Notus CFD code presentation

S. Glockner, A. Lemoine, M. Coquerelle, J. Picot,
A. Jost, F. Henri, F. Desmons, F. Salmon, ...

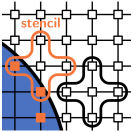
Institut de Mécanique et d'Ingénierie de Bordeaux
Université de Bordeaux, Bordeaux-INP, CNRS UMR 52 95

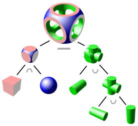
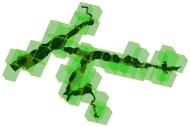
https://notus-cfd.org

Journées Calcul & Simulation - Dec. 2021

Presentation content

- The main features of Notus
- Short focus of some proposed numerical methods
- GENCI HPC challenge 2020 at TGCC

Notus code purposes

Open-source project started from scratch in 2015 (CeCILL Licence)

- Modeling and simulation of **incompressible fluid flows**, multiphysics
- 2D/3D Finite Volume/Difference methods on staggered grids, **massively parallel**
- **Still under development (version 0.5.0)**
- From 0.5.0 to 1.0.0, make all things work together!

Intended users

- **Mechanical community**: easy to use and adapt, proven state-of-the-art numerical methods, towards numerical experiments
- **Mathematical community**: develop new numerical schemes, fast and efficient framework for comparative and qualitative tests, benchmark methods on identified physical test cases, numerical toolbox
- Take advantage of **synergies between Research** / **Teaching** / **Industry** / **HPC**

What is not Notus

- A concurrent of, a commercial tool, a click button code

Supercomputers

- **GENCI/PRACE**: Joliot Curie at TGCC, Occigen at CINES, Jean-Zay at IDRIS
- Curta at mesocentre MCIA... *also on Linux laptop!*

Development environment

Modern development framework

- **Fortran 2008**
- **MPI** parallel coding library
- **OpenMP** share memory parallel coding library
- **Mask parallelism** complexities for easy programming
- **Git** distributed version control system
- **CMake** cross-platform build system → easy installation
- **Doxygen** documentation generator from source code
- **Linux only!**
- **Build scripts, Notus and third party libraries**
- A thoroughly **validated and documented code**, **non-regression** approach
- **Web sites**: general, doc, git
  https://notus-cfd.org, https://doc.notus-cfd.org, https://git.notus-cfd.org

Portability

- GNU + OpenMPI; Intel + IntelMPI
- Sequential and Parallel versions
- → "Same" results between 10^{-8} and 10^{-15})

Features - Modeling

Domain

- 2D/3D Cartesian, immersed sub-domains

Incompressible Navier-Stokes equations

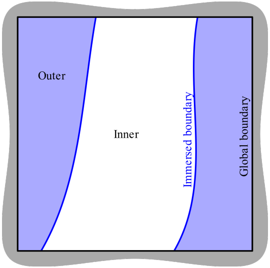
- Buoyancy force (Boussinesq approximation)
- Surface tension force (CSF model)
- Large Eddy Simulation (mixed scale model, WALE)
- RANS k-ω SST

Multiphase immiscible flows

- One-fluid model
- Volume-of-Fluid, Moment-of-Fluid, Level-Set interface representations

Energy equation, Species transport equations

- Free, mixed or forced convection, phase change liquid/solid
- Passive scalar, thermosolutal flows



Features - Numerical methods

Discretization

- **2D/3D Cartesian on staggered grids**, automatic partitioning
- **Implicit schemes**: up to O(2) implicit schemes (advection and diffusion)
- **Explicit schemes**: O(2) TVD LV Superbee, O(3) & O(5) WENO, HOUC (advection); O(2) & O(4) centered (diffusion)
- **Complex geometry**: Immersed Boundary Method (1^{st} & 2^{nd} order)

Navier-Stokes

- **Non conservative or momentum preserving approaches**
- **Velocity/pressure coupling**: time splitting methods (Goda, Timmermans)
- **Surface tension**: Height Function or Closest-Point methods to compute curvature

Fluid / fluid interface representation and transport

- **Volume-of-Fluid** method 2D-3D / PLIC (directional splitting)
- **Moment-of-Fluid** method 2D-3D / backward RK2 advection
- **Level-set** / WENO

HYPRE library (LLNL)

- BiCGStab, GMRES iterative solvers
- Preconditioners: **geometric & algebraic multigrid**

MUMPS direct solver

- Mainly for 2D linear systems
- PORD, Metis graph partitioners

Domain is partitioned, data are distributed

→ How to write and plot data efficiently on thousands of processors?

