LATEST NEWS and PROSPECTS in *Code_Saturne*

Development team presentation

April 9, 2013

SUMMARY

1. TOOLCHAIN
2. WEB SITE DEMONSTRATION
3. VERIFICATION AND VALIDATION OF CODE_SATURNE 3.0
4. NEW FEATURES
5. PROSPECTS

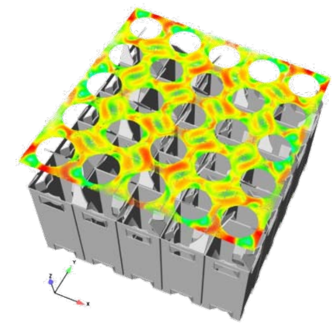
SIMULATE AND DECIDE

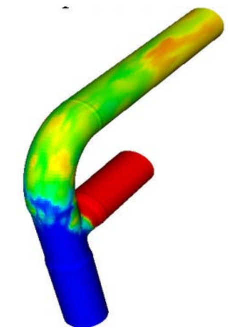
- **Power generation**
  - Improve the efficiency and safety of our facilities
  - Optimization of maintenance and life span
  - Response to specific events (flood, heat wave, incidents,…)

- **Preparation of the future**
  - New technologies for power generation
  - Innovation in renewable energies and storage
  - Anticipation of climate constraints on shared resources
- **Promotion of sustainable development**
  - Help customers optimize their energy consumption
- **Global Challenges**
  - Evaluation and reduction of environmental impact, waste management
- **Development of future simulation tools**

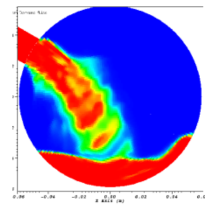
CODE DEVELOPMENT AT EDF R&D (1/2)

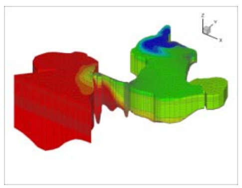
- *Code_Saturne*
  - general usage single phase CFD, plus specific physics
  - property of EDF, open source (GPL)
  - http://www.code-saturne.org



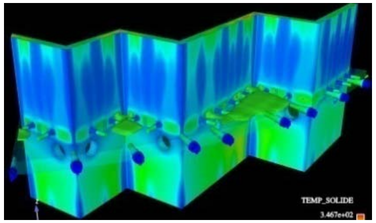


- **NEPTUNE_CFD**
  - multiphase CFD, esp. water/steam
  - property of EDF/CEA/AREVA/IRSN

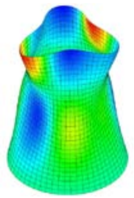




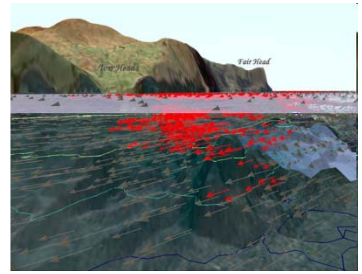
- **SYRTHES**
  - thermal diffusion in solid and radiative transfer
  - property of EDF, open source (GPL)
  - http://rd.edf.com/syrthes

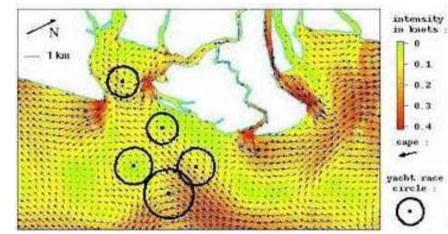


- ***Code_Aster***
  - general usage structure mechanics
  - property of EDF, open source (GPL)
  - http://www.code-aster.org

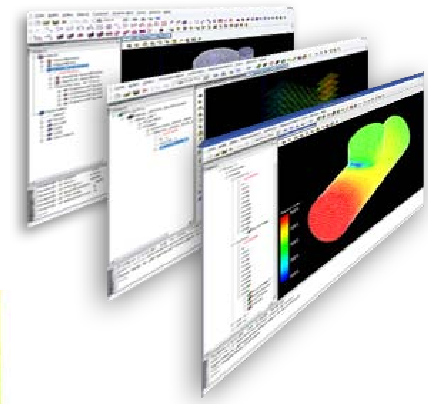


CODE DEVELOPMENT AT EDF R&D (2/2)

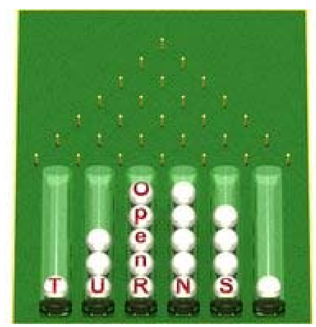




- **TELEMAC system**
  - free surface flows
  - Many partners, mostly open source (GPL, LGPL)
  - http://www.opentelemac.org



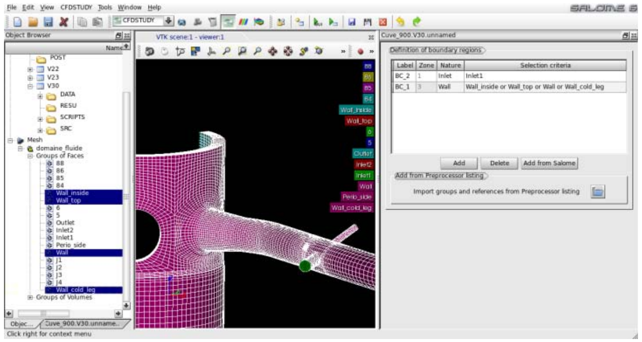
- **SALOME platform**
  - integration platform (CAD, meshing, post-processing, code coupling)
  - property of EDF/CEA/OpenCascade, open source (LGPL)
  - http://www.salome-platform.org

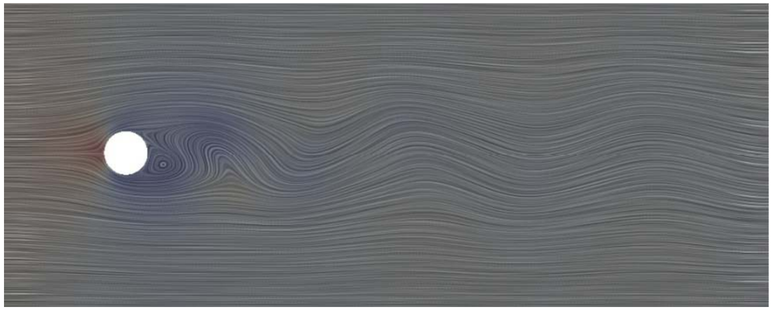


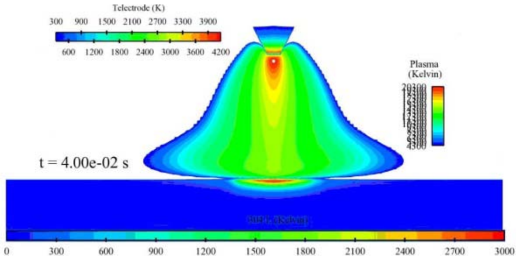
- **Open TURNS**
  - tool for uncertainty treatment and reliability analysis
  - property of EDF/CEA/Phimeca, open source (LGPL)
  - http://trac.openturns.org
- **and many others**
  - Neutronics, electromagnetism, acoustics
  - component codes, system codes

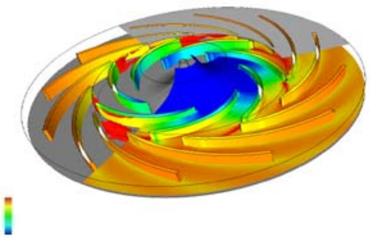
Code_Saturne
EDF'S GENERAL PURPOSE CFD TOOL

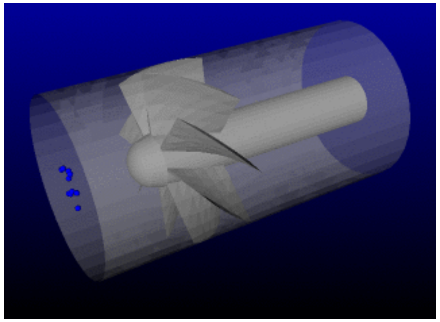
- **Co-located finite volume**
  - Arbitrary unstructured (polyhedral meshes)
- **Physical modeling**
  - Laminar and turbulent flows: k-ε, k-ω SST, v2f, RSM, LES
  - Radiative transfer (DOM, P-1)
  - Coal, heavy-fuel and gas combustion
  - Electric arcs and Joule effect
  - Lagrangian module for particles tracking
  - Atmospheric modeling
  - ALE method for deformable meshes
  - Rotor / stator interaction for pumps modeling
  - Conjugate heat transfer (SYRTHES & 1D)





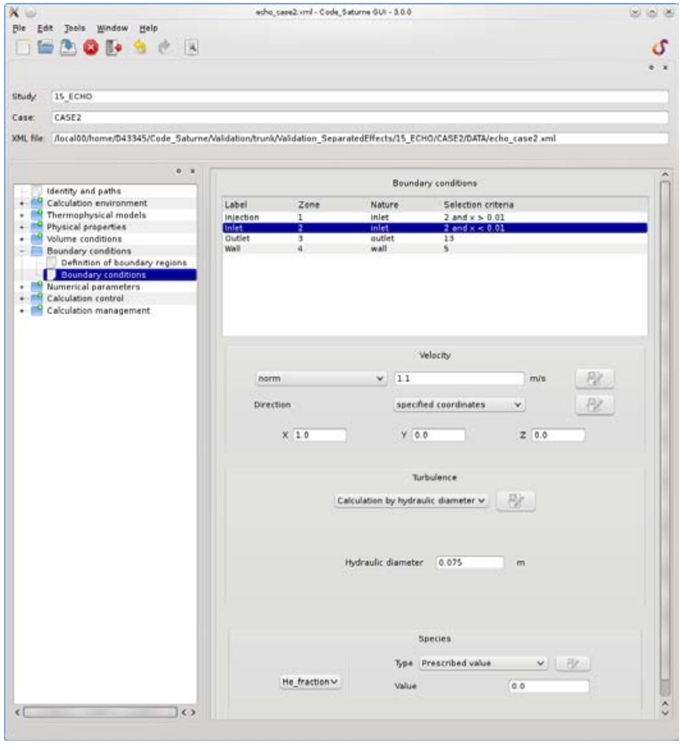


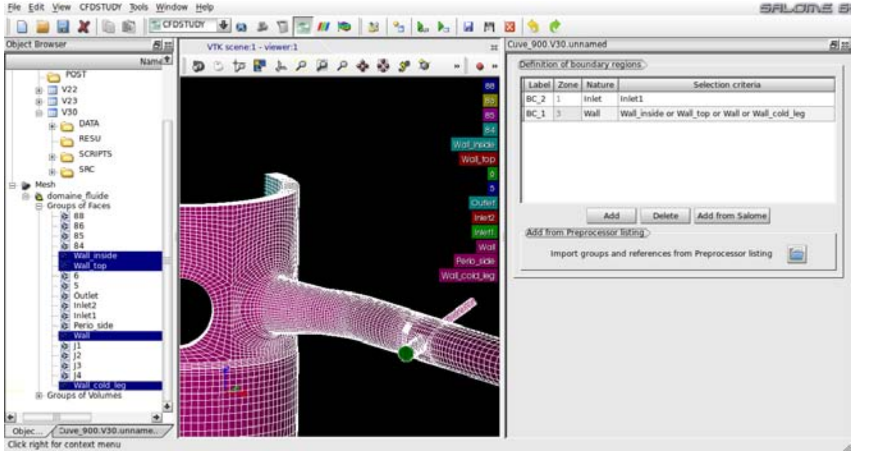




REQUIRED ENVIRONMENT

- **Runs on all Unix-type systems**
  - Linux, Unix, Mac OS-X
  - Windows port underway
- **Pre-requisites**
  - Compilers and interpreters
    - C 99 and Fortran 95
    - Python (2.6 and above)
  - for parallel computing
    - MPI library : Open MPI, MPICH, ...
    - METIS or SCOTCH partitioner (optional)
  - for GUI
    - Qt4, PyQt, SIP, libxml2
  - Optional data exchange libraries
    - MED, CGNS, libCCMIO
- **Optional SALOME platform integration**
  - GUI extensions
    - mouse selection of boundary zones
    - advanced user files management
  - from CAD to post-processing in one tool



