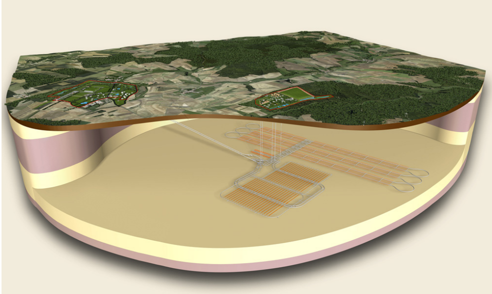
- Solve mesh displacement with mesh deformation imposed at the boundaries
- Or impose the displacement of any node
- Fluid-structure interaction
- Free surface modelling (no wave break)

Groundwater flows

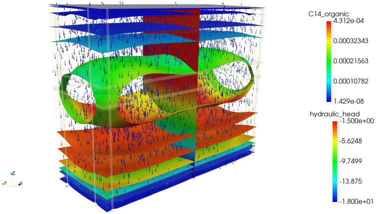
- Richards equation is solved (Darcy law injected in mass equation)
- Species mass fractions transport
- Heterogeneous and anisotropic permeability
- Large meshes: several hundred million cells
- Possible long physical period: up to a million years (1 time step $\approx$ 1000 years)



2 geological fractures (lower permeability)

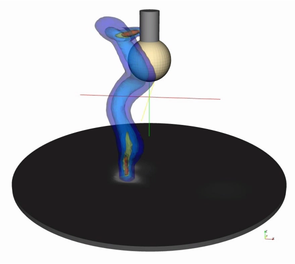


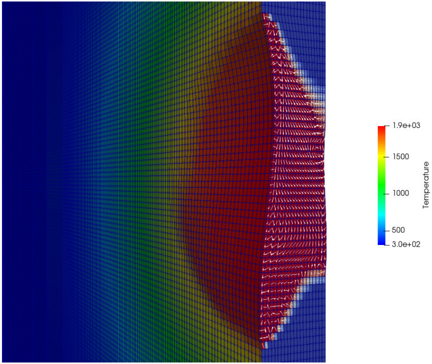
source www.andra.fr, CIGEO project



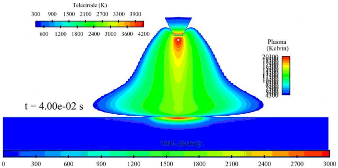
Iso-surfaces of C^{14} on a storage site

Electric arcs and Joule effect

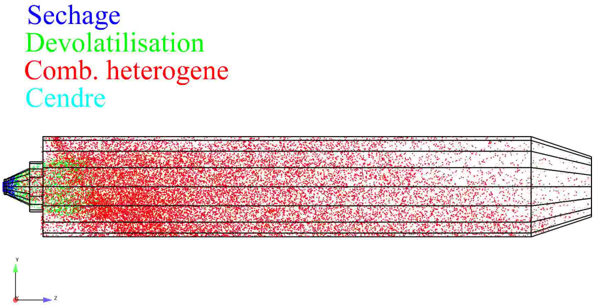




- Includes both Joule effect and EM forces modelling which can induce flow and 3D unsteady behaviour of the arc.
- Multiphysics modelling of arc welding: cathode, plasma, melting solid (anode) with ALE and internal coupling.
- Can solve complex potential for the treatment of triphase alternative current.

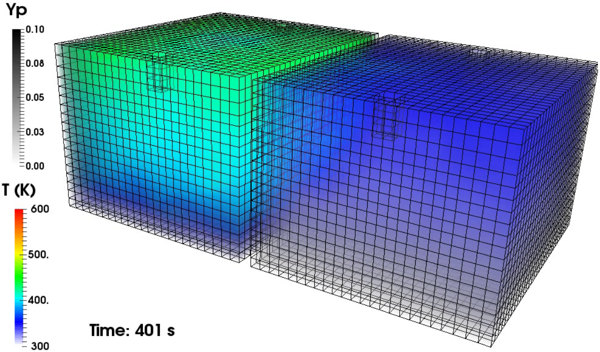


Combustion



- Pulverized coal, heavy fuel, biomass combustion
- Gas combustion
- NOx prediction,
- Slagging, fouling

Fire modelling



Dodecane fire

- Weakly compressible algo. for gas mixture
- Free inlet
- Soot models
- Radiative transfer models (DOM and P1)

Volume of Fluid module

- Mixture dynamics - incompressible Navier-Stokes equations:

$$\mathrm{div}\,(\underline{u}) = 0$$

$$\frac{\partial}{\partial t}(\rho \underline{u}) + \underline{\mathrm{div}}\,(\underline{u} \otimes \rho \underline{u}) = -\underline{\nabla} P + \underline{\mathrm{div}}\,\underline{\underline{\tau}}$$

- Homogeneous mixture:

$\rho = \alpha \rho_v + (1-\alpha)\rho_l$ and $\mu = \alpha \mu_v + (1-\alpha)\mu_l$

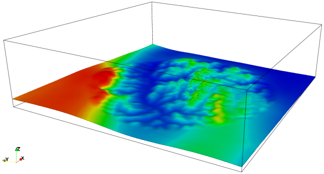
- Void fraction pure convection:

$$\frac{\partial \alpha}{\partial t} + \mathrm{div}\,(\alpha \underline{u}) = 0$$