Use of ADIOS library (Oak Ridge National Laboratory)

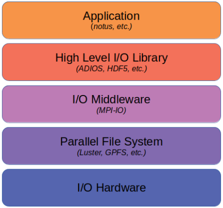
- Open-source
- Simple and flexible way to describe the data
- Masks IO parallelism, different methods: POSIX, MPI-IO, aggregation
- From 1 to 100 000 processors, around $50 GBs^{-1}$ on Joliot Curie Lustre file system

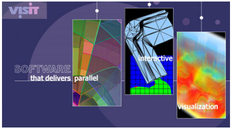
Visualisation of the results → VisIt (LLNL), Paraview

- With ADIOS file format, VisIt is limited to 2 billion cells, **sequential version only**!

Pixie

- Based on **HDF5 library** (`.h5` files)
- **Compatible with parallel VisIt** (automatic parallel domain decomposition)
- Non-uniform rectilinear grids
- Notus Pixie output less efficient then ADIOS, around $8 GBs^{-1}$ on Joliot Curie Lustre file system





Verification and Validation V&V

Notus V&V python script

Non-regression

- **list of V&V test cases files**
- quick or full validation (up to 1600 MPI jobs), **database of reference values for each case**
- validation runs on supercomputers thanks to slurm (2 pass script: submit jobs, collect and post-process)
- results in *txt* file: OK, NO, FAIL, difference to expected values. Summary.

Grid convergence: run the same case varying mesh or time step

- run (interactivly or submission) the test case with different meshes
- collect the results of the chosen quantities, compute convergence order and eventually extrapolated values
- **included into the non-regression process**

Performance python script

- Verify weak and strong scalability
- Identify and measure relevant parts of the code
- Verify I/O performance
- On several supercomputers (from local to GENCI/PRACE)
- Determine optimal use of supercomputers (number of cells per core)
- Compare measured scalability to the expected one
- Ensure non regression of these performances

User Interface: `.nts` file

Concept

- Text `.nts` files (unicode)
- Self-explanatory keywords, precise grammar
- Efficient parser that supports:
  - variable declaration
  - formula
  - 'include'
  - logical tests, loops
- **Associated documentation** → `test_cases/doc` directory

.nts file structure

- Physical fluid properties data base: `std/physical_properties.nts` file
- One `.nts` file per test case, block structure:
  - `include and variable declarations`
  - `system{}`
  - `domain{}`
  - `mesh{}`
  - `modeling{}`
  - `numerical_methods{}`
  - `post_processing{}`

User Interface: `.nts` file example

```
include std "physical_properties.nts";
system { measure_cpu_time;}
domain {
  spatial_dimension 2;
  corner_1_coordinates (0.0, 0.0);
  corner_2_coordinates (1.0, 2.0);
}
grid {
  grid_type regular;
  number_of_cells (32, 32);
}
modeling {
  fluids {fluid "water";}
  equations {
    energy {
      boundary_condition{
        left dirichlet 0.0;
        right dirichlet 1.0;
        top neumann 0.0;
        bottom neumann 0.0;
      }
      source_term {constant -2.0;}
      disable_advection_term;
      disable_temporal_term;
    }
  }
}
numerical_parameters {
  time_iterations 1;
  energy {
    solver mumps_metis;
  }
}
post_processing {
  output_library adios;
  output_frequency 1;
  output_fields temperature;
}
```

Some development keys - Masking parallelism

Numerical domain and MPI process ghost cells

- The global domain is partitioned into subdomains
- Addition of a few layers of cells surrounding the local domain: $nx \times ny \times nz$ cells

MPI generic routines to exchange data

- 2D/3D, whatever overlapping zone size
- Integer, double
- Cell array, or vector defined on staggered grid

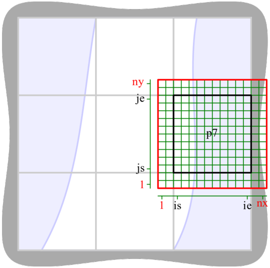
```
call mpi_exchange(pressure)
call mpi_exchange(velocity)
```

- Mandatory after any spatial derivative computations
- MPI Exchange + Fill boundary ghost nodes

```
call fill_ghost_nodes(scalar,
boundary_condition)
```

Global reduction routines

- encapsulate MPI ones
- generic routines for min, max of local arrays, sum of scalars



OpenMP generic algebraic operations for 3-dimensional arrays and face-fields

x = a + b

```
call field_operation_add(a, b, x)
```

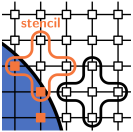
a = a + b*c

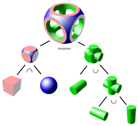
```
call field_operation_add_mult(a,
b, c)
```

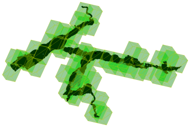
...