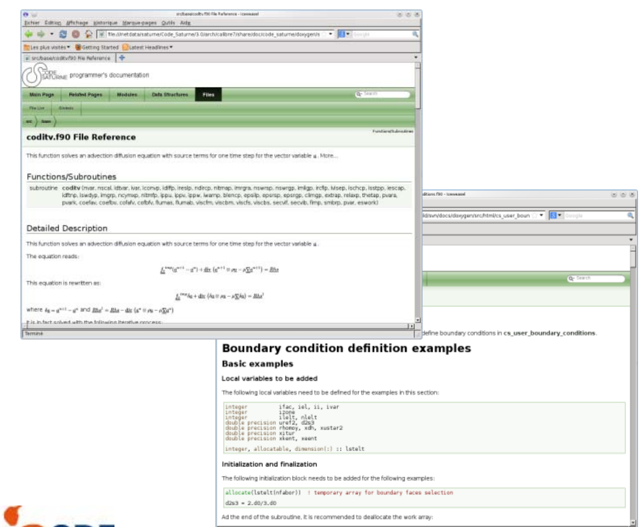


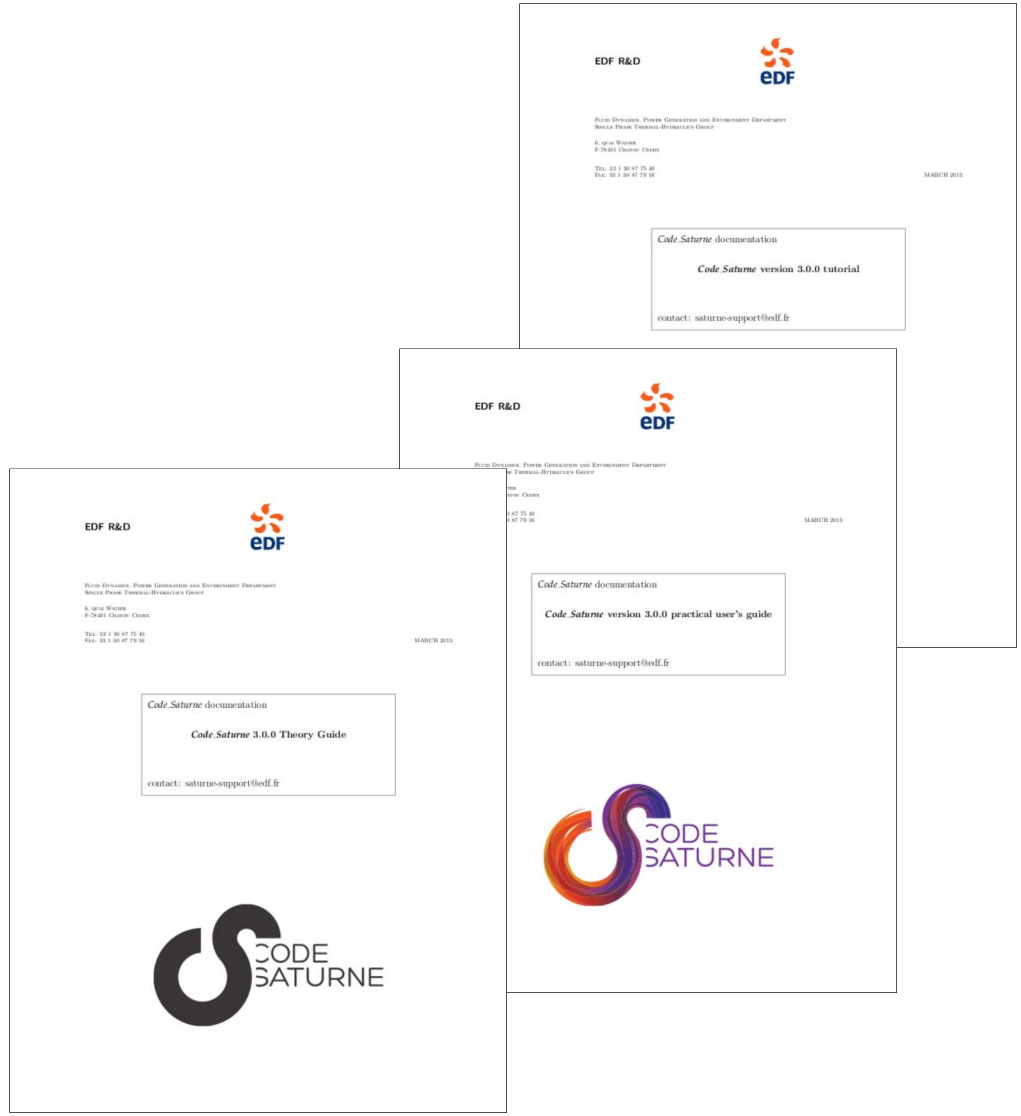
BUILD SYSTEM

- **Based on the GNU Autotools**
  - Autoconf, automake, libtool, gettext
  - Fortran 90 module dependencies requiremanual additions in Makefiles
- **Build system records parameters of the code's Python-based scripts**
  - Allow compilation of user functions (in C or Fortran)
- **1 single package**
  - PLE "Parallel location and exchange" sub-package
- **Multiple install options**
  - Debian, Mageia, and other Linux distribution packages
    - Easiest, but may not be up to date
  - Automatic installer
    - Python scripts, installs some prerequisites
  - Manual install
    - Detailed documentation available
      - Especially oriented towards installs on clusters

DOCUMENTATION

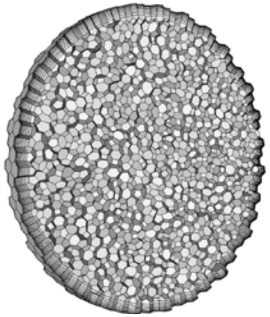
- **PDF documentation**
  - User manual, Tutorial
  - Theory manual
  - Developer's guide
  - Installation manual
  - V&V documentation
- **Inline programmer's documentation**
  - Doxygen-based

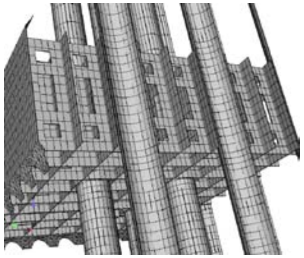




SUPPORTED MESHES

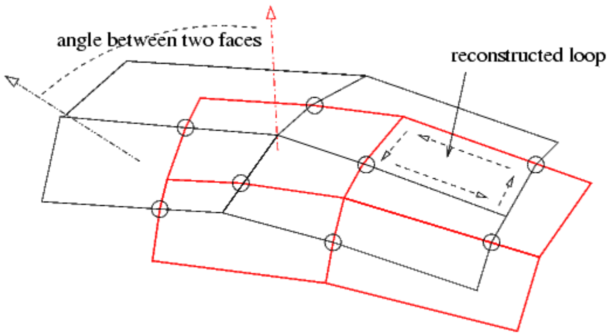
- **Mesh generators**
  - Simail: easy-to-use, with command file, but no CAD
  - I-deas NX
  - GAMBIT (Fluent), ICEM-CFD, ANSYS meshing
  - Star-CCM+
  - SALOME SMESH (http://www.salome-platform.org)
  - Gmsh
  - Harpoon, ...
- **Formats**
  - **MED**, **CGNS**, Star-CCM+, Simail, I-deas universal,
  - GAMBIT neutral, EnSight Gold…
- **Cells: arbitrary arrangement of polyhedra**
  - For example: tetrahedra, hexahedra, prisms, pyramids, general n-faced polyhedra, ...



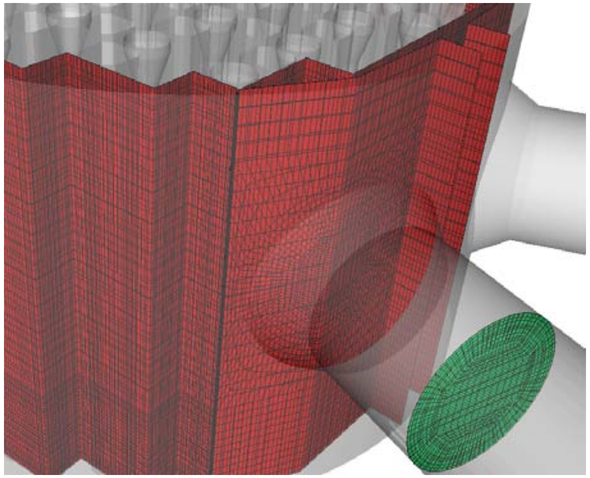


MESH PREPROCESSING

- **Read and append meshes**
  - Allows mixing supported formats, changing coordinates, and group names upon reading
- **Conforming or non-conforming mesh joining**
  - Expertise may be required if arbitrary interfaces are used:
    - In critical regions
    - With LES
    - With very different mesh refinements
    - On curved CAD interfaces
  - Now done in parallel

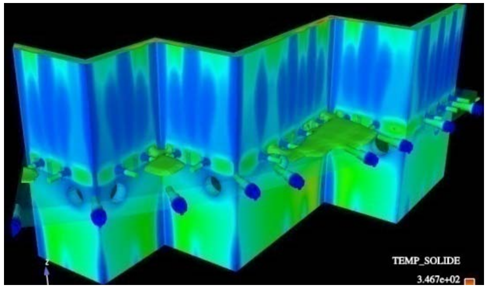


- **Other preprocessing options now available**
  - Mesh coordinates transformation
  - Mesh smoothing
    - Currently limited to de-warping
  - Insert thin walls (baffles)

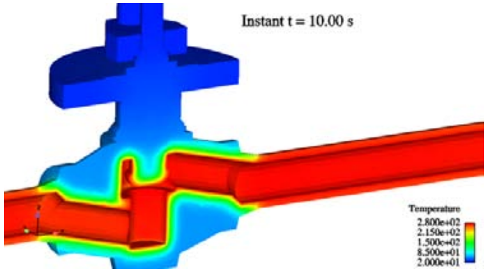


PARALLEL CODE COUPLING

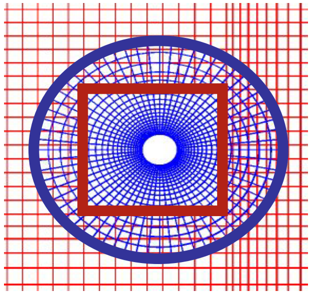
- **Parallel n to p coupling using “Parallel Location and Exchange” sub-library**
  - Uses MPI
  - Fully distributed (no master node requiring global data)
  - Successor to / refactoring of FVM (also used in CWIPI)
    - Core communication in PLE, rest moved to code

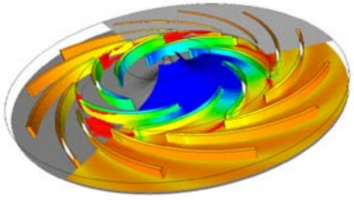


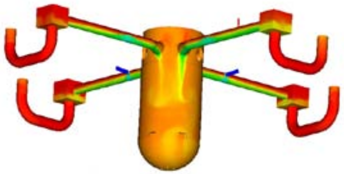
- **SYRTHES (conjugate heat transfer)**
  - Coupling with (parallel) SYRTHES 4 run on industrial study using 128+ processors



- ***Code_Saturne* / *Code_Saturne* coupling**
  - RANS / LES
  - Different turbulence models and time step, fixed overlapping domains
  - Turbomachinery
    - Same turbulence model and time step, moving sub domain



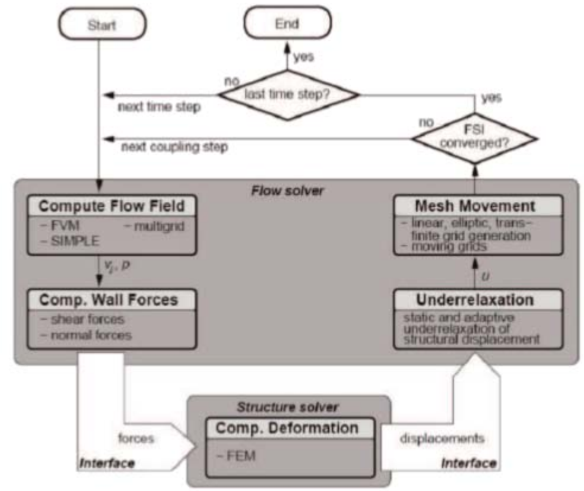


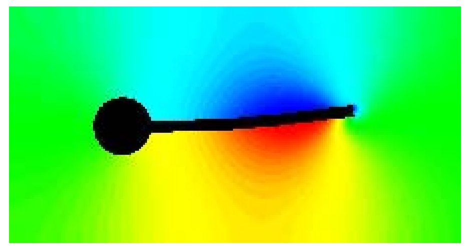


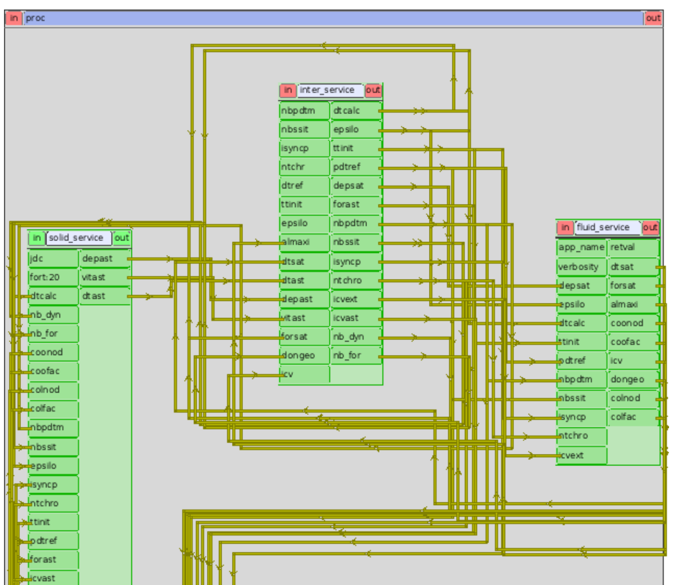
- **Others**
  - MedCoupling support

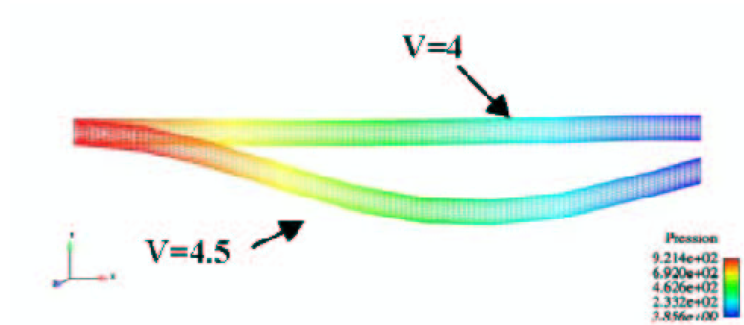
OTHER CODE COUPLINGS

- **Coupling with *Code_Aster***
  - Based on the SALOME kernel
    - Uses YACS for running
    - Uses the CALCIUM API
      - From the SALOME kernel, not the old CALCIUM tool
  - Partially working, still seems to require some minor fixes





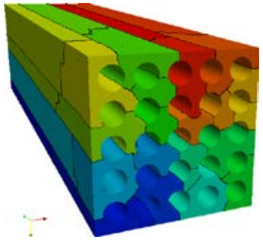


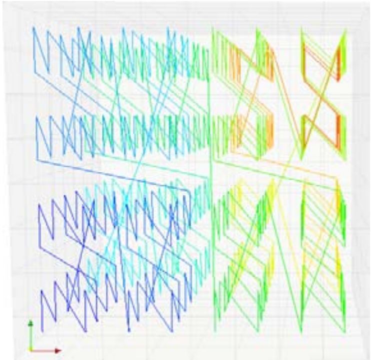


PARALLEL PARTITIONING

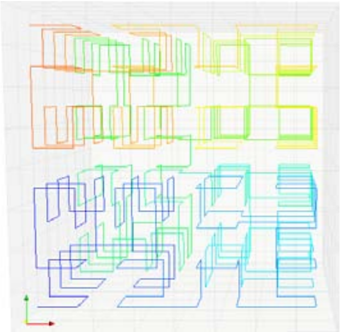
- **Multiple partitioning options are possible**
  - Serial Scotch or METIS
  - Parallel PT-Scotch or ParMETIS
    - Possibly run on a subset of the active ranks (allows for quality/memory optimization checks)
- **Morton or Hilbert space-filling curve (in bounding box or bounding cuve)**
  - Built-in, requires no external library
  - Safe to have this as a back-up (and reference)
  - Advantage: deterministic

PT-Scotch (32 ranks)