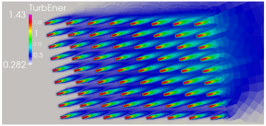
with Compressive Interface Capturing

Scheme for Arbitrary Meshes (CICSAM)

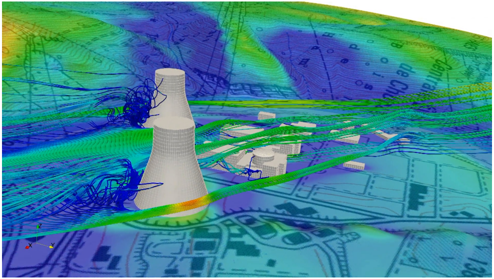
Atmospheric flows



Wind potential estimates on complex terrain



Wake effects on an offshore wind farm



Atmospheric dispersion

Lagrangian stochastic modelling for pollutant atmospheric dispersion

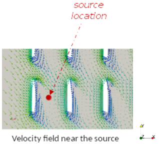
MUST (Mock Urban Setting Test) campaign (idealized city)

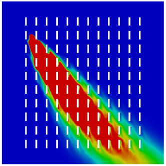
Model equations (Simplified Langevin model, cf. Pope 2000)

$$d\underline{X}_p = \underline{U}_p dt$$
$$d\underline{U}_p = -\frac{1}{\rho_f}\underline{\nabla} P_f dt - \frac{\underline{U}_p - \langle \underline{U}_f \rangle}{T_L} dt + \sqrt{C_0 \varepsilon_f} d\underline{W}$$

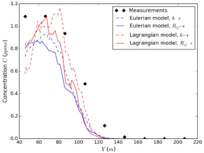
where

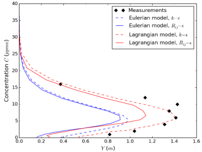
$$T_L = \frac{1}{\frac{1}{2} + \frac{3}{4} C_0} \frac{k_f}{\varepsilon_f}$$





Particle concentration (kg/m^3)





Concentration profiles (ppmv)

Announcement from the joint Lab CEREA

contact: CEREA Director pietro.bernardara@edf.fr

Air quality From the street to the room

Complete chain of CFD Modelling of air quality from the neigbourhood scale to the room

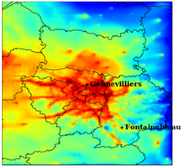
Coupling *Code_Saturne* atmospheric module to aeraulic simulations

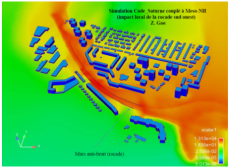
OPEN Positions

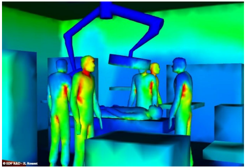
Post-doc “Grand Paris” project

Post-doc “Hopital Saint Louis” project

Phd on aeraulic CFD







Announcement from the joint Lab CEREA

contact: CEREA Director pietro.bernardara@edf.fr

CFD for energy application

Modelling wind fields, temperature and radiation for renewable energy and nuclear

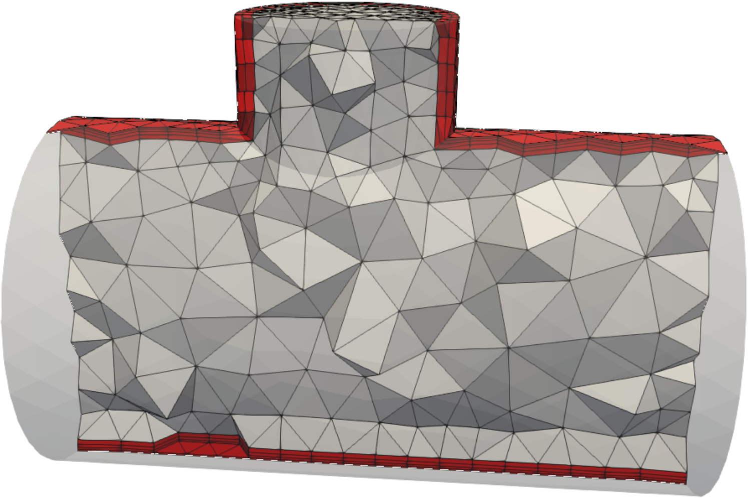
OPEN Positions

Phd position on Solar power and radiation CFD modelling

Ing. Temporary position on CFD modelling for industry (renewable and nuclear)