Presentation content

- The main features of Notus
- Short focus of some proposed numerical methods
- GENCI HPC challenge 2020 at TGCC







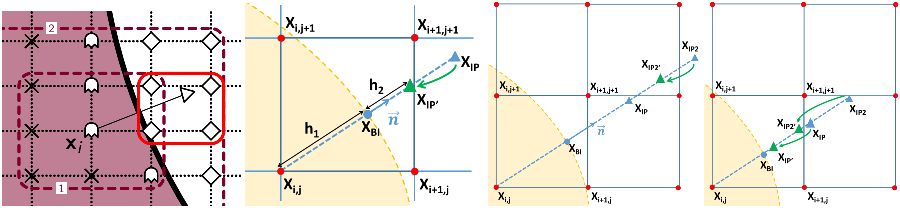
Direct forcing method of Mittal [JCP2008]

- **Extrapolate solution on the ghost nodes thanks a linear relation between the ghost nodes and their image nodes (△)**
  Interpolation of the solution on the image point, Lagrange interpolation, p=2 or p=3 (4 or 9 points in 2D)

  → **non-compact stencils** (loss of precision, non banded matrix, less efficient solver)

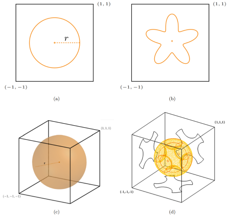
  Regular grid (square cells)
  Dirichlet : stencil size = 2, 2^{nd} order
  Neumann : stencil size = 2, 1^{st} order only → stencil size = 3 (2^{nd} order)

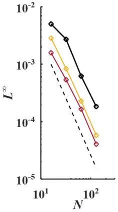
- **Improvements - through stencil size reduction - thanks to different class of shifting methods**
  Ghost node shifting method for irregular grids
  Square shifting methods for regular grid applied to linear interpolation and extended to quadratic one

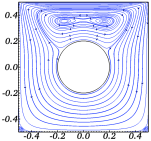


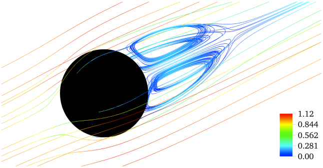
2D/3D verification and validation

- Laplacian: circle, 2D & 3D flowers
- Navier-Stokes: 3D channel, driven cavity with obstable, flow around a heated cylinder, around a sphere
- **Stencil 1:** 2^{nd} **order Dirichlet and** 1^{st} **order Neumann (with several shifting methods)**
- **Stencil 2:** 2^{nd} **order Neumann is possible with quadratic and double shifting method**
- **Conclusion : stencil size reduced, precision and regularity of the convergence also improved**







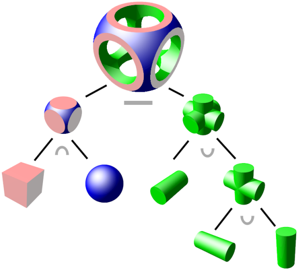


Performance analysis

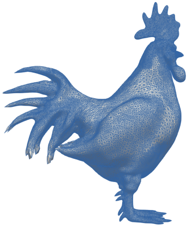
- CPU time. 3D pipe flow: x2.5 faster than original method. Flow past a sphere: x3 faster.

A. Jost, S. Glockner, Direct forcing immersed boundary methods: Improvements to the Ghost Node Method, *Journal of Computational Physics*, volume 438, 110371, 2021.