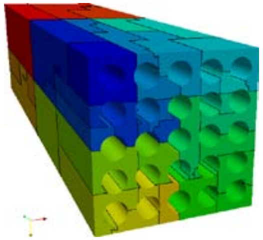


Morton curve



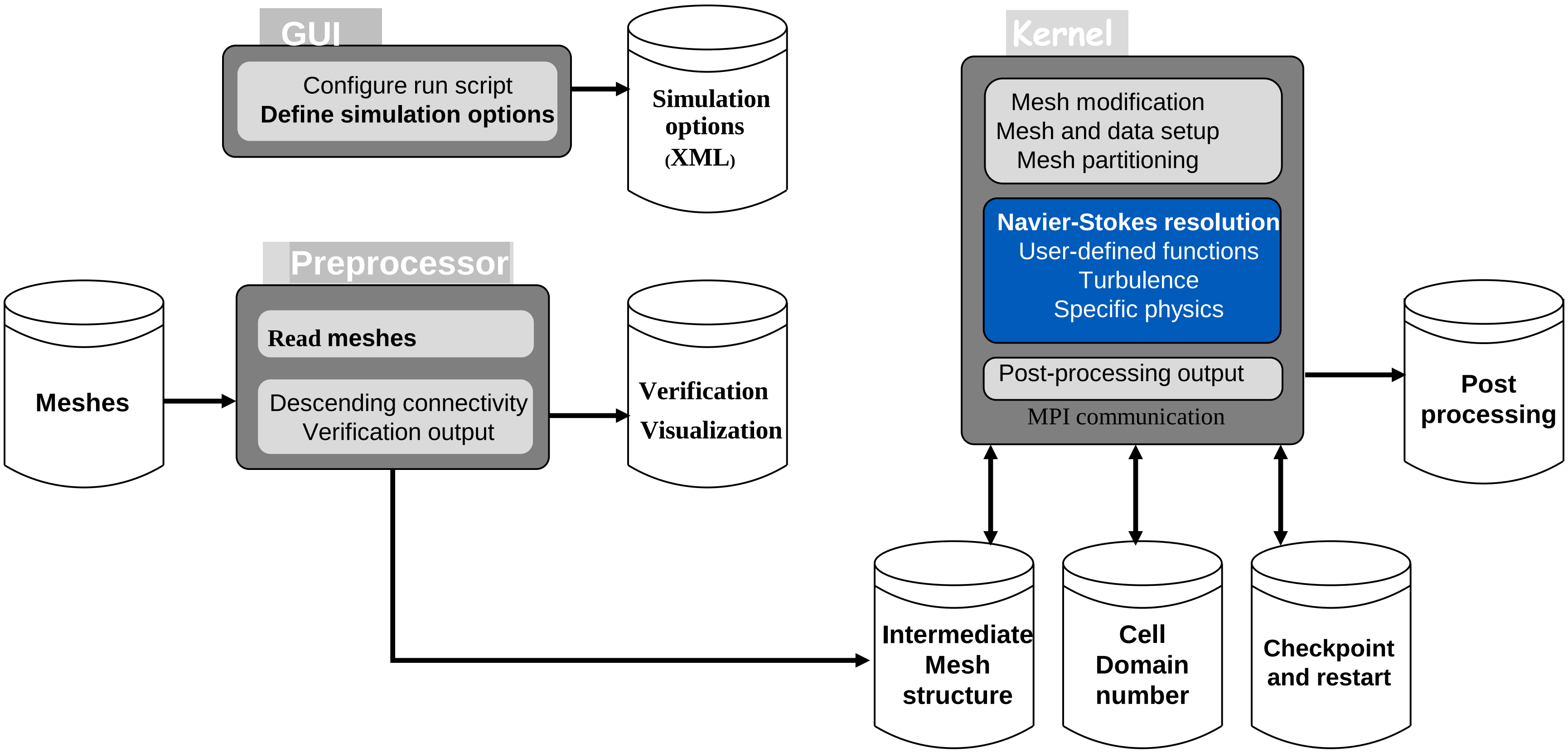
Hilbert curve

Hilbert, bounding box (32 ranks)



TOOLCHAIN

- **Reduced number of tools**
  - Each with rich functionality
  - Natural separation between interactive and potentially long-running parts



RUNNING *Code_saturne*

- **Single `code_saturne` command**
  - now extends to running
    - `code_saturne run`
    - `code_saturne autovnv`
- **Only requires path, all other environment aspects encapsulated**
  - Allows side-by side installation of multiple versions
    - Potentially with aliases (such as `cs20=<prefix_20>/bin/code_saturne`, `cs20=<prefix_20>/bin/code_saturne`)
- **Other extensions**
  - `code_saturne io_dump` now has `–diff` option
- **Python run script allows for minimal `runcase`**
  - Can exploit options from XML setup file
  - Call points for user script python functions
    - Logic consistent with user subroutines, with user functions compatible with and taking priority over the GUI-defined XML file
- **Many user subroutines renamed and reorganized for clarity**

And now, the details…

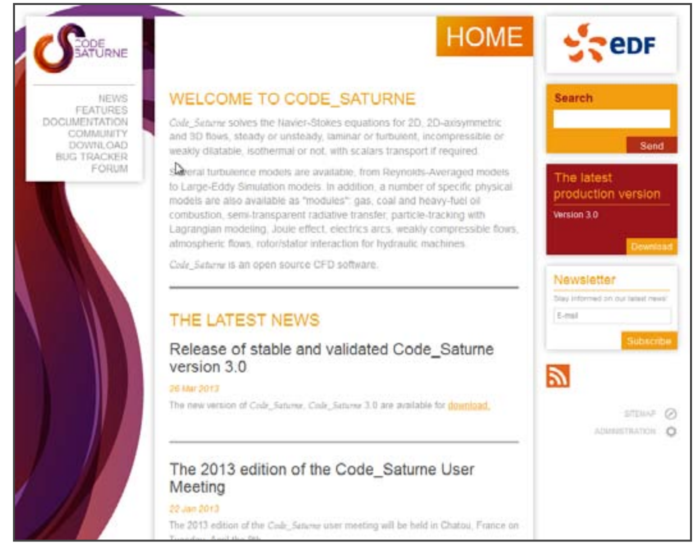
Code_Saturne Website demo

Code_Saturne User Meeting

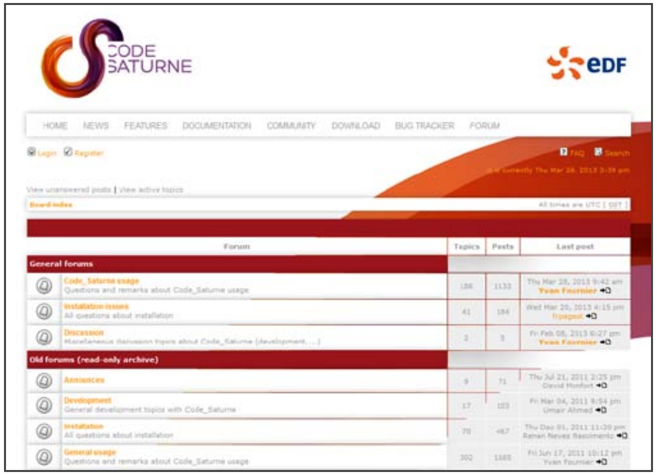
April 9, 2013

Code_Saturne website

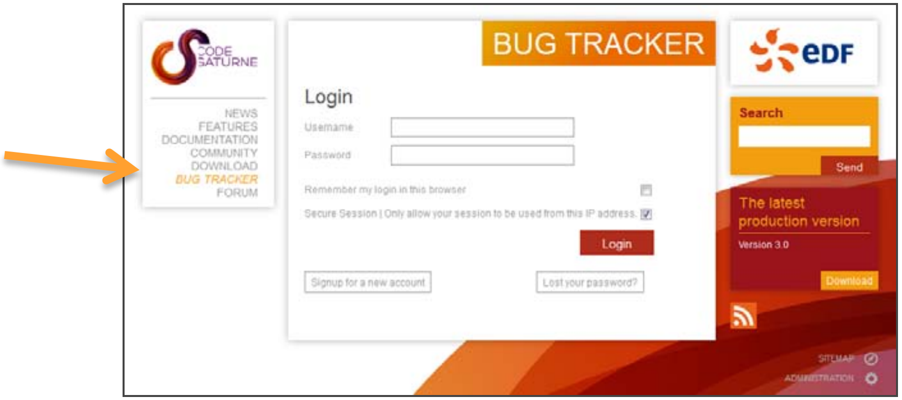
- **Questions you might ask yourself**
  - Where can I find this new website?
    http://code-saturne.org
  - Where can I find the latest production version?
    - On every page or in the « DOWNLOAD » section
  - Where can I find the documentation?
    - In the « DOCUMENTATION » section

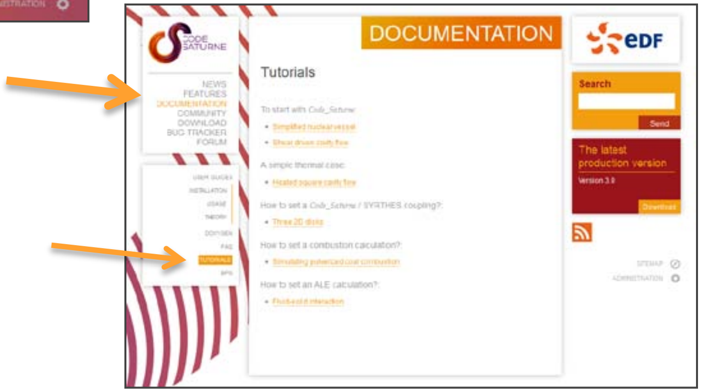


Code_Saturne website



- What should I do in case of difficulties?
  - Forum
  - Best practices guides lines
  - Tutorials
  - Bug tracker





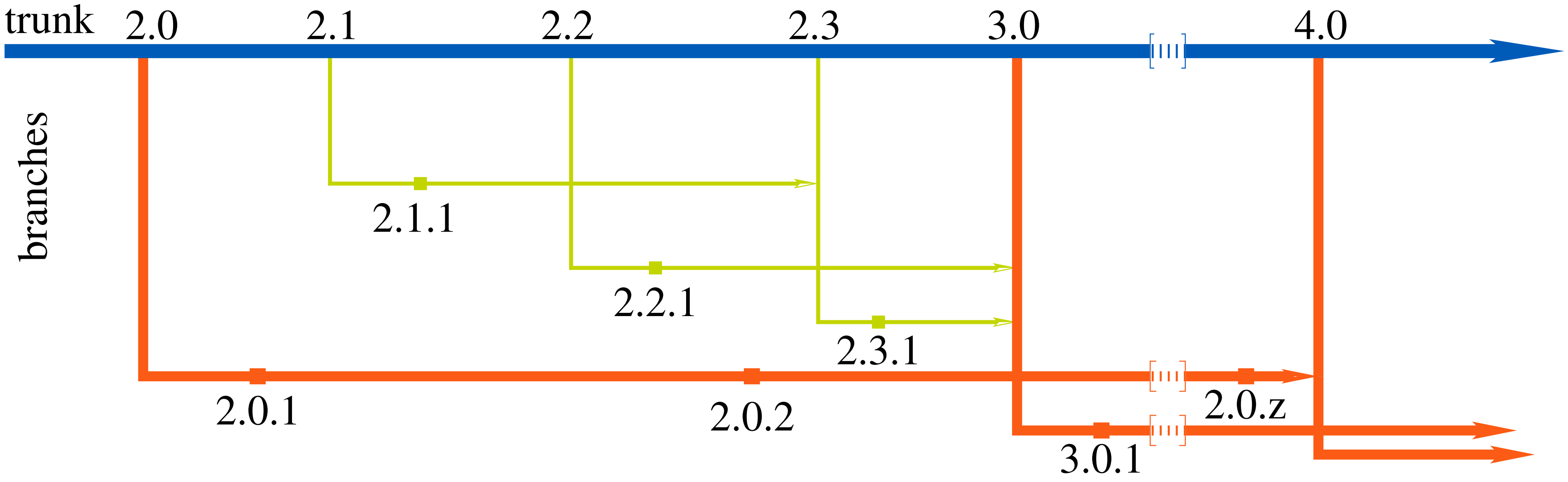
A few words about V&V of *Code_Saturne* 3.0 and the AutoVnV tool

Code_Saturne user meeting

April 9, 2013

Code_Saturne versioning

- From version 2.0 on, different kinds of versions "*x.y.z*" are released
  - Production version every two years (x increasing)
    With the release of a ***Verification & Validation*** summary report
  - An intermediate version every six months (y increasing)
    With non-regression tests to ensure the code quality
  - Corrective versions when needed (z increasing)
    To make sure the users are provided with the fixes in time



Code_Saturne 3.0 now available!

Code_Saturne version 3.0 has been released at the end of March 2013!

- *Code_Saturne* is a production version
- *Verification & Validation* summary report finalised under code quality assurance
- *Code_Saturne* package is available on the website: *www.code-saturne.org*

Verification & Validation (V&V) in *Code_Saturne*

- Different concepts exist to determine the quality of a release
  - Non-regression (on a nightly or weekly basis)
  - Verification & Validation (for each major release, at least)

- According to Oberkampf et al.
  - *Measures of agreement between computation an experiment: Validation metrics, Journal of Computational Physics, 217 (2006)*
  - *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations, AIAA (1998)*

  - Verification
    - "the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model"

  - Validation
    - "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model"

Verification & Validation (V&V) in *Code_Saturne*

- Verification and Validation of *Code_Saturne* 3.0 with 30 test cases
  - 10 test cases on academic configurations
  - 10 test cases on semi-industrial or industrial configurations
  - 10 test cases with specific physics and/or advanced modelling

- Around 5 computations per test case
  - To verify and validate numerical options
  - To compare models (particularly turbulence models)
  - To ensure non-regression

- About 150 calculations for the whole V&V process

Due to the long *Verification & Validation* process (about 1 man year):

An automatic framework to handle V&V stage of *Code_Saturne* 3.0 has been developed

Code_Saturne AutoVnV tool

- Features
  - Setup *Code_Saturne* studies by copying a reference study from a given database
  - Three different modes:
    - Run all simulations and return an error code whether the calculation ran smoothly or not
    - Post process results (cross plotting, slice extraction with pyVTK) with possible reference data
    - Compare results from previous calculation
  - Send several reports (short, detailed, …) with the results, cross plots, slice extraction, results from user scripts and computational time, … to given e-mail addresses

- Basic functions for automatic run of simulations
  - No specific installation is required
  - Setup with an xml file
  - Possibility to do specific operations with user Python scripts
  - Use the `matplotlib` and `pyVTK` python modules for post processing
  - Simple use with `code_saturne autovnv` commands

Basic AutoVnV commands

```
code_saturne autovnv -f <file> [options]
```

✓ **Help**	`code_saturne autovnv --help`
✓ **Update scripts**	`code_saturne autovnv -f <file> -u`
✓ **Run the autovnv xml**	`code_saturne autovnv -f <file> -r`
✓ **Post process**	`code_saturne autovnv -f <file> -p`
✓ **Run & Post process**	`code_saturne autovnv -f <file> -rp`
✓ **Compare with previous calculation**	`code_saturne autovnv -f <file> -c`