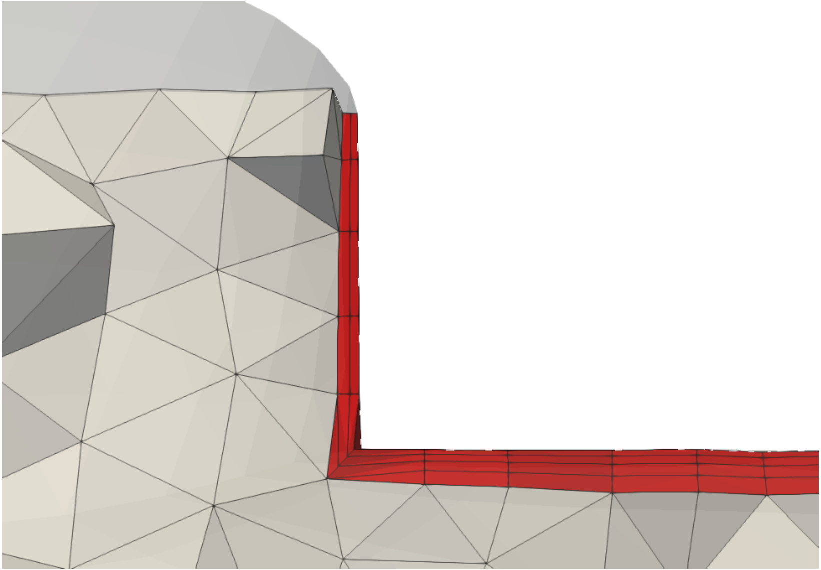
Preprocessing: automatic insertion of wall-layer cells

using CDO vertex based ALE solver



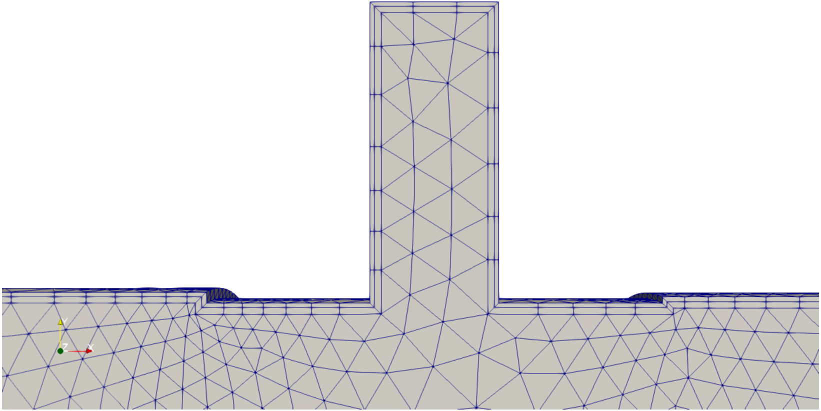
Preprocessing: automatic insertion of wall-layer cells

using CDO vertex based ALE solver



Preprocessing: automatic insertion of wall-layer cells

using CDO vertex based ALE solver



see function `cs_user_mesh-modify.c`.

Preprocessing: automatic insertion of wall-layer cells

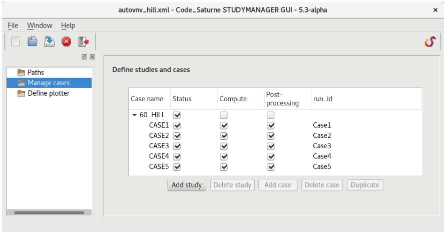
using CDO vertex based ALE solver

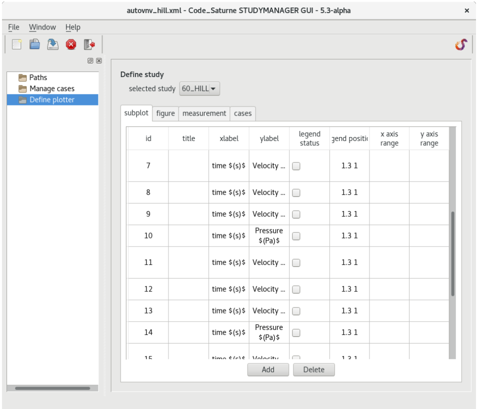
Evolutions in `studymanager`

- Separate compilation test and repository update features:
  - `-test-compilation` or `-t` to test compilation for all cases
  - `-update` or `-u` to update GUI parameter files and set paths in scripts `SaturneGui` and `runcase`
  - `-update-xml` or `-x` to update GUI parameter files only
- add **tags** feature:
  - `-with-tags` to only process runs with all passed tags
  - `-without-tags` to exclude runs with one of the passed tags.
- **Disable cases instead of exiting** when a run fails.
- **Removed** handling of **VTK-based scalar slice** outputs; graphical files may now be included, and using `Catalyst` to produce images is recommended.

```
code_saturne studymanager -f smgr.xml -r -n 5  -with-tags=coarse
```

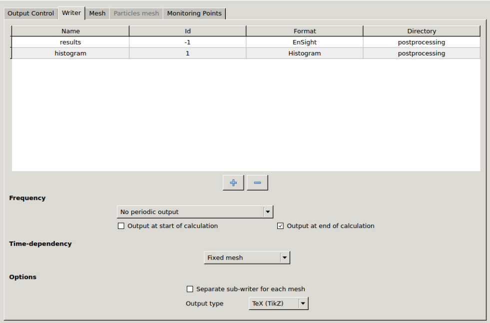
Evolutions in `studymanager`





Use it with its GUI!