- Volumetric representation
- Boolean operations (CSG)
  - Union
  - Intersection
  - Difference
  - Complement
- Analytic shapes (primitives)
  - Half-space
  - Sphere
  - Box
  - Cylinder
  - Torus
- Surface mesh
  - Orientable
  - Inside/Outside
  - OBJ Wavefront format
- Transformations
  - Translation
  - Rotation
  - Scaling

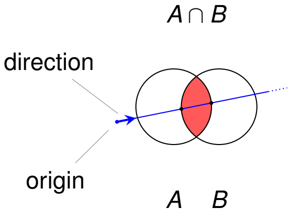


CSG tree (Wikimedia Commons)



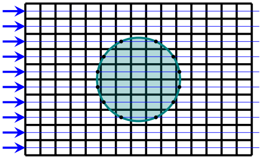
Surface mesh example

Ray-tracing (list of intersection points + distance + normal)

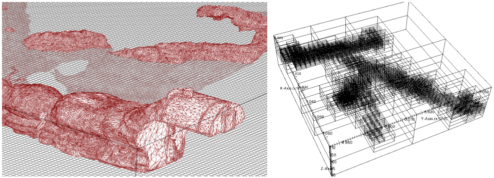


- Ray-tracing with $\cup$, $\cap$, and $\setminus$
- Jordan-Brouwer theorem
  $\rightarrow$ odd: inside
  $\rightarrow$ even: outside

Ray-trace once per row



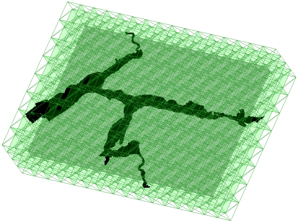
Space subdivision using octree (surface mesh only)

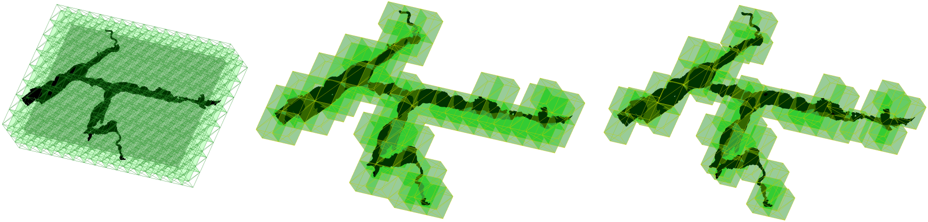


Cartesian grid partitioning and complex geometries

With immersed boundaries

- The number of inactive cells may be large
- Full inactive partitions
- $\rightarrow$ **Extra computational cost** (ex. Lascaux cave: 99% of inactive cells)
- **Solution**: remove inactive partitions and contract line or row of partitions (partition sliding)



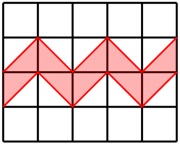


Lascaux cave example, 100 000 cells / core

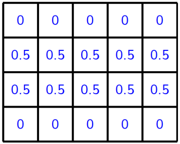
- **Global grid of 189 billion cells** - 99% of inactive cells
- Without sliding 3.2 billion cells - 35% of inactive cells
- **With sliding 3.0 billion cells - 31% of inactive cells**

Volume-of-Fluid - PLIC

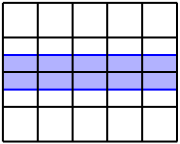
- Volume fraction + **normal** to the interface → linear construction of the interface
- Requires a **9 pts stencil** (2D)



Original interface



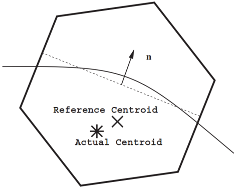
VOF representation



PLIC reconstruction

Moment-of-Fluid

- Volume fraction + **centroid** → linear reconstruction that:
  - matches the volume fraction
  - minimises the discrepancy between the specified centroid and the centroid of the reconstructed polygon
- → **1 pt stencil**, 2nd order
- Generalised to n materials



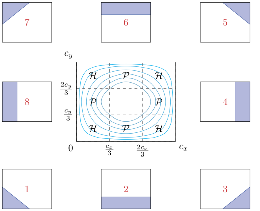
Source: Dyadechko & Shashkov (JCP 2006)

Moment-of-Fluid method

Remove minimisation for Cartesian grids in 2D

- analytic form of the centroid curve (for a given volume fraction)
- from 20% to 300% faster

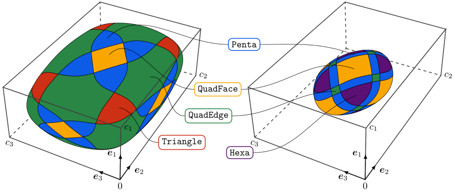
A. Lemoine, S. Glockner, J. Breil, Moment-of-Fluid Analytic Reconstruction on 2D Cartesian Grids, *Journal of Computational Physics*, vol. 328, pp. 131–139, 2017.