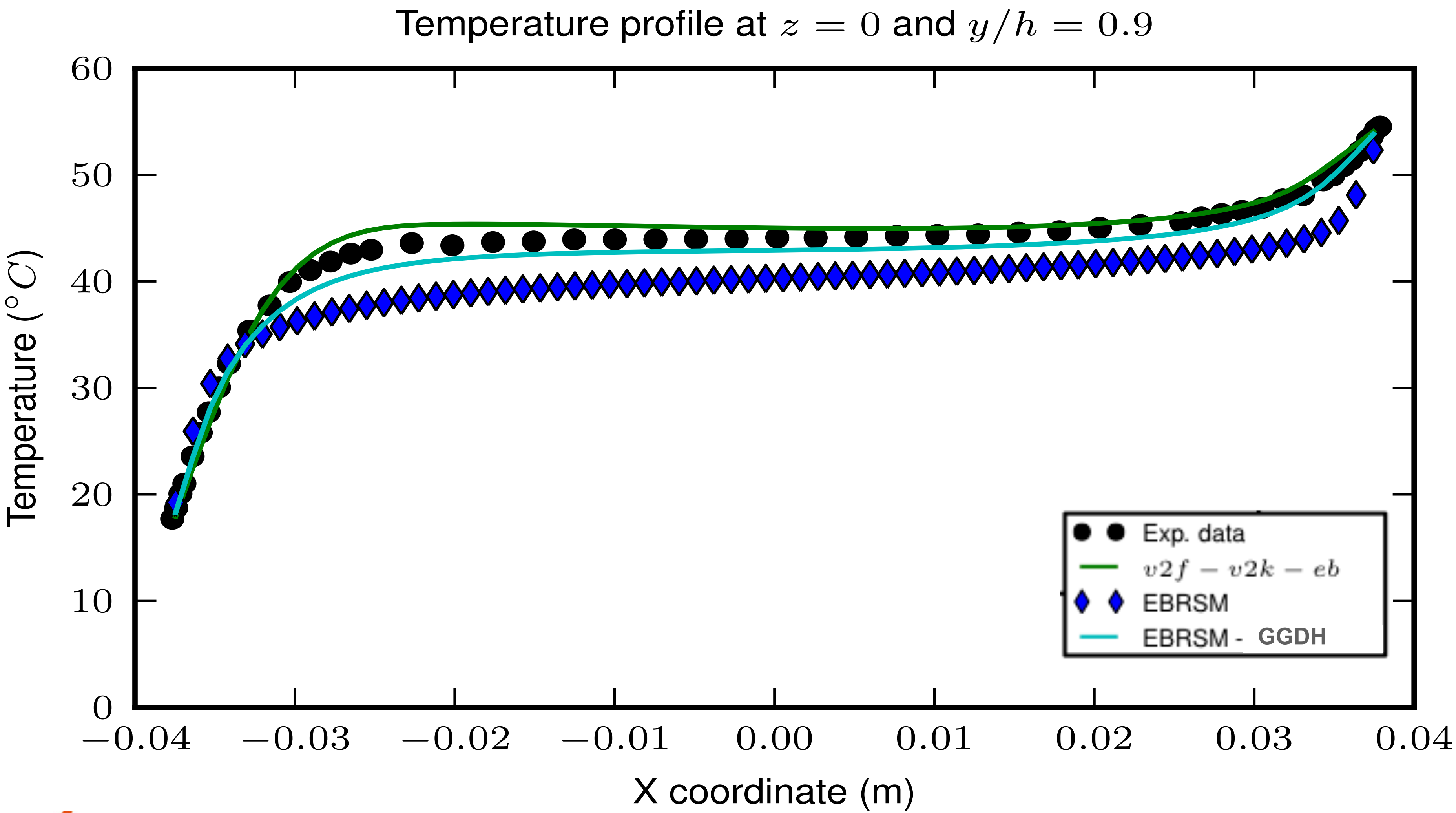
- See the documentation

```
code_saturne info --guide=autovnv
```

AutoVnV example: plot 1D profile

```
<study label=”BETTS_CAVITY" status="on">

   <!--
       ***************************************
       *  Experimental data                  *
       ***************************************
       Data from Betts and Bokhari's experiment, IJHFF, 21, 675-683, 2000.
   -->

   <measurement file="mt_z0_90_hi.dat" path="">
     <plot fig="56" xcol="1" ycol="2"  xfois="0.001" xplus="-0.038" legend="Exp. data"/>
   </measurement>

  <!--
       ***************************************
       *  Case                               *
       ***************************************
   -->

   <case compute="on" label="CASE1" post="on" status="on">
     <probes dest="" fig="101" file="probes_TempC.dat"/>
     <data dest="" file="profil90.dat">
       <plot fig="54" fmt="r" legend="$Legend$" xcol="2" ycol="5"> </plot>
     </data>
   </case>

...
```

AutoVnV example: plot 1D profile (extracted from the automatic report)



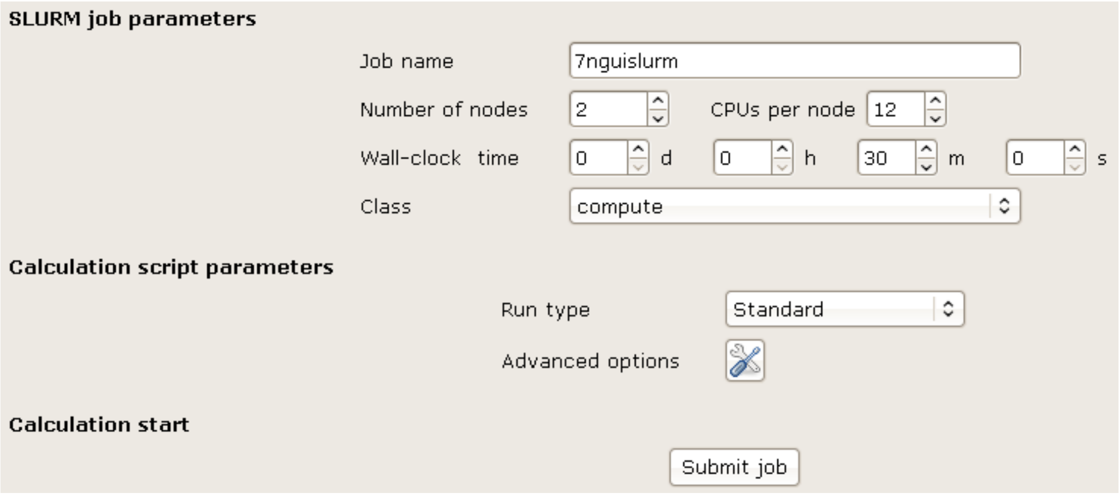
EDF

Improving user experience with *Code_Saturne* 3.0

- Through the installation
  - A single installation package
    - configure, make and make install commands
  - A Debian package (and derivatives)
    - Available in Debian unstable, soon in testing
    - With main features enabled (with graphical interface, parallelism, CGNS and MED support, ...)
  - An installation documentation (basic and advanced installations)
    - Available on the website (www.code-saturne.org/cms/documentation)
    - Available in the *Code_Saturne* package

- Through the GUI
  - About "90% of functionalities" of *Code_Saturne* 3.0 are available trough the GUI

  Major improvements in:
    - Specific physics
    - Advanced post-processing tools and options
  - Job submission for batch systems



NEW FEATURES IN *Code_Saturne* V3.0

Code_Saturne development team

April 9, 2013

Overview

1. General user changes
   - User subroutines
   - runcase
   - Changes in default options
2. New features
   - Architecture
   - Numerics
   - Physical modelling
   - Turbulence modelling
3. Appendix
   - Correction on the Temperature equation
   - Porosity module
   - Mass conservation in the weakly compressible Algorithm
   - Hydrostatic pressure computation for stratified flows

Overview

1 General user changes
- User subroutines
- runcase
- Changes in default options

2 New features
- Architecture
- Numerics
- Physical modelling
- Turbulence modelling

3 Appendix
- Correction on the Temperature equation
- Porosity module
- Mass conservation in the weakly compressible Algorithm
- Hydrostatic pressure computation for stratified flows

General user changes

See release note
(http://code-saturne.org/cms/download/3.0)

Changes in the user subroutines

Mainly merging and renaming to be maintainable.

SRC/REFERENCE/cs_user_parameters.f90

previously: `usini1.f90` + dedicated user routines for specific physics

SRC/REFERENCE/cs_user_boundary_conditions.f90

previously: `usclim.f90` + dedicated user routines for specific physics

SRC/REFERENCE/cs_user_extra_operations.f90

previously: `usproj.f90` + dedicated user routines for specific physics

SRC/REFERENCE/cs_user_modules.f90

previously: `rtuser`, `rdevel` arrays, ...

SRC/REFERENCE/cs_user_physical_properties.f90

previously: `usphyv.f90` + dedicated user routines for specific physics

Changes in the user subroutines

SRC/REFERENCE/cs_user_initialization.f90

previously: `usiniv.f90` + dedicated user routines for specific physics

you can call here the initialization of user arrays (see `cs_user_modules.f90`)

SRC/REFERENCE/cs_user_mesh.c

previously in: `runcase` ...

you can duplicate, move or stretch your mesh...

SRC/REFERENCE/cs_user_postprocess.c

previously in: `usini1.f90` and `usdpst.f90` et `usmpst.f90` + dedicated user routines for specific physics

Changes in the user subroutines

SRC/REFERENCE/ cs_user_postprocess_var.f90

previously: `usvpst.f90` + dedicated user routines for specific physics

SRC/REFERENCE/ cs_user_source_terms.f90

previously: `ustsns.f90` (`ustsnv.f90`, `usttke.f90`, ...)

New examples (not exhaustive) in SRC/EXAMPLES

`cs_user_boundary_conditions-advanced.f90`,
`cs_user_extra_operations-energy_balance.f90`,
`cs_user_extra_operations-extract_1d_profile.f90`,...
the REFERENCE routines are therefore empty.

Changes in the runcase: everything

`runcase` is almost empty

If you do NOT use the GUI:

- mesh selection is in `DATA/cs_user_scripts.py`
- writers are in `SRC/cs_user_postprocess.c`
- mesh modifications are in `SRC/cs_user_mesh.c`
- The SCRATCH directory is now optional, and cleaned is the calculation finish normally
- Every single file in the DATA directory is now copied automatically (no need to select it in the GUI or in the runcase)

Changes in the listing

Separated listings

- `performance.log`
- `preprocessor.log`
- `setup.log`
- `listing`

Change in the listing

`Norm. residual` was an information on the last linear solver resolution, it is now on the convergence of global equation to be solved

```
---------------------------------------------------------------
   Variable    Rhs norm      N_iter  Norm. residual      derive
---------------------------------------------------------------
c  Pressure     0.32339E-01     301    0.94936E-06   0.23768E+00
c  VelocityX    0.16243E+02       8    0.12248E-03   0.32377E-06
c  VelocityY    0.16243E+02       8    0.12248E-03   0.21420E-38
c  VelocityZ    0.16243E+02       8    0.12248E-03   0.13603E-06
c  mesh_u       0.36729E-02      67    0.28807E-04   0.00000E+00
c  mesh_v       0.36729E-02      67    0.28807E-04   0.00000E+00
c  mesh_w       0.36729E-02      67    0.28807E-04   0.12883E-06
---------------------------------------------------------------
```

New default options

Coupling of the velocity components

- `ivelco=1`, go back to the previously version with `ivelco=0`

Default turbulence model

- $k-\epsilon$ Linear production (`iturb=21`)

Relaxation

- Relaxation on $k-\omega$ switched off (`relaxv(iomg)=1`)
- Relaxation in steady algorithm is set to 0.7 (in accordance to the CFD community)

Default symmetry boundary condition on R_{ij}

- Improved symmetry BCs for the tensor $\underline{\underline{R}}$ (`iclsyr=1`)

Overview

1 General user changes
- User subroutines
- runcase
- Changes in default options

2 New features
- Architecture
- Numerics
- Physical modelling
- Turbulence modelling

3 Appendix
- Correction on the Temperature equation
- Porosity module
- Mass conservation in the weakly compressible Algorithm
- Hydrostatic pressure computation for stratified flows

New programming features

Architecture

- Fortran 95 (dynamic memory allocation, modules)
- User scripts
- Unification of install packages
- Factorization and rewriting to make the kernel of the code clearer
- Field structure with Fortan API (to facilitate the addition of new variables and the future translation in C)

New programming features

High Performance Computing (HPC)

- Mesh joining and parallel partitioning done by the kernel (therefore in parallel)
- Parallelising of the Lagrangian module for particle tracking
- Hybrid parallelism MPI/OpenMP

Many functionalities added or improved in the GUI

- Compressible model, electric arcs, and other advanced modelling management
- Improvement of pre-processing management
- Linear algebra, numerical options
- Performance tuning, batch system calculation management

New numerical features

Generalities

- Mesh tools (unwarp smoother, thin walls)
- Vector coupled components solver (fluid velocity, mesh velocity in ALE) (`ivelco=1`)
- New internal formulation for boundary conditions (split between the gradient and for the diffusive flux BCs)
- Radiative outlet (Orlanski) (`icodcl(ivar, ifac)=2`)
- Free surface BC for ALE (`altycl(ifac)=ifresf`)
- Rewriting the temperature equation (to be consistent with the energy equation when C_p is variable)
- Algebra:
  - Vectorial multi-grid algorithm (mesh velocity in ALE)
  - GMRES linear solver
- Add an anisotropic diffusion operator on scalars and vectors

A word about anisotropic diffusion

Formulation

- $\text{div}\left(\underline{\underline{K}}\,\underline{\nabla}\,T\right)$ for a scalar field T
- $\underline{\text{div}}\left(\underline{\underline{K}}\,\underline{\underline{\nabla}}\,\underline{u}\right)$ for a vector field $\underline{u}$

$\text{div}\left(\underline{\underline{K}}\,\underline{\nabla}\,T\right)$ used in...

- the GGDH on scalar transport equations
- on the pressure solving with head losses or reinforced velocity pressure coupling (ipucou=1)

$\underline{\text{div}}\left(\underline{\underline{K}}\,\underline{\underline{\nabla}}\,\underline{u}\right)$ used in...

- the GGDH on scalar turbulent flux $\overline{T'\underline{u}'}$ transport equation
- ...