New post-processing writer: histogram

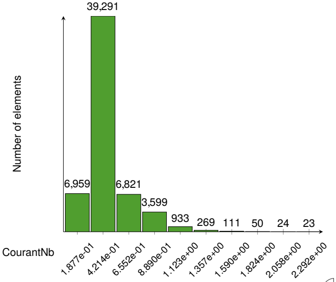


- available in GUI
- formats: txt, tex, png (based on vtk)

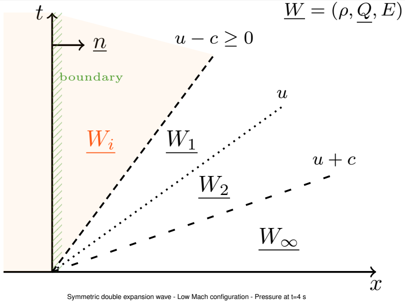
```
# Code_Saturne histograms
#
# time step id: 50
# time:          1.00000e+01
#
# Variable : CourantNb

   valeur minimale =      6.58066e-02
   valeur maximale =      2.42866e+00

     1 : [ 6.58066e-02 ; 5.38377e-01 [ = 46157
     2 : [ 5.38377e-01 ; 1.01095e+00 [ = 10506
     3 : [ 1.01095e+00 ; 1.48352e+00 [ =  1214
     4 : [ 1.48352e+00 ; 1.95609e+00 [ =   156
     5 : [ 1.95609e+00 ; 2.42866e+00 ] =    47
```

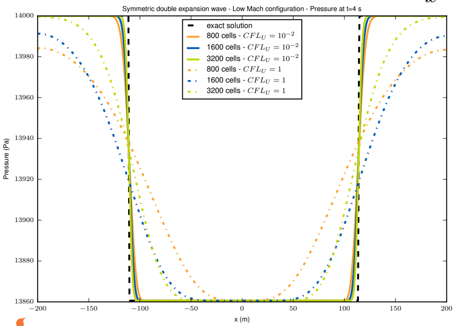


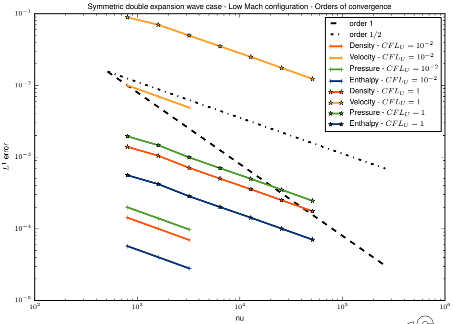
Compressible: supersonic outlet made implicit



$$\underline{W}_b = \underline{W}_i^{n+1}$$

homogeneous Neumann on P, $\underline{u}$, E.





use MEDCoupling (V5.2)

Example 1: 2D fields Example 2: 3D fields

MEDCoupling:

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

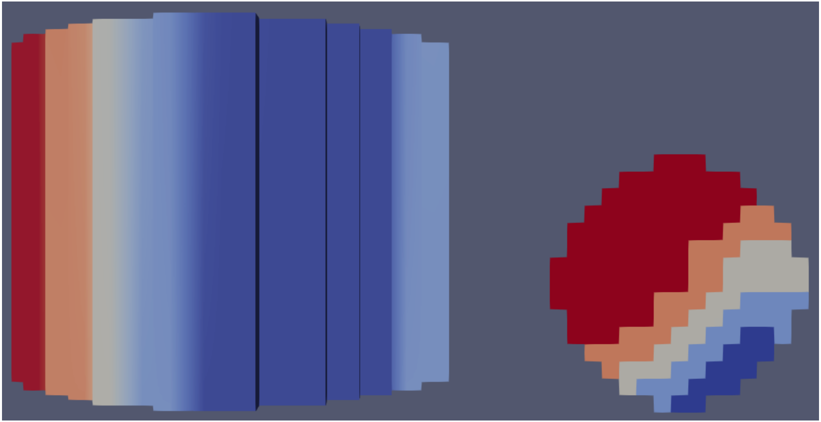
What is implemented:

- Sequential remapping
- Parallel (MPI-based) code coupling

use MEDCoupling (V5.2)

Example 1: 2D fields Example 2: 3D fields

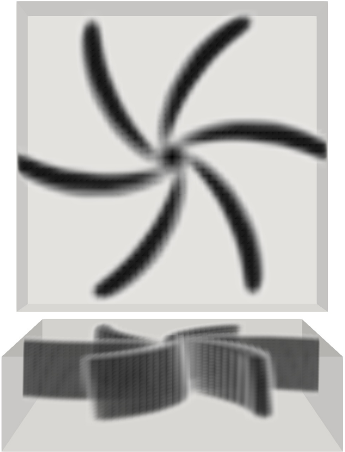
Imposing a temperature field at an inlet



use MEDCoupling (V5.2)

Example 1: 2D fields Example 2: 3D fields

Time and space dependent porosity field

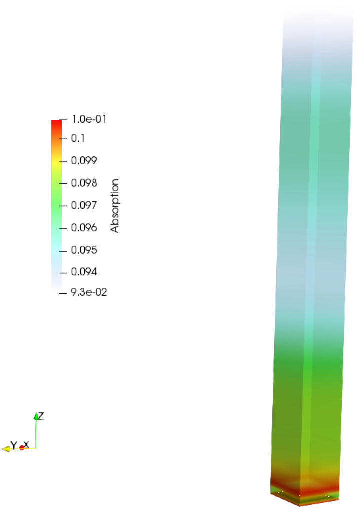


- Configuration: a rotor in a fluid domain
- On the fly computation of the fluid volume during the simulation.