Extension to 3D hexahedral cells

- uses explicit analytic formulas of the objective function
- more robust and 100x faster

T. Milcent, A. Lemoine, Moment-of-Fluid Analytic Reconstruction on 3D Rectangular Hexahedrons, *Journal of Computational Physics*, 409, 109346, 2020



→ CPU time reduction as a source of motivation for new numerical methods

Level-set and closest point methods

Surface tension computation

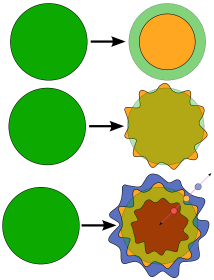
Context

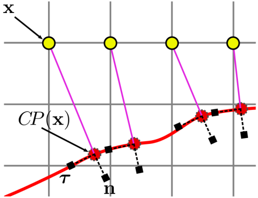
- Accurate computation of the curvature and precise transport of the interface is **still challenging**
- **Continuum Surface Force [Brackbill]**: $\sigma \kappa \nabla c$
- $\kappa = \nabla . (\frac{\nabla \phi}{|\nabla \phi|})$ on nodes where κ is not defined

Solution

- Curvature computation based on second derivatives of the surface/interface $\rightarrow$ **transport at least 4th order precise**
- WENO5/Level-Set framework
- Accurate κ computation
  - compared to exact curvature
  - with minimum variation along the surface
  - with minimum variation following the normal direction
- κ inside the domain = κ of the closest Γ point [Hermann, 2008]
- $\rightarrow$ **Extension of the curvature along the normal direction**

$$\kappa(\mathbf{x}) = \kappa\left(CP(\mathbf{x})\right)$$



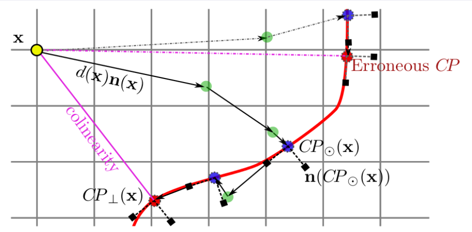


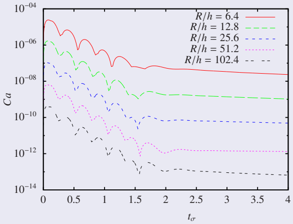
Level-set and closest point methods

Surface tension computation

Contribution

- Level-set $\neq$ distance function
- Improvement of the gradient descent to find the closest points to ensure **colinearity to the interface normal**





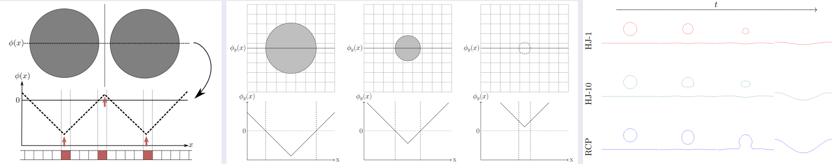
Results

- Viscous column equilibrium: 4th order decrease of spurious current
- **Advected** viscous column: 4th order (not even 1 for VOF method)

M. Coquerelle, S. Glockner, *A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces*, Journal of Computational Physics, 305, pp. 838-876, 2015.

Geometrical level set reinitialization method

- Use the CP approach to reinitialize and to detect precisely all the ill-defined points of the level set (kinks)
- Equivalent or even better results compared to solving the Hamilton-Jacobi equation



F. Henri, M. Coquerelle, P. Lubin, Geometrical level set reinitialization using closest point method and kink detection for thin filaments, topology changes and two-phase flows *Journal of Computational Physics*, vol. 448, 2022.

Hybrid advection scheme WENO5 / HOUC5

- Efficient HOUC5 in smooth level-set region
- Robust WENO5 where the spatial discretization of the advection equation is subject to large error, i.e. where level-set is ill-defined.
- $\rightarrow$ CPU lowered with a factor up to 2.

F. Henri, M. Coquerelle, P. Lubin, *An efficient hybrid advection scheme in a level set framework coupling WENO5 and HOUC5 schemes based on kink detection*, accepted in Journal of Computational Physics.