New numerical features

Improve robustness

- Improve the time-stepping of the EVM (in particular $k - \varepsilon$ LP (`iturb=21`) and $k - \omega$ (`iturb=60`) with buoyancy)
- Automatic relaxation on the pressure solving (`iswdyn(ipr)=1 or 2`): ensure the convergence by increasing the number of reconstruction sweeps with an optimised relaxation

Coupling

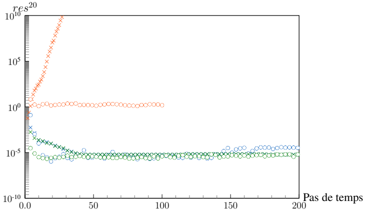
- SYRTHES 4 (parallel version) and only SYRTHES 4
- SALOME module: FSI_coupling with *Code_Aster* 11.0 (Work in Progress)

A word about on the automatic relaxation algorithm

Handle non-orthogonalities

$$\left\{ \begin{aligned} \mathbf{EM}\left(\delta \mathbf{P}^{k+1}\right) &= \mathbf{E}\left(\mathbf{P}^{k}\right) \\ \mathbf{P}^{k+1} &= \mathbf{P}^{k} + \alpha^{k+1}\delta \mathbf{P}^{k+1} \end{aligned} \right.$$

- $\left(\mathbf{P}^k\right)_{k\in\mathbb{N}}$: unknown vectors, **EM**: matrix solved at each iteration k
- α^{k+1}: optimum relaxation factor (minimize the residual)



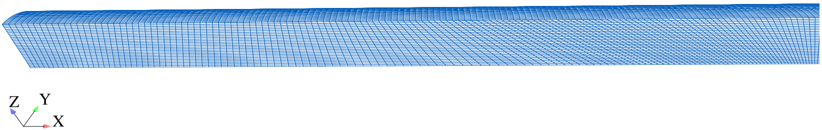
× R.H.S norm

○ Norm. residual

No relaxation

`iswdyn(ipr)=1`

`iswdyn(ipr)=2`



New physical models

Specific physics

- New Lagrangian model (particle deposition)
- Unified model for fuel and coal combustion in gas phase
- Low Mach number algorithm (possibility of Multi species model)
- Dilatation algorithm for fire
- Atmospheric models (humid atmosphere, soil model, solar radiation)
- Porosity for incompressible flows
- New improved hydrostatic pressure for stratified flows (`iphydr=2`)

New physical models

Specific physics

- New Lagrangian model (particle deposition)
- Unified model for fuel and coal combustion in gas phase
- Low Mach number algorithm (possibility of Multi species model)
- Dilatation algorithm for fire
- Atmospheric models (humid atmosphere, soil model, solar radiation)
- Porosity for incompressible flows
- New improved hydrostatic pressure for stratified flows (`iphydr=2`)

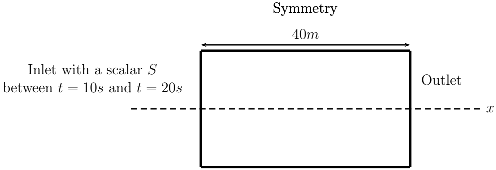
Utilisation of the dilatable and Low Mach algorithms

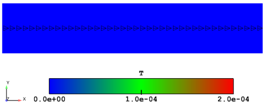
Settings using Fortran user subroutine

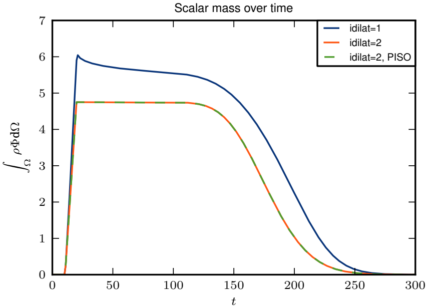
- Set key word `idilat` to 2 in `cs_user_parameters.f90` for the dilatable algorithm
- Set key word `idilat` to 3 in `cs_user_parameters.f90` for the Low Mach number algorithm in confined flows (then $P_{thermodynamics}$ is added as a source term in the enthalpy equation)

Conservation over time with the dilatable algorithm

[idilat]: Basic 1D slug test case

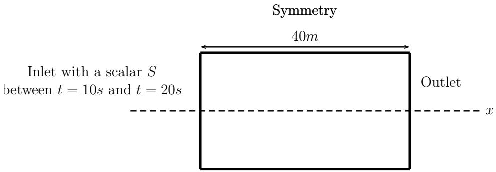


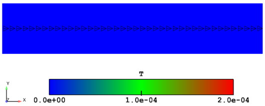


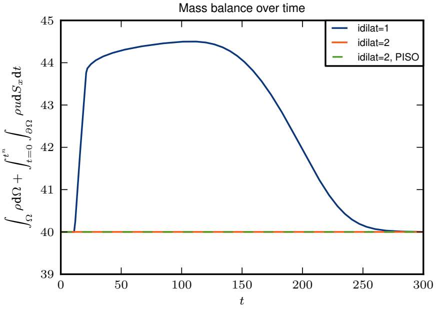


Conservation over time with the dilatable algorithm

[idilat]: Basic $1D$ slug test case

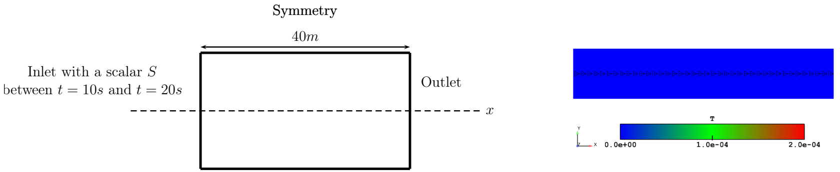


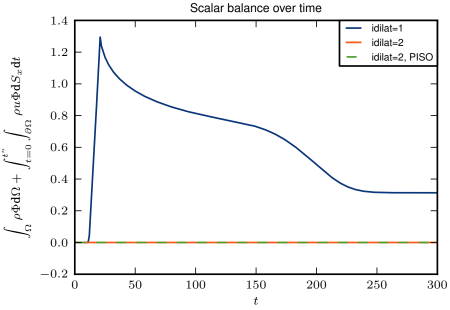




Conservation over time with the dilatable algorithm

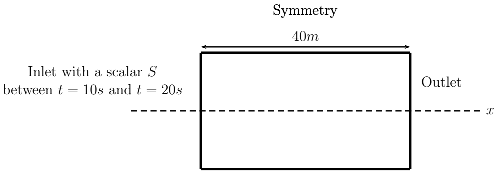
[idilat]: Basic $1D$ slug test case

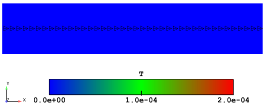


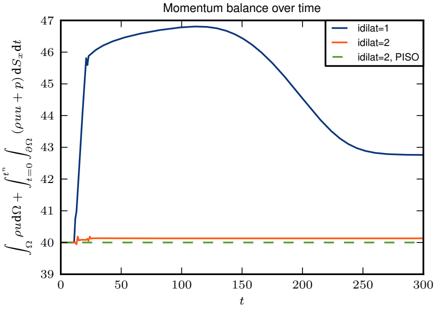


Conservation over time with the dilatable algorithm

[idilat]: Basic 1D slug test case







New physical models

Specific physics

- New Lagrangian model (particle deposition)
- Unified model for fuel and coal combustion in gas phase
- Low Mach number algorithm (possibility of Multi species model)
- Dilatation algorithm for fire
- Atmospheric models (humid atmosphere, soil model, solar radiation)
- Porosity for incompressible flows
- New improved hydrostatic pressure for stratified flows (`iphydr=2`)

Porosity utilisation

Some comments on the formulation

- modification of the mass flux at the cell faces
- modification of the diffusivity at the cell faces
- pressure gradient, sources terms and non stationary terms are multiplied by the porosity

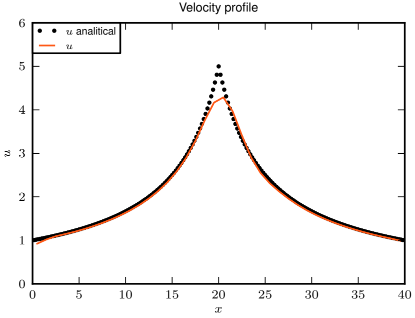
Porosity module available only using Fortran user subroutine

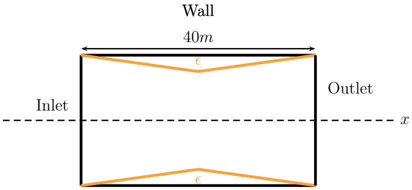
- Set key word `iporos` to 1 in `cs_user_parameters.f90`
- Set $\epsilon_i \in]0, 1]$ for all $i \in N_{cell}$ in `usporo.f90`

Analytical test case for the porosity module

[iporos]: Analytical problem

- Laminar flow $\nu = \frac{1}{12} m^2.s^{-1}$, $u^{inlet} = 1 m.s^{-1}$,
- ϵ profile is linear by part.





New physical models

Specific physics

- New Lagrangian model (particle deposition)
- Unified model for fuel and coal combustion in gas phase
- Low Mach number algorithm (possibility of Multi species model)
- Dilatation algorithm for fire
- Atmospheric models (humid atmosphere, soil model, solar radiation)
- Porosity for incompressible flows
- New improved hydrostatic pressure for stratified flows (`iphydr=2`)

Utilisation of the hydrostatic pressure for stratified flows

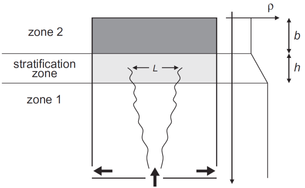
Settings using Fortran user subroutine

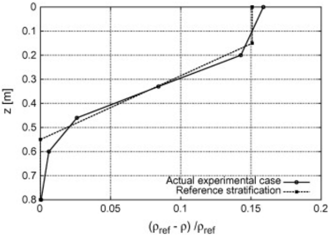
- the key word `iphydr` is equal to 0 by default in the sources, the hydrostatic pressure is not computed,
- Set key word `iphydr` to 1 in `cs_user_parameters.f90` to compute a hydrostatic pressure which handles the balance between pressure gradient and source terms (such as gravity and head losses)
- Set key word `iphydr` to 2 in `cs_user_parameters.f90` to compute and add a hydrostatic pressure who handles the imbalance between pressure gradient and gravity source term (currently limited to orthogonal meshes)

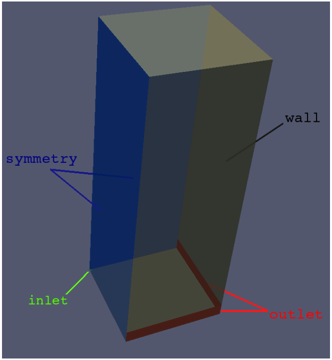
Forces unbalance effects on stratification erosion

Air-fountain test case - stratification erosion [a]

[a] E. Deri and al. J. Heat Fluid Flows (2008)

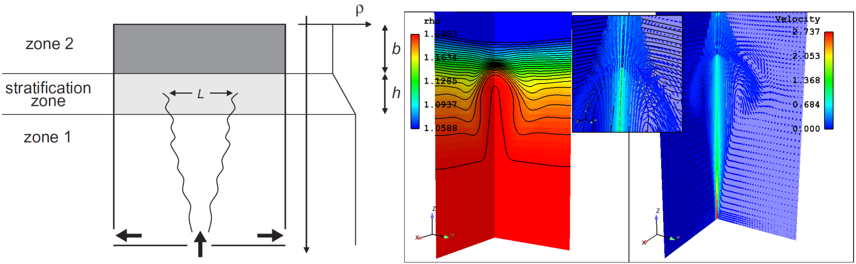






Forces unbalance effects on stratification erosion

[iphydr = 2]: Air-fountain test case - stratification erosion



New turbulence models

Dynamic turbulence

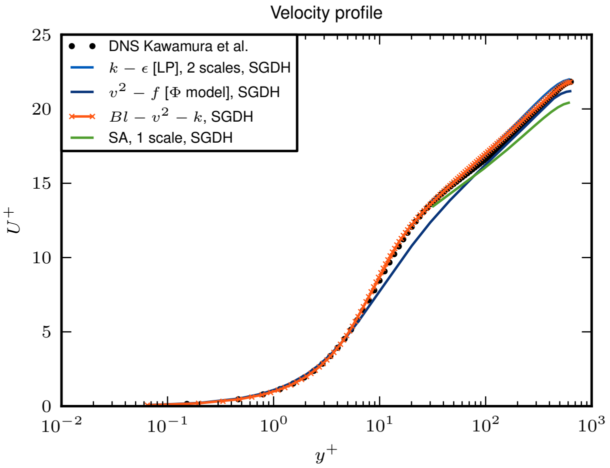
- Spalart-Allmaras turbulence model (aerospace)
- $BL - v^2 - k$ (see Billard *et al.*) (Low-Reynolds EVM)
- EBRSM (Low Reynolds 2^{nd} order closure Reynolds stress model see Manceau *et al.*)
- Synthetic Eddy Method (SEM), (Inlet condition for LES)
- Add a correction term for rotation or curvature on turbulence models

Turbulent heat fluxes (or turbulent fluxes for any scalar)

- Generalized Gradient Diffusion Hypothesis (GGDH) $\overline{T'\underline{u}'} = c_\theta \frac{k}{\epsilon} \underline{\underline{R}} \cdot \underline{\nabla} \overline{T}$
- an other Algebraic Flux Model (AFM: Hanjalic)
- Dynamic Flux Model (DFM: vectorial transport equation on $\overline{T'u'}$)

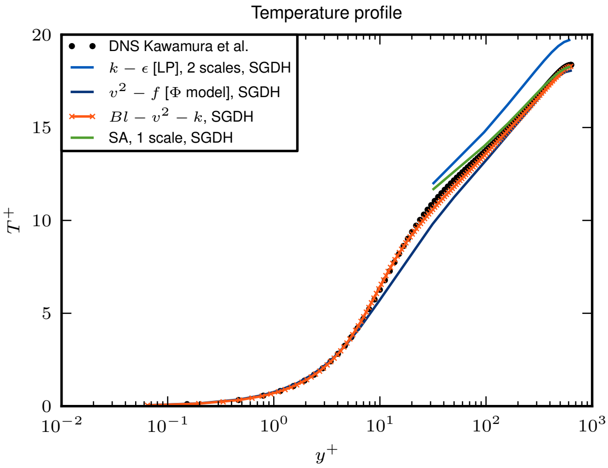
Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

$bl - v^2 - k$ versus the previous version of the $v^2 - f$ model



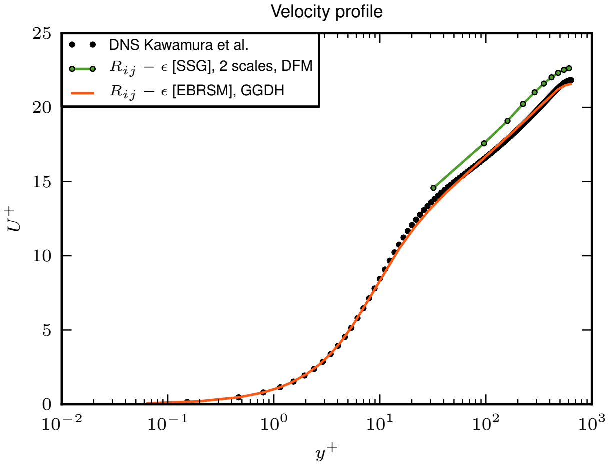
Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