Atmospheric module (V5.1)

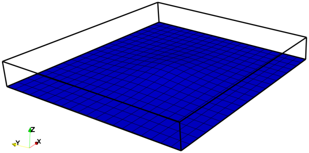
Add an atmostpheric infra-red absorption feature. Based on L. Makke's PhD work.

Activate it with `cs_glob_rad_transfer_params->atmo_ir_absorption = true` in `cs_user_radiative_transfer.c`.

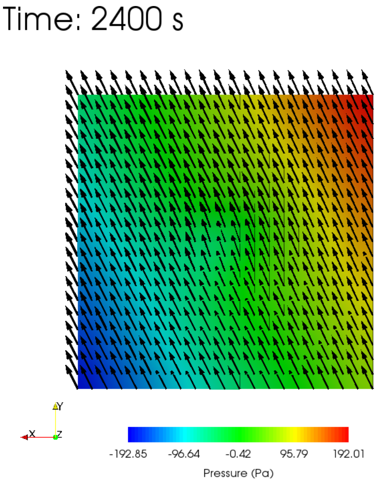


Atmospheric module (V5.2)

Add new BCs for open-boundary flow such as atmospheric flows. It consists in changing the solved pressure. This change results in adding a (constant over space) momentum source terme.

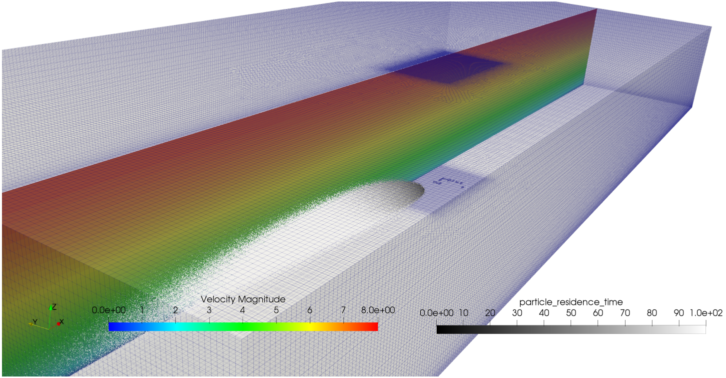


Flow over a hill



Lagrangian particle tracking

Refactored boundary injection and added volume injection



Both injection in the volume and injection density profiles are now available; see function `cs_user_lagr_in` in `cs_user_lagr_particle.c`.

Turbulence modelling

- **Elliptic Blending Differential Flux Models for scalars** (EBDFM `ityturt = 31`, V5.1)
  → 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`, V5.1)
  → Additional equation on the parameter α_θ to improve models behaviour at walls (see Dehoux 2012).

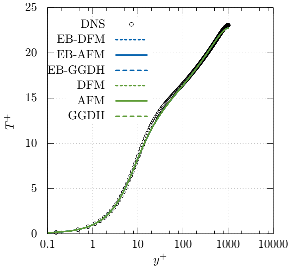
E.g. EB-GGDH (GGDH $\Longleftrightarrow$ $\alpha_\theta = 1$) thermal fluctuations are computed as follows:

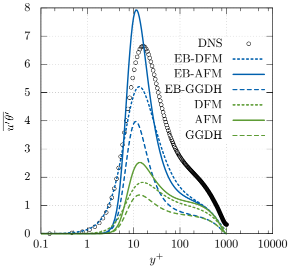
$$\overline{u'_i\theta'} = -C_\theta T \left[\overline{u'_i u'_j} \frac{\partial \overline{\theta}}{\partial x_j} + \chi \frac{\varepsilon}{k} \overline{u'_j\theta'} n_i n_j \right] \quad \text{with} \quad C_\theta = \frac{C'_\theta \sqrt{R}}{\alpha_\theta C_{1\theta} \sqrt{R^h} + (1 - \alpha_\theta) \sqrt{\mathrm{Pr}} C_{\varepsilon^w}}$$

and $\chi = (1 - \alpha_\theta) \left(C_{\epsilon^w} + C_{\phi^w} \right)$, $R = \alpha_\theta R_h + (1 - \alpha_\theta)\mathrm{Pr}$

Turbulence modelling: EB-DFM

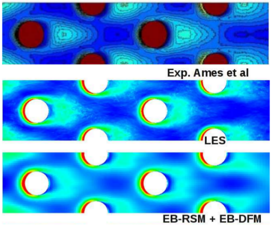
Validation case (1D channel flow, Dirichlet BC on T)

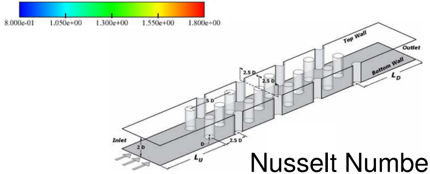




Major **improvement** for the prediction of **fluctuations** and **temperature variance** compared to **standard models**

Industrial/semi-academic configuration (Dehoux PhD 2012)