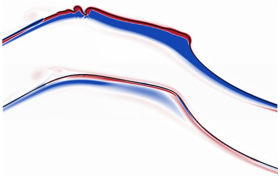
High order momentum preserving method

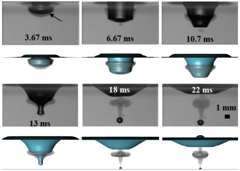
Context

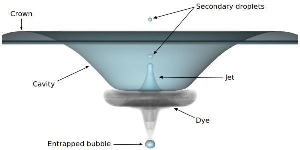
- Multiphase flows with high density ratio
- One-Fluid approach (VOF, MoF, Level-Set)
- Unconsistancy between mass and momentum flux can lead to instabilities and large errors

Solution

- Use the conservative momentum form
- Add a mass conservation equation to predict density for momentum equation
- Use of a synchronized temporal integration for the advective part of the momentum and mass equations.
- Use of consistent spatial conservative schemes for both
- $\rightarrow$ Discontinuity of the momentum is advected at the same speed as a discontinuity of the density
- $\rightarrow$ A high order approach (WENO5+RK2) proposed independent on the interface representation (VOF, MoF, Level-Set)



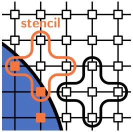


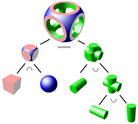
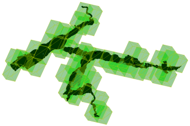


F. Desmons, M. Coquerelle, *A generalized high-order momentum preserving (HOMP) method in the one-fluid model for incompressible two phase flows with high density ratio*, Journal of Computational Physics, 437, 110322, 2021.

Presentation content

- The main features of Notus
- Short focus of some proposed numerical methods
- GENCI HPC challenge 2020 at TGCC

Continuous microfluidic process to generate nanoparticles
with A. Erriguible (I2M/ICMCB), S. Marre (ICMCB)

Macro-reactors *vs* micro-reactors. The scale-up challenge.

- Mixing of ethanol and supercritical CO2 in high pressure microfluidic system
  $\rightarrow$ Nucleation and growth of particles
- **Fast mixing reduces particle size**
  $\rightarrow$ **Turbulent flow in micro tubes (ICMCB experiments)**
  $\rightarrow$ **very good mixing (compare to larger reactor) and particle size around** $20nm$
- **Direct Numerical Simulation: all resolved mixing scale** (Kolmogorov $\approx$ Batchelor)
- **Limited production**. From lab to industrial scale? Sizing-up or numbering-up?
  $\rightarrow$ **Scale-up process, a pure numerical approach...**

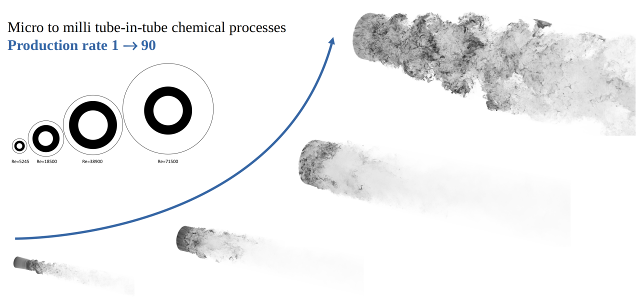
Large reactor (1cm). Injection velocity $3m.s^{-1}$, $R_{inj} = 90\mu m$, co-flow velocity $0.001m.s^{-1}$, $Re = 432$
Fully explicit (except pressure), 390.10^6 cells, 3584 processors

Micro reactor (0,3mm). Injection velocity $2.81m.s^{-1}$, $R_{inj} = 50\mu m$, co-flow velocity $3.97m.s^{-1}$, $Re = 5505$
Fully explicit (except pressure), 300.10^6 cells, 3584 processors

Process scale-up

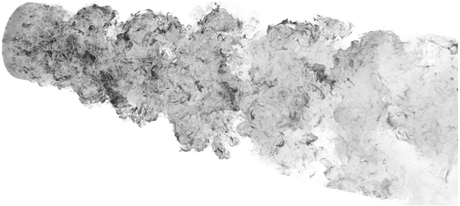
- Full explicit, Hybrid MPI/OpenMP version → **Improved strong and weak scalabilities**
- **From micro to milli tubes, TKE dissipation rate conserved**
- Re=5245 → **Re=71000**, all resolved mixing scale (DNS)
- GENCI HPC challenge (TGCC), 40 millions CPU hours
  from 300 million to 11 billion cells, up to 131 072 processors

→ **Still very fast mixing**
→ **Production x90**



S. Glockner, A.M.D. Jost, A. Erriguible, *Advanced petascale simulations of the scaling up of mixing limited flow processes for materials synthesis*, accepted in Chemical Engineering Journal.

HPC example: GENCI HPC challenge at TGCC



Concentration volume rendering, Kelvin-Helmholtz instabilities,
fast mixing, co-flow velocity effect, confinement effect