$bl - v^2 - k$ versus the previous version of the $v^2 - f$ model



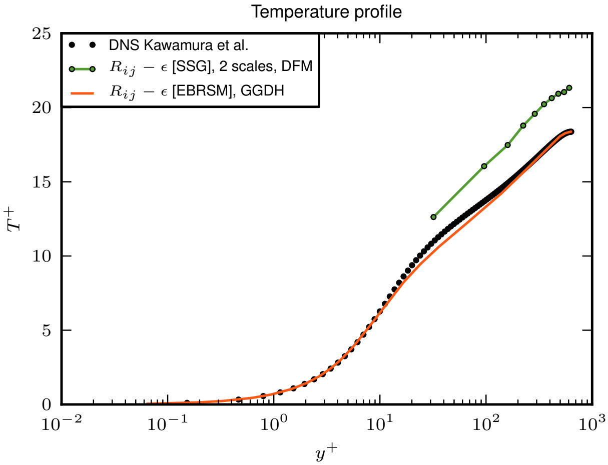
Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

EBRSM versus $R_{ij} - \varepsilon$ SSG



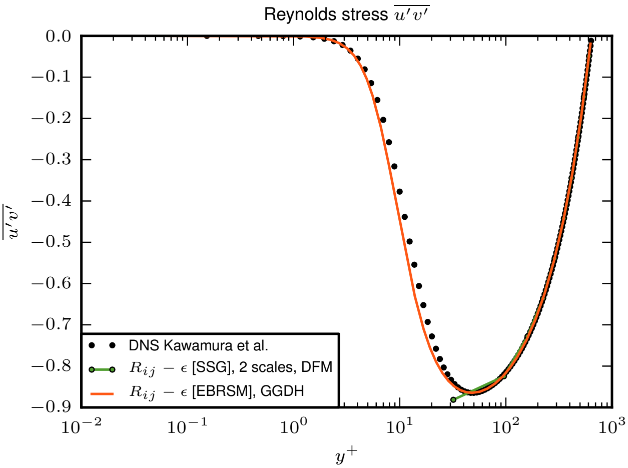
Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

EBRSM versus $R_{ij} - \varepsilon$ SSG



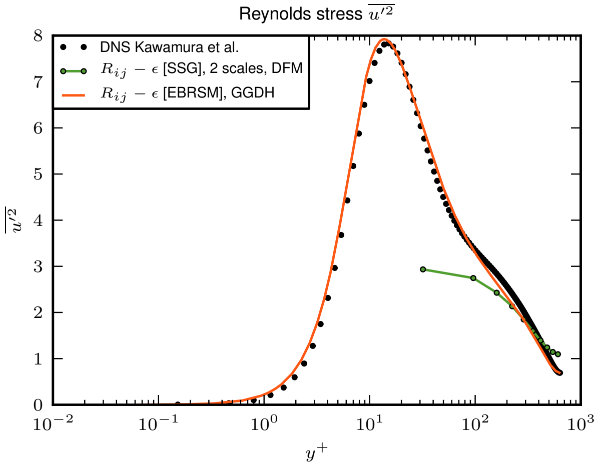
Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

EBRSM versus $R_{ij} - \varepsilon$ SSG



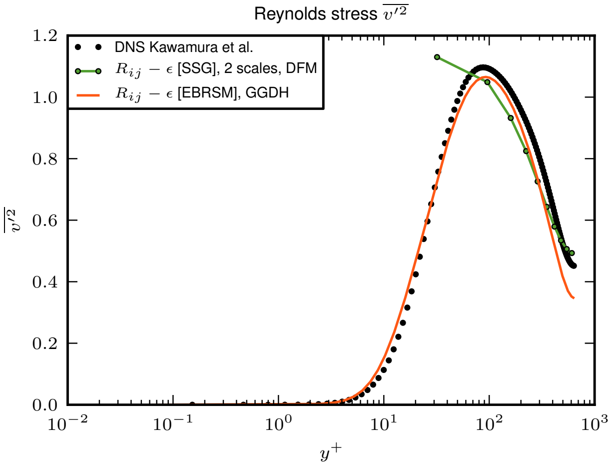
Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

EBRSM versus $R_{ij} - \varepsilon$ SSG



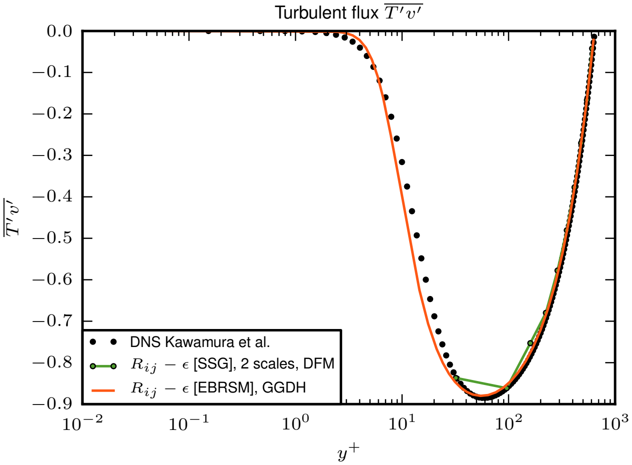
Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

EBRSM versus $R_{ij} - \varepsilon$ SSG



Periodic heated channel of Kawamura *et al.* ($Re_\tau = 640$)

EBRSM versus $R_{ij} - \varepsilon$ SSG



And now for road map...

Overview

1 General user changes
- User subroutines
- runcase
- Changes in default options

2 New features
- Architecture
- Numerics
- Physical modelling
- Turbulence modelling

3 Appendix
- Correction on the Temperature equation
- Porosity module
- Mass conservation in the weakly compressible Algorithm
- Hydrostatic pressure computation for stratified flows

Why change the equation on the temperature

Going back to the energy equation (or enthalpy equation)

$$\rho \frac{\mathrm{d}H}{\partial t} = \rho \frac{\partial H}{\partial t} + \rho \underline{u} \cdot \underline{\nabla} H = \text{div} \left(\underline{q} \right) + \ldots$$

with (Fourier law) $\underline{q} = \dfrac{\lambda}{C_p} \underline{\nabla} T$

Moreover $\mathrm{d}H = C_p \mathrm{d}T + \dfrac{1 - \beta T}{\rho} \mathrm{d}P$ (thermodynamics) So, for incompressible flows (or perfect gas)

$$\mathrm{d}H = C_p \mathrm{d}T$$

So the Temperature Equation reads

$$\rho C_p \frac{\mathrm{d}T}{\partial t} = \rho C_p \frac{\partial T}{\partial t} + \rho C_p \underline{u} \cdot \underline{\nabla} T = \text{div} \left(\lambda \underline{\nabla} T \right) + \left(\frac{1 - \beta T}{\rho} \mathrm{d}P \right) + \ldots$$

So C_p is now kept at his place (crucial when C_p varies in space).

Porosity formulation

Continuous formulation of the momentum Eq. and the continuity Eq. (Conservative form)

$$
\begin{aligned}
\frac{\partial\left(\epsilon\rho\underline{u}\right)}{\partial t}+\underline{\operatorname{div}}\left(\underline{u}\otimes\epsilon\rho\underline{u}\right) &= \underline{\operatorname{div}}\left(\epsilon\mu\underline{\underline{\nabla}}\,\underline{u}\right)-\epsilon\underline{\nabla}P \\
\frac{\partial\left(\epsilon\rho\right)}{\partial t}+\operatorname{div}\left(\epsilon\rho\underline{u}\right) &= 0
\end{aligned}
$$

Continuous formulation of the momentum Eq. and the continuity Eq. (Non-conservative form)

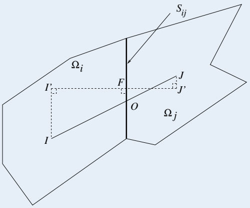
$$
\begin{aligned}
\epsilon\rho\frac{\partial\underline{u}}{\partial t}+\overbrace{\underline{\operatorname{div}}\left(\underline{u}\otimes\epsilon\rho\underline{u}\right)-\operatorname{div}\left(\epsilon\rho\underline{u}\right)\underline{u}}^{\equiv\underline{\underline{\nabla}}\,\underline{u}\cdot\left(\epsilon\rho\underline{u}\right)} &= \underline{\operatorname{div}}\left(\epsilon\mu\underline{\underline{\nabla}}\,\underline{u}\right)-\epsilon\underline{\nabla}P \\
\frac{\partial\left(\epsilon\rho\right)}{\partial t}+\operatorname{div}\left(\epsilon\rho\underline{u}\right) &= 0
\end{aligned}
$$

Porosity formulation

Discrete formulation of the momentum Eq. (Prediction–Correction)

$$\left.\begin{aligned} \epsilon_i \rho_i \left|\Omega_i\right| \frac{\underline{\widetilde{u}} - \underline{u}^n}{\Delta t} + \sum_f \left(\underline{\widetilde{u}}_f - \underline{\widetilde{u}}_i\right) \left(\epsilon \rho \underline{u}\right)^n_f \cdot \underline{S}_f = & \sum_{f_{ij}} \left(\epsilon \mu S\right)_{f_{ij}} \frac{\underline{\widetilde{u}}_{J'} - \underline{\widetilde{u}}_{I'}}{I'J'} \\ & + \sum_{f_b} \left(\epsilon \mu S\right)_{f_b} \frac{\underline{\widetilde{u}}_{f_b} - \underline{\widetilde{u}}_{I'}}{I'F} \\ & - \epsilon_i \left|\Omega_i\right| \underline{\nabla}_i P^n \end{aligned}\right\} (P)$$

$$\left.\begin{aligned} \epsilon_i \rho_i \left|\Omega_i\right| \frac{\underline{u}^{n+1} - \underline{\widetilde{u}}}{\Delta t} + \epsilon_i \left|\Omega_i\right| \underline{\nabla}_i \left(P^{n+1} - P^n\right) = 0 \\ \frac{\partial \left(\epsilon \rho\right)}{\partial t} + \operatorname{div}\left(\epsilon \rho \underline{u}^{n+1}\right) = 0 \end{aligned}\right\} (C)$$



Porosity utilisation

Conclusion on the formulation

- modification of the mass flux at the cell faces,
- modification of the diffusivity at the cell faces,
- pressure gradient, sources terms and non stationary terms are multiplied by the porosity.

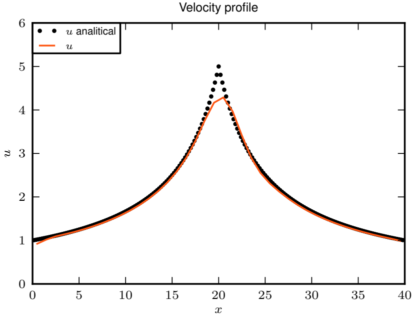
Porosity module available only using Fortran user subroutine

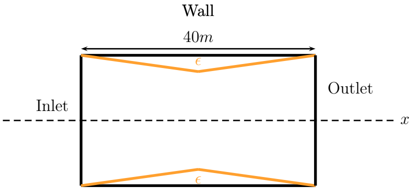
- Set key word `iporos` to 1 in `cs_user_parameters.f90`,
- Set $\epsilon_i \in]0, 1]$ for all $i \in NCELET$ in `usporo.f90`.

Analytical test case

Problem description

- Laminar flow $\nu = \frac{1}{12} m^2.s^{-1}$, $u^{inlet} = 1 m.s^{-1}$
- ϵ profile is linear by part
- 40 cells





Current formulation of Navier Stokes and scalar transport equations

Current time-discretized formulation of the Navier Stokes equation and of the transport equation of a scalar

$$(R)\left\{ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n\right)\right.$$

$$(P)\left\{\begin{array}{rcl} \rho^n \dfrac{\underline{\widetilde{u}} - \underline{u}^n}{\Delta t} + \underbrace{\underline{\underline{\nabla}}\, \underline{\widetilde{u}} \cdot \left(\rho^{n-1} \underline{u}^n\right)}_{\underline{\mathrm{div}}\left(\underline{\widetilde{u}} \otimes \rho^{n-1} \underline{u}^n\right) - \mathrm{div}\left(\rho^{n-1} \underline{u}^n\right) \underline{\widetilde{u}}} & = & \underline{\mathrm{div}}\left[\mu\left(\underline{\underline{\nabla}}\, \underline{\widetilde{u}} + \underline{\underline{\nabla}}\, \underline{\widetilde{u}}^T\right)\right] \\ & - & \underline{\nabla} P^n + \left(\rho^n - \rho_0\right) \underline{g} \end{array}\right.$$

$$(C)\left\{\begin{array}{rcl} \rho^n \dfrac{\underline{u}^{n+1} - \underline{\widetilde{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \cancel{\dfrac{\partial \rho}{\partial t}} + \mathrm{div}\left(\rho^n \underline{u}^{n+1}\right) & = & 0 \end{array}\right.$$

$$(S)\left\{ \rho^n \frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla} \Phi^{n+1} \cdot \left(\rho^n \underline{u}^{n+1}\right) = \mathrm{div}\left[K \underline{\nabla} \Phi^{n+1}\right] + S_\Phi \right.$$

Current formulation of Navier Stokes and scalar transport equations

Current time-discretized formulation of the Navier Stokes equation and of the transport equation of a scalar

$$(R)\left\{ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n\right) \right.$$

$$(P)\left\{ \begin{aligned} \rho^n \frac{\widetilde{\underline{u}} - \underline{u}^n}{\Delta t} + \underbrace{\underline{\underline{\nabla}}\,\widetilde{\underline{u}} \cdot \left(\rho^{n-1}\underline{u}^n\right)}_{\underline{\mathrm{div}}\left(\widetilde{\underline{u}} \otimes \rho^{n-1}\underline{u}^n\right) - \mathrm{div}\left(\rho^{n-1}\underline{u}^n\right)\widetilde{\underline{u}}} &= \underline{\mathrm{div}}\left[\mu\left(\underline{\underline{\nabla}}\,\widetilde{\underline{u}} + \underline{\underline{\nabla}}\,\widetilde{\underline{u}}^T\right)\right] \\ &- \underline{\nabla} P^n + \left(\rho^n - \rho_0\right)\underline{g} \end{aligned} \right.$$

$$(C)\left\{ \begin{aligned} \rho^n \frac{\underline{u}^{n+1} - \widetilde{\underline{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) &= 0 \\ \frac{\partial \rho}{\partial t} + \mathrm{div}\left(\rho^n \underline{u}^{n+1}\right) &= 0 \end{aligned} \right.$$

$$(S)\left\{ \rho^n \frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla}\Phi^{n+1} \cdot \left(\rho^n \underline{u}^{n+1}\right) = \mathrm{div}\left[K \underline{\nabla}\Phi^{n+1}\right] + S_\Phi \right.$$