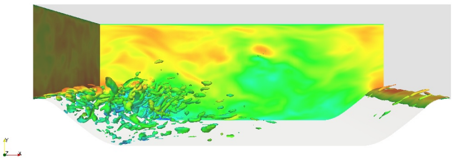


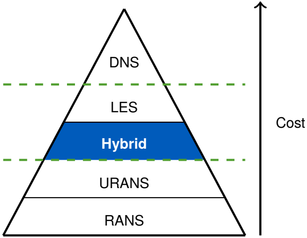


Nusselt Number along the Wall at $Re = 30000$

Delayed Detached Eddy Simulation (V5.1)

- Hybrid RANS/LES model (Spalart 2006) for *k-ω-SST* model
- Mesh requirements between U-RANS and LES



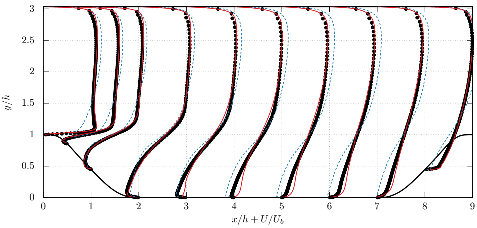


$$\widetilde{L} = f_d L_{LES} + (1 - f_d)\, L_{RANS} \quad \text{with} \quad \begin{array}{ll} f_d = 0 & \text{if } L_{LES} > L_{RANS} \\ f_d = f\,(S, \nu_T, y) \leq 1 & \text{otherwise} \end{array}$$

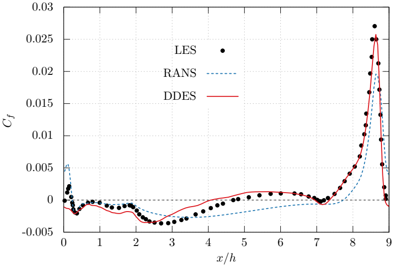
$$\frac{Dk}{Dt} = P_k - \underbrace{\frac{k^{3/2}}{\widetilde{L}}}_{\textit{modified}} + \frac{\partial}{\partial x_i}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right]$$

Delayed Detached Eddy Simulation

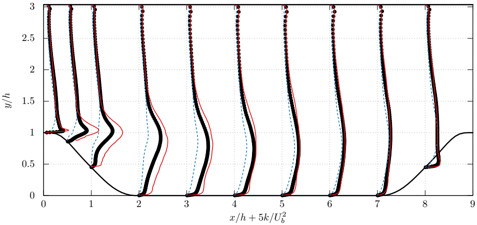
Mean velocity profiles



Mean skin friction coefficient



Mean turbulent kinetic energy profiles



- Improvement in velocity predictions
- Better level of turbulent kinetic energy
- Reattachement point in agreement with LES

Coupled solver for the DRSM turbulence models

(default option in V5.1)

$\underline{\underline{R}} = \overline{\underline{u}' \otimes \underline{u}'}$ is symmetric positive definite (SPD)

That is to say $\forall \underline{v} \neq \underline{0}$, $\underline{v} \cdot \underline{\underline{R}} \cdot \underline{v} = v_i R_{ij} v_j > 0$

(with Einstein convention on i, j).

A new time stepping for $\underline{\underline{R}}$

If $\underline{\underline{R}}^{n+1}$ is solution of

$$\left(\underline{\underline{R}} \cdot \underline{\underline{O}} + \underline{\underline{O}}^T \cdot \underline{\underline{R}}\right) + \underline{\underline{S}}_1 \otimes \underline{\underline{S}}_2 : \underline{\underline{R}} = \underline{\underline{S}}_3$$

with $\underline{\underline{O}}$ a positive tensor and $\underline{\underline{S}}_1$, $\underline{\underline{S}}_2$, $\underline{\underline{S}}_3$ SPD tensors, then $\underline{\underline{R}}^{n+1}$ is SPD.

Remarks:

- A part of the production term is of the form $\left(\underline{\underline{R}} \cdot \underline{\underline{O}} + \underline{\underline{O}}^T \cdot \underline{\underline{R}}\right)$
- The dissipation $\varepsilon \underline{\underline{1}}$ may be written as $\varepsilon / k \underline{\underline{1}} \otimes \underline{\underline{1}} : \underline{\underline{R}}$

Transport equation for the Reynolds stress tensor

Equation for the Reynolds stress tensor

$$\boxed{\begin{gathered}\rho \frac{\partial \underline{\underline{R}}}{\partial t} + \underline{\underline{\underline{\nabla}}}\,\underline{\underline{R}} \cdot (\rho \underline{u}) - \underline{\underline{\mathrm{div}}}\left(\mu \underline{\underline{\underline{\nabla}}}\,\underline{\underline{R}}\right) \\ = \\ \underline{\underline{d}} + \underline{\underline{\mathcal{P}}} + \underline{\underline{G}} + \underline{\underline{\Phi}} - \rho \underline{\underline{\varepsilon}} + \Gamma \underline{\underline{R}}^{in} + \underline{\underline{ST}}_{R_{ij}}\end{gathered}}$$

Goal:

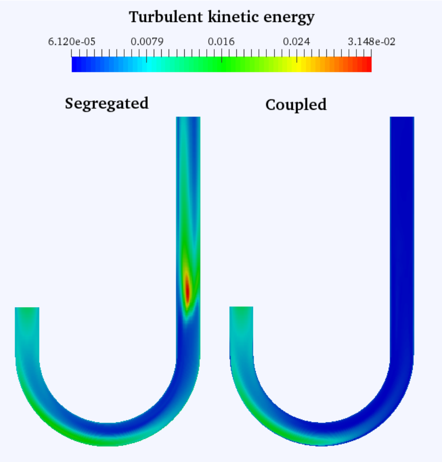
Increase the robustness of the linear solver, and ensure the realisability of $\underline{\underline{R}}$.