Current formulation of Navier Stokes and scalar transport equations

Current time-discretized formulation of the Navier Stokes equation and of the transport equation of a scalar

$$(R)\left\{ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n\right) \right.$$

$$(P)\left\{ \begin{array}{lcl} \rho^n \dfrac{\widetilde{\underline{u}} - \underline{u}^n}{\Delta t} + \underbrace{\underline{\underline{\nabla}}\, \widetilde{\underline{u}} \cdot \left(\rho^{n-1} \underline{u}^n\right)}_{\underline{\mathrm{div}}\left(\widetilde{\underline{u}} \otimes \rho^{n-1} \underline{u}^n\right) - \mathrm{div}\left(\rho^{n-1} \underline{u}^n\right) \widetilde{\underline{u}}} & = & \underline{\mathrm{div}}\left[\mu\left(\underline{\underline{\nabla}}\, \widetilde{\underline{u}} + \underline{\underline{\nabla}}\, \widetilde{\underline{u}}^T\right)\right] \\ & - & \underline{\nabla} P^n + \left(\rho^n - \rho_0\right) \underline{g} \end{array} \right.$$

$$(C)\left\{ \begin{array}{lcl} \rho^n \dfrac{\underline{u}^{n+1} - \widetilde{\underline{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \cancel{\dfrac{\partial \rho}{\partial t}} + \mathrm{div}\left(\rho^n \underline{u}^{n+1}\right) & = & 0 \end{array} \right.$$

$$(S)\left\{ \rho^n \frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla} \Phi^{n+1} \cdot \left(\rho^n \underline{u}^{n+1}\right) = \mathrm{div}\left[K \underline{\nabla} \Phi^{n+1}\right] + S_\Phi \right.$$

Dilatable formulation

Time-discretization of the Correction step

$$(R)\left\{ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n, P^n_{thermodynamics}\right)\right.$$

$$(C)\left\{\begin{array}{rcl} \rho^n \dfrac{\underline{u}^{n+1} - \widetilde{\underline{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\rho^n \underline{u}^{n+1}\right) & = & 0 \end{array}\right.$$

Time-discretized scalar transport equation

$$(S)\left\{ \underbrace{\rho^{n-1}\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \overbrace{\underline{\nabla}\Phi^{n+1}\cdot\left(\rho^n \underline{u}^{n+1}\right)}^{\text{div}\left(\Phi^{n+1}\rho^n \underline{u}^{n+1}\right) - \text{div}\left(\rho^n \underline{u}^{n+1}\right)\Phi^{n+1}}}_{\frac{\rho^n\Phi^{n+1} - \rho^{n-1}\Phi^n}{\Delta t} + \text{div}\left(\Phi^{n+1}\rho^n \underline{u}^{n+1}\right)} = \text{div}\left[K\underline{\nabla}\Phi^{n+1}\right] + S_\Phi \right.$$

Dilatable formulation

Time-discretization of the Correction step

$$(R)\left\{\ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n, P^n_{thermodynamics}\right)\right.$$

$$(C)\left\{\begin{array}{rcl} \rho^n \dfrac{\underline{u}^{n+1} - \underline{\widetilde{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \mathrm{div}\left(\rho^n \underline{u}^{n+1}\right) & = & 0 \end{array}\right.$$

Time-discretized scalar transport equation

$$(S)\left\{\ \underbrace{\rho^{n-1}\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \overbrace{\underline{\nabla}\Phi^{n+1}\cdot\left(\rho^n \underline{u}^{n+1}\right)}^{\substack{\mathrm{div}\left(\Phi^{n+1}\rho^n\underline{u}^{n+1}\right) \\ -\mathrm{div}\left(\rho^n\underline{u}^{n+1}\right)\Phi^{n+1}}}}_{\frac{\rho^n\Phi^{n+1} - \rho^{n-1}\Phi^n}{\Delta t} + \mathrm{div}\left(\Phi^{n+1}\rho^n\underline{u}^{n+1}\right)} = \mathrm{div}\left[K\underline{\nabla}\Phi^{n+1}\right] + S_\Phi\right.$$

Dilatable formulation

Prediction step

To preserve the property:

$$(P)\left\{ \frac{\rho^n \underline{\widetilde{u}} - \rho^{n-1}\underline{u}^n}{\Delta t} + \underline{\mathrm{div}}\left(\underline{\widetilde{u}} \otimes \underline{q}\right) = \rho^{n-1}\frac{\underline{\widetilde{u}} - \underline{u}^n}{\Delta t} + \underline{\mathrm{div}}\left(\underline{\widetilde{u}} \otimes \underline{q}\right) + \underbrace{\frac{\rho^n - \rho^{n-1}}{\Delta t}}_{-\mathrm{div}\left(\underline{q}\right)}\underline{\widetilde{u}} \right.$$

the convective flux $\underline{q}$ must verify the the continuity equation(C) with $\dfrac{\rho^n - \rho^{n-1}}{\Delta t}$. Note that the convective flux $\rho^{n-1}\underline{u}^n$ verifies (C) with $\dfrac{\rho^{n-1} - \rho^{n-2}}{\Delta t}$.

Dilatable formulation

Prediction step

To preserve the property:

$$(P)\left\{\begin{array}{l} \dfrac{\rho^n \underline{\widetilde{u}} - \rho^{n-1}\underline{u}^n}{\Delta t} + \underline{\mathrm{div}}\left(\underline{\widetilde{u}} \otimes \underline{q}\right) = \rho^{n-1}\dfrac{\underline{\widetilde{u}} - \underline{u}^n}{\Delta t} + \underline{\mathrm{div}}\left(\underline{\widetilde{u}} \otimes \underline{q}\right) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad + \underbrace{\dfrac{\rho^n - \rho^{n-1}}{\Delta t}}_{-\mathrm{div}\left(\underline{q}\right)}\underline{\widetilde{u}} \end{array}\right.$$

the convective flux $\underline{q}$ must verify the the continuity equation (C) with $\dfrac{\rho^n - \rho^{n-1}}{\Delta t}$. Note that the convective flux $\rho^{n-1}\underline{u}^n$ verifies (C) with $\dfrac{\rho^{n-1} - \rho^{n-2}}{\Delta t}$.

Dilatable formulation

Prediction step

So that the summation of the prediction and the correction steps is:

$$
\begin{array}{ll}
(P) \left\{ \rho^{n-1} \dfrac{\widetilde{\underline{u}} - \underline{u}^n}{\Delta t} + \underline{\operatorname{div}} \left(\widetilde{\underline{u}} \otimes \underline{q} \right) \underbrace{- \operatorname{div} \left(\underline{q} \right)}_{\dfrac{\rho^n - \rho^{n-1}}{\Delta t}} \widetilde{\underline{u}} \right. & = -\underline{\nabla} P^n + \cdots \\
+ (C) \left\{ \rho^n \dfrac{\underline{u}^{n+1} - \widetilde{\underline{u}}}{\Delta t} \right. & = -\underline{\nabla} \left(P^{n+1} - P^n \right) \\
= \left\{ \dfrac{\rho^n \underline{u}^{n+1} - \rho^{n-1} \underline{u}^n}{\Delta t} + \underline{\operatorname{div}} \left(\widetilde{\underline{u}} \otimes \underline{q} \right) \right. & = -\underline{\nabla} P^{n+1} + \cdots
\end{array}
$$

Dilatable formulation

Prediction step

It was chosen to solve an additional Poisson equation to obtain a convective field $\underline{q}$ so that:

$$(MC)\left\{\begin{array}{rcl}\underline{q}\cdot S_f = \rho^{n-1}\underline{u}^n\cdot\underline{S}_f - \Delta t\underline{\nabla}_f\left(Potential\right)\cdot\underline{S}_f & = & 0\\ \dfrac{\rho^n-\rho^{n-1}}{\Delta t}+\text{div}\left(\underline{q}\right) & = & 0\end{array}\right.$$

Dilatable formulation

Prediction step

So the prediction step becomes:

$$(P)\left\{\begin{array}{lcl} \rho^{n-1}\dfrac{\widetilde{\underline{u}}-\underline{u}^n}{\Delta t}+\underbrace{\underline{\underline{\nabla}}\,\widetilde{\underline{u}}\cdot\underline{q}}_{\underline{\mathrm{div}}\left(\widetilde{\underline{u}}\otimes\underline{q}\right)-\mathrm{div}\left(\underline{q}\right)\widetilde{\underline{u}}} & = & \underline{\mathrm{div}}\left[\mu\left(\underline{\underline{\nabla}}\widetilde{\underline{u}}+\underline{\underline{\nabla}}\widetilde{\underline{u}}^T\right)\right] \\ & - & \underline{\nabla}P^n+\left(\rho^n-\rho_0\right)\underline{g} \end{array}\right.$$

and "conserve" momentum over time.

Dilatable formulation

Prediction step

So the prediction step becomes:

$$(P)\left\{\begin{array}{rcl} \rho^{n-1}\dfrac{\widetilde{\underline{u}}-\underline{u}^n}{\Delta t}+\underbrace{\underline{\underline{\nabla}}\,\widetilde{\underline{u}}\cdot\underline{q}}_{\underline{\text{div}}\left(\widetilde{\underline{u}}\otimes\underline{q}\right)-\text{div}\left(\underline{q}\right)\widetilde{\underline{u}}} & = & \underline{\text{div}}\left[\mu\left(\underline{\underline{\nabla}}\widetilde{\underline{u}}+\underline{\underline{\nabla}}\widetilde{\underline{u}}^T\right)\right] \\ & - & \underline{\nabla}P^n+\left(\rho^n-\rho_0\right)\underline{g} \end{array}\right.$$

and "conserve" momentum over time.

Summarize of the conservative time stepping

Time stepping

$$(R)\left\{\ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n, P^n_{thermodynamics}\right)\right.$$

$$(MC)\left\{\begin{array}{rcl} \dfrac{\underline{q} - \rho^{n-1}\underline{u}^n}{\Delta t} - \underline{\nabla}_f\left(Potential\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\underline{q}\right) & = & 0 \end{array}\right.$$

$$(P)\left\{\ \rho^{n-1}\frac{\widetilde{\underline{u}} - \underline{u}^n}{\Delta t} + \underline{\underline{\nabla}}\,\widetilde{\underline{u}}\cdot\underline{q} = \underline{\text{div}}\left[\mu\left(\underline{\underline{\nabla}}\widetilde{\underline{u}} + \underline{\underline{\nabla}}\widetilde{\underline{u}}^T\right)\right] - \underline{\nabla}P^n + \left(\rho^n - \rho_0\right)\underline{g}\right.$$

$$(C)\left\{\begin{array}{rcl} \rho^n\dfrac{\underline{u}^{n+1} - \widetilde{\underline{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\rho^n\underline{u}^{n+1}\right) & = & 0 \end{array}\right.$$

$$(S)\left\{\ \rho^{n-1}\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla}\Phi^{n+1}\cdot\left(\rho^n\underline{u}^{n+1}\right) = \text{div}\left[K\underline{\nabla}\Phi^{n+1}\right] + S_\Phi\right.$$

Summarize of the conservative time stepping

Time stepping

$$(R)\left\{\ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n, P^n_{thermodynamics}\right)\right.$$

$$(MC)\left\{\begin{aligned} \frac{\underline{q} - \rho^{n-1}\underline{u}^n}{\Delta t} - \underline{\nabla}_f\left(Potential\right) &= 0 \\ \frac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\underline{q}\right) &= 0 \end{aligned}\right.$$

$$(P)\left\{\ \rho^{n-1}\frac{\widetilde{\underline{u}} - \underline{u}^n}{\Delta t} + \underline{\underline{\nabla}}\,\widetilde{\underline{u}} \cdot \underline{q} = \underline{\text{div}}\left[\mu\left(\underline{\underline{\nabla}}\widetilde{\underline{u}} + \underline{\underline{\nabla}}\widetilde{\underline{u}}^T\right)\right] - \underline{\nabla}P^n + \left(\rho^n - \rho_0\right)\underline{g}\right.$$

$$(C)\left\{\begin{aligned} \rho^n\frac{\underline{u}^{n+1} - \widetilde{\underline{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) &= 0 \\ \frac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\rho^n\underline{u}^{n+1}\right) &= 0 \end{aligned}\right.$$

$$(S)\left\{\ \rho^{n-1}\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla}\Phi^{n+1} \cdot \left(\rho^n\underline{u}^{n+1}\right) = \text{div}\left[K\underline{\nabla}\Phi^{n+1}\right] + S_\Phi\right.$$

Summarize of the conservative time stepping

Time stepping

$$(R)\left\{ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n, P^n_{thermodynamics}\right)\right.$$

$$(MC)\left\{ \begin{array}{rcl} \dfrac{\underline{q} - \rho^{n-1}\underline{u}^n}{\Delta t} - \underline{\nabla}_f\left(Potential\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\underline{q}\right) & = & 0 \end{array}\right.$$

$$(P)\left\{ \rho^{n-1}\frac{\widetilde{\underline{u}} - \underline{u}^n}{\Delta t} + \underline{\underline{\nabla}}\,\widetilde{\underline{u}} \cdot \underline{q} = \underline{\text{div}}\left[\mu\left(\underline{\underline{\nabla}}\widetilde{\underline{u}} + \underline{\underline{\nabla}}\widetilde{\underline{u}}^T\right)\right] - \underline{\nabla}P^n + \left(\rho^n - \rho_0\right)\underline{g}\right.$$

$$(C)\left\{ \begin{array}{rcl} \rho^n\dfrac{\underline{u}^{n+1} - \widetilde{\underline{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\rho^n\underline{u}^{n+1}\right) & = & 0 \end{array}\right.$$

$$(S)\left\{ \rho^{n-1}\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla}\Phi^{n+1} \cdot \left(\rho^n\underline{u}^{n+1}\right) = \text{div}\left[K\underline{\nabla}\Phi^{n+1}\right] + S_\Phi\right.$$