How to:

Implicit as many terms of the RHS as possible

Validation on the U-bend case

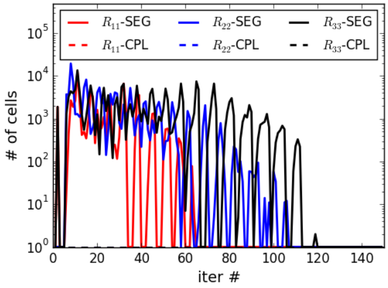
Comparison between segregated and coupled versions

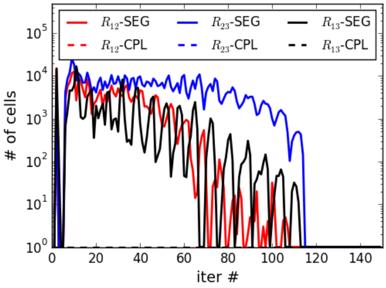


- 100 iterations for each case
- Same initial state
- Faster convergence for the coupled version

Validation on the U-bend case

Improvement of the linear solver efficiency: no clipping at all for any of the components.

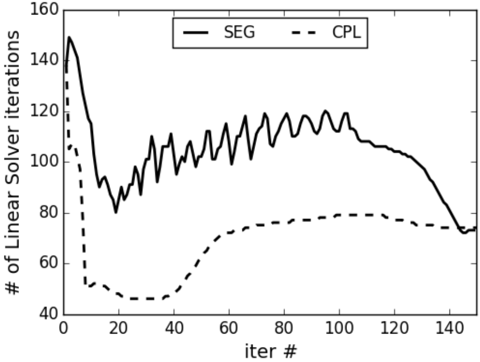




Number of clipped cells for each component of the $\underline{\underline{R}}$ tensor. Total number of cells $\approx$ 110 000.

Validation on the U-bend case

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



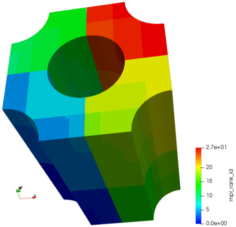
- Number of required iterations divided by 2
- Smooth evolution of the number of iterations

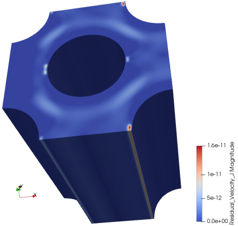
Default options changes in V5.1

- `[Linear solvers]` Switch from Jacobi to **processor-local symmetric Gauss-Seidel** as default type for non-symmetric systems.
- `[Turbulence DRSM]` Make **coupled solving** for DRSM turbulence models default option.
- `[Init. with RANS turbulence]` Make **advanced turbulence initialisation** for $k-\omega$, $Bl-v^2/k$ and EBRSM turbulence models default option.
- `[Dilatable algo.]` Make prediction of mass fluxes before momentum solving optional. This impact variable density algorithms (`idilat>=2`).
- `[Convection schemes]` Make vectorial and tensorial slope tests invariant by rotation and identical for all components (previous version available for the velocity with `isstpc = -1` in `var_cal_opt` structure). It is set by default.
- `[Convection schemes]` Add an option to blend with upwind when the slope test is activated. The key word is `blend_st` in `var_cal_opt` structure.

Some more information for linear solvers (V5.1)

- Default solver for convection-diffusion changed to symmetric Gauss-Seidel
- Performance gain slightly less than 2 expected
- On BUNDLE benchmark case (12M cell variant), Pression 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.
  - GMRES not yet available for vector variables





Jacobi

Add experimental options for robustness

Available in version 5.1, in common with NEPTUNE_CFD, see `cs_user_parameters.c`

- for warped faces (for `imrgra` =0 & 4):

  `cs_glob_mesh_quantities_flag |=`

  `CS_BAD_CELLS_WARPED_CORRECTION;`

- bad cell regularisation:

  `|= CS_BAD_CELLS_REGULARISATION;`

- face center modification:

  `|= CS_CELL_FACE_CENTER_CORRECTION;`

- cell center modification:

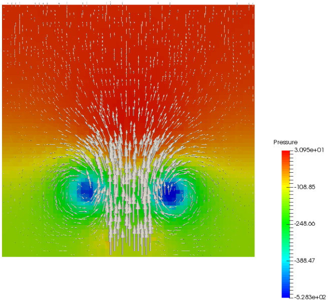
  `|= CS_CELL_CENTER_CORRECTION;`

- face distance clipping:

  `|= CS_FACE_DISTANCE_CLIP;`

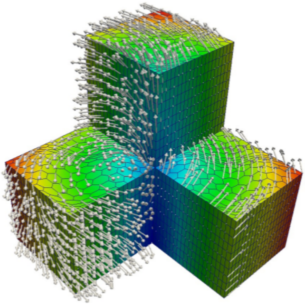
- reconstruction clipping:

  `|= CS_FACE_RECONSTRUCTION_CLIP;`

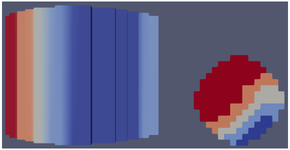


News

- Buoyant scalars and density updates in the velocity/pressure coupling algorithm



- Vector Laplacian for CDO-Face based
  $\longrightarrow$ Towards (Navier–)Stokes [POSTER]



- MEDCoupling: Use functionalities for interpolating from one mesh to another

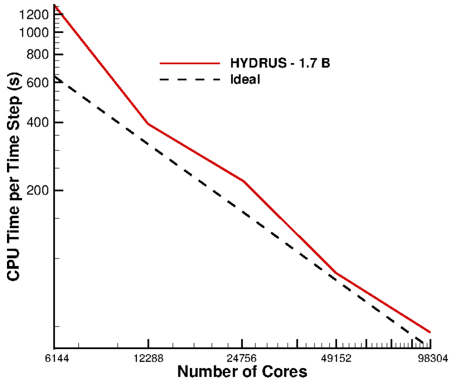
Stress tests on HPC capabilities of CDO schemes

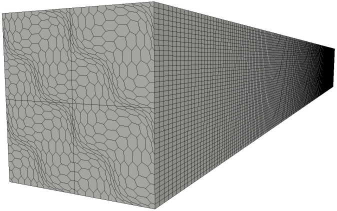
Groundwater flow applications

ARCHER CRAY supercomputer (UK)

STFC collaboration

[POSTER]



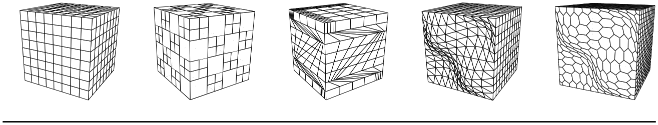


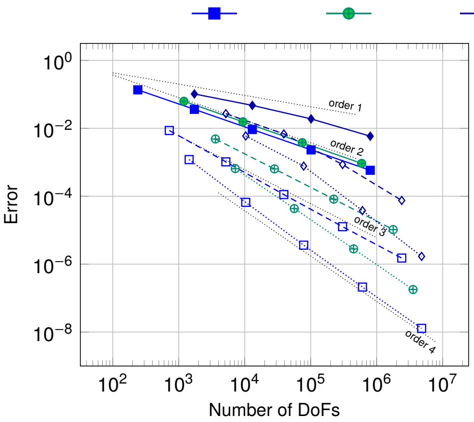
- 1.7 Billion polyhedral cell mesh
- Nearly optimal speedup up to 89% of the machine

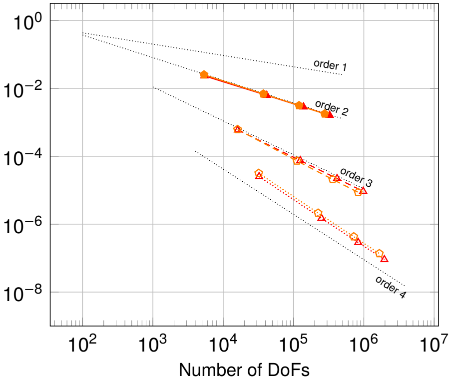
☺ Enhanced two-level MPI/OpenMP parallelism sucessfully tested on Intel Xeon Phi MIC (Many Integrated Core)

HHO schemes for anisotropic diffusion

FVCA6 benchmark testcase



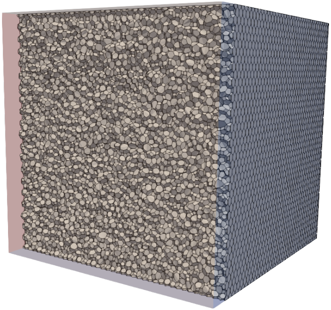




Mesh format support changes

Add support for Polyhedral (NFACE_n) CGNS meshes.

- Allows importing meshes produced by some recent ANSYS or STAR-CCM+ products.
- Backport: also available in 5.1.4 and 5.0.8 releases.
- Import only; CGNS output still subdivides these cells.



Other architectural changes since V5.0

- Improve `Catalyst` environment handling for **better automation with SALOME**.
- Default to PyQt5 instead of PyQt4 when both are available.
- Removed handling of VTK-based scalar slice outputs in studymanager; graphical files may now be included, and using `Catalyst` to produce images is recommended.
- Add **Melissa** (https://melissa-sa.github.io/) type writers for coupling with **in-situ statistics**.
- GUI enforces the 'runcase' name for the script, allowing a more robust handling of the runcase/XML file dependency relation.

Development version access

Migrated from Subversion to Git

- Internal repository migrated to the PAM GitLab forge.
  - previous Subversion repository now read-only, no updates (will be locked in the near future).
- developement tab on code-saturne.org now links to https://github.com/code-saturne/code_saturne public git mirror.
- Bugtracker still active for now, but recommend posting new issues on the GitHub 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).

Conclusion messages

Code_Saturne is NOT a black-box!
Ask what you want! Do what you want!

WebSite

BugTracker

Forum