Summarize of the conservative time stepping

Time stepping

$$(R)\left\{\ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n, P^n_{thermodynamics}\right)\right.$$

$$(MC)\left\{\begin{array}{rcl} \dfrac{\underline{q} - \rho^{n-1}\underline{u}^n}{\Delta t} - \underline{\nabla}_f\left(Potential\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\underline{q}\right) & = & 0 \end{array}\right.$$

$$(P)\left\{\ \rho^{n-1}\frac{\underline{\widetilde{u}} - \underline{u}^n}{\Delta t} + \underline{\underline{\nabla}}\,\underline{\widetilde{u}}\cdot\underline{q} = \underline{\text{div}}\left[\mu\left(\underline{\underline{\nabla}}\,\underline{\widetilde{u}} + \underline{\underline{\nabla}}\,\underline{\widetilde{u}}^T\right)\right] - \underline{\nabla}P^n + (\rho^n - \rho_0)\,\underline{g}\right.$$

$$(C)\left\{\begin{array}{rcl} \rho^n\dfrac{\underline{u}^{n+1} - \underline{\widetilde{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\rho^n\underline{u}^{n+1}\right) & = & 0 \end{array}\right.$$

$$(S)\left\{\ \rho^{n-1}\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla}\Phi^{n+1}\cdot\left(\rho^n\underline{u}^{n+1}\right) = \text{div}\left[K\underline{\nabla}\Phi^{n+1}\right] + S_\Phi\right.$$

Summarize of the conservative time stepping

Time stepping

$$(R)\left\{\ \rho^n = \rho\left(\Phi_1^n, \cdots, \Phi_k^n, P^n_{thermodynamics}\right)\right.$$

$$(MC)\left\{\begin{array}{rcl} \dfrac{\underline{q} - \rho^{n-1}\underline{u}^n}{\Delta t} - \underline{\nabla}_f\left(Potential\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\underline{q}\right) & = & 0 \end{array}\right.$$

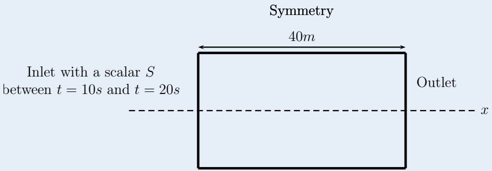
$$(P)\left\{\ \rho^{n-1}\frac{\underline{\widetilde{u}} - \underline{u}^n}{\Delta t} + \underline{\underline{\nabla}}\,\underline{\widetilde{u}} \cdot \underline{q} = \underline{\text{div}}\left[\mu\left(\underline{\underline{\nabla}}\,\underline{\widetilde{u}} + \underline{\underline{\nabla}}\,\underline{\widetilde{u}}^T\right)\right] - \underline{\nabla}P^n + \left(\rho^n - \rho_0\right)\underline{g}\right.$$

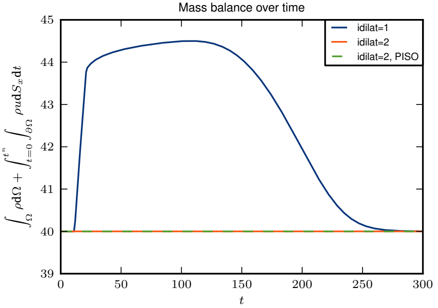
$$(C)\left\{\begin{array}{rcl} \rho^n\dfrac{\underline{u}^{n+1} - \underline{\widetilde{u}}}{\Delta t} + \underline{\nabla}\left(P^{n+1} - P^n\right) & = & 0 \\ \dfrac{\rho^n - \rho^{n-1}}{\Delta t} + \text{div}\left(\rho^n\underline{u}^{n+1}\right) & = & 0 \end{array}\right.$$

$$(S)\left\{\ \rho^{n-1}\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + \underline{\nabla}\Phi^{n+1} \cdot \left(\rho^n\underline{u}^{n+1}\right) = \text{div}\left[K\underline{\nabla}\Phi^{n+1}\right] + S_\Phi\right.$$

Dilatable algorithm

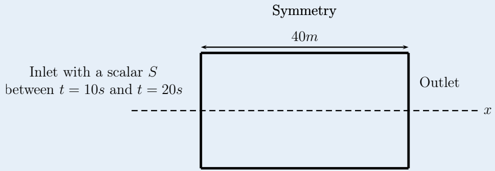
Test case

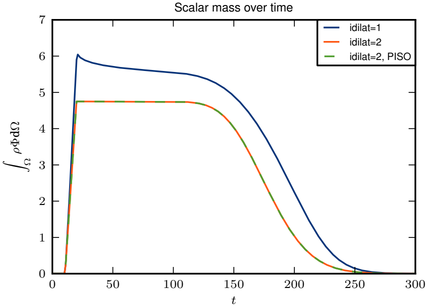




Dilatable algorithm

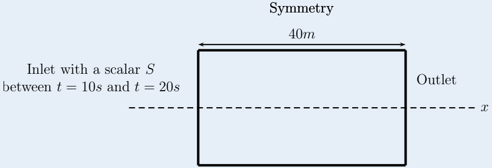
Test case

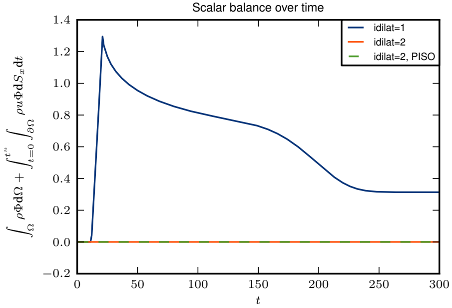




Dilatable algorithm

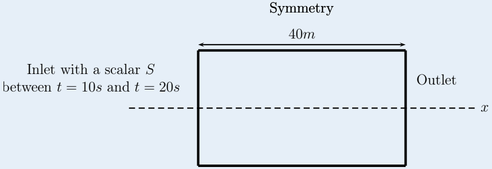
Test case

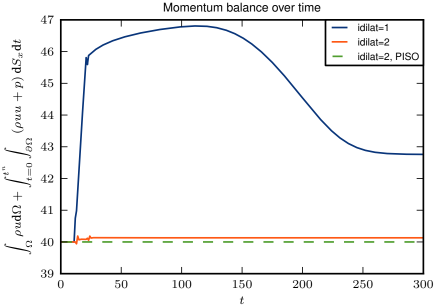




Dilatable algorithm

Test case





Utilisation of the dilatable and Low Mach algorithm

Settings using Fortran user subroutine

- Set key word `idilat` to 2 in `cs_user_parameters.f90` for the dilatable algorithm,
- Set key word `idilat` to 3 in `cs_user_parameters.f90` for the Low Mach number algorithm in confined flows (then $P_{thermodynamics}$ is added as a source term in the enthalpy equation).

Hydrostatic pressure computation – iphydr =2

The hydrostatic pressure gradient resolution

With the discrete formulation of the "*a priori*" simplified momentum equation as below:

$$\rho^n \frac{\left(\underline{u}_{hydro} - \underline{u}^n\right)}{\Delta t} = \left(\rho \underline{g}\right)^n - \underline{\nabla} P^n_{hydro}$$

$$\rho^n \underline{\text{div}} \left(\delta \underline{u}_{hydro}\right) = 0$$

finally we resolve the simplified momentum equation:

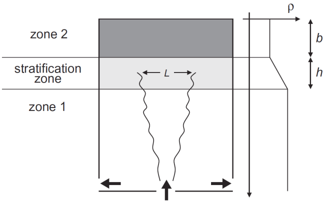
$$\underline{\text{div}} \left(K \underline{\nabla} P_{hydro}\right) = \underline{\text{div}} \left(\underline{g}\right)$$

with: $K \equiv (1/\rho^n)$ and a Neumann boundary condition on the hydrostatic pressure:

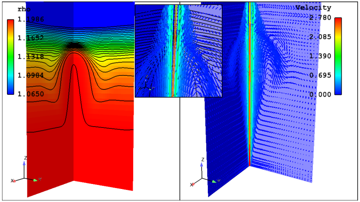
$$D_{fib}\left(K, P_{hydro}\right) = \underline{g} \cdot \underline{n}_{ib}$$

Hydrostatic pressure computation for stratified flows

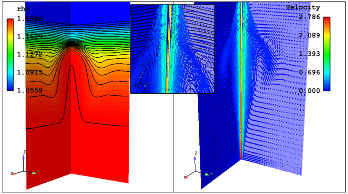
[iphydr]: Air-fountain test case - stratification erosion



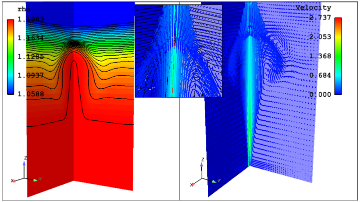
iphydr=0



iphydr=1



iphydr=2



And now for road map...

Prospects for *Code_Saturne* 4.0 and beyond…

Code_Saturne User Meeting

April 9, 2013

Improving user experience

- **The documentation: an on-going work to achieve**
  - Theory manual: complete overhauling and translation in English
  - Source code: complete translation in English
  - Online documentation (Doxygen) of sources and user examples
  - Best practice guidelines (six guides available online)
  - Tutorials: one for each code feature (six tutorials available online)

- **Graphical User Interface**
  - SALOME platform: from a good existing tool to a “can’t live without it” platform

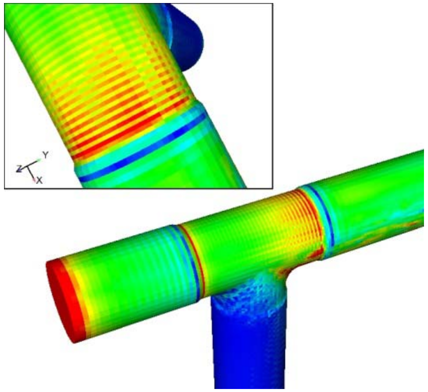
- **Meshing tools**
  - Automatic settings of boundary layer made of prisms from an existing mesh
  - Automatic homogeneous refinement with simple geometry for Verification (AutoVnV tool)
  - Volumic smoother and local remeshing of bad quality mesh

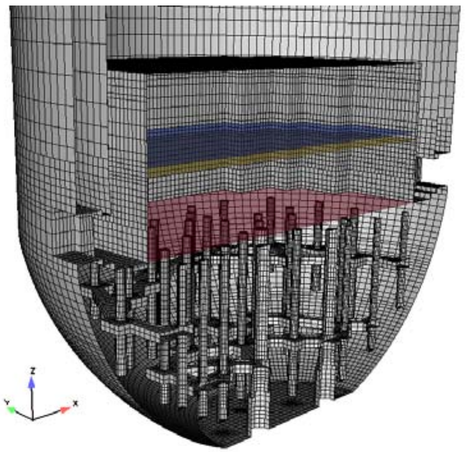
Developments to be integrated or consolidated (not exhaustive…)

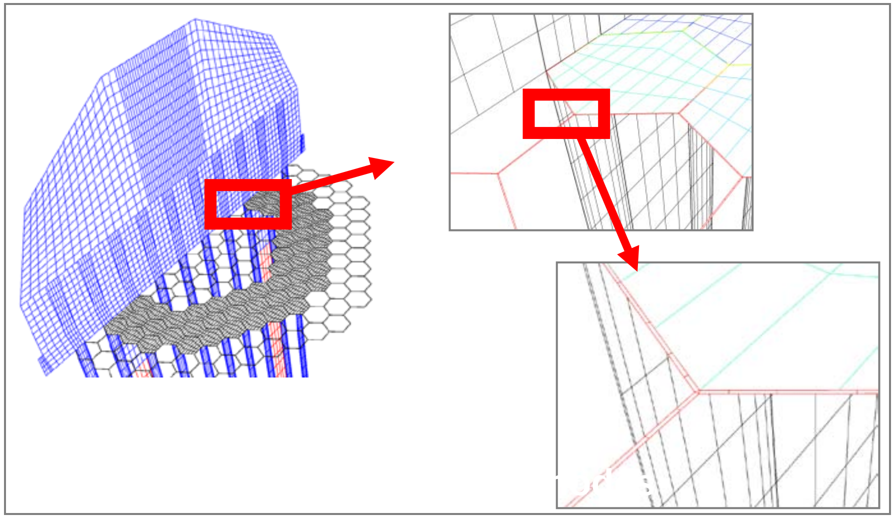
- Atmospheric flows: micro-chemistery of the atmosphere, …
- Lagrangian: particles re-suspension and agglomeration
- Fire driven flows specific enhancements
- Combustion: new Nox models, drift flux between particles, …
- Turbomachinery: fully conservative rotor/stator algorithm by mesh joining
- Radiative transfers: non-grey gas models, WBM approach, …
- Turbulence modelling: zonal RANS / LES coupling with volumetric forcing
- Compressible flows: new algorithm based on pressure
- Aerosol transport: drift flux models for heavy scalars

Roadmap for numerics

- **Increasing accuracy and robustness for any mesh quality/type**
  - Get rid of numerical limitations of cell-centered schemes?
  - Increasing the coupling level of equations
  - Work on new promising numerical approach
    - Recent progress in Finite Volume schemes
    - Compatible Discrete Operator

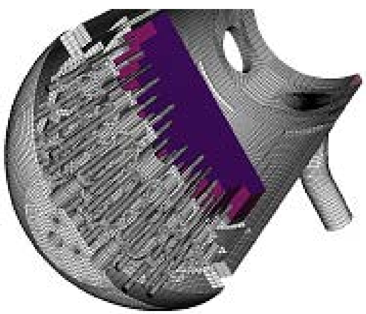


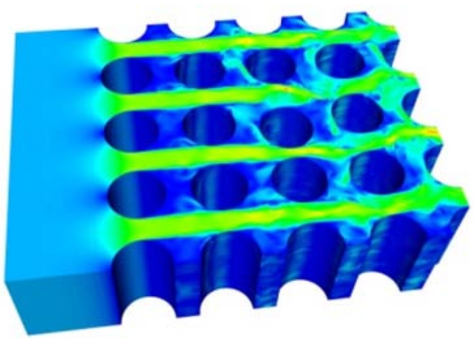


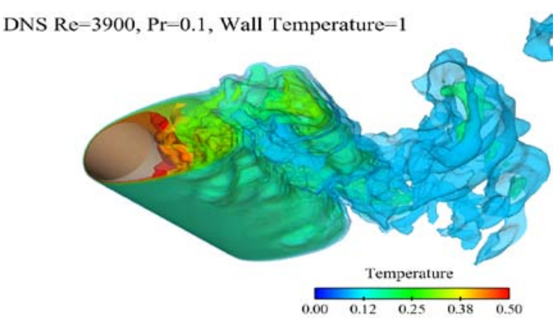


Roadmap for physical modelling

- **Turbulence modelling**
  - Toward a second order low/high Reynolds turbulence model for all meshes
  - Improvements in species and temperature transport and mixing
- **Homogeneous two-phase flows**
  - Models for cavitating flows
- **Improvement of simulation of flows in porous medium**
  - Interface between Fluid and porous medium







Thank you for your attention

And a special thank for every contribution
to *Code_Saturne*, users and developers!