TRUST & TrioCFD V1.7.6
User's Training Session

TRUST/TrioCFD Training Sessions

- Users training session:
  - 1st day: TRUST/TrioCFD presentation & practices,
  - 2nd day: Automated test case with TRUST presentation & practices, mesh with tools (Internal tools, Salomé or Gmsh) presentation & practices, TRUST/TrioCFD practices.
- Developer training session:
  - 1st day: Basic OOC concepts used in TRUST, practices, Baltik project, Exploring the Kernel modules, How to debug TRUST,
  - 2nd day: Managing input/output files with TRUST classes, How to parallelize in TRUST, TRUST test coverage, TRUST coding rules
- Custom user session of one day to help on a specific problem

TRUST/TrioCFD support team
triou@cea.fr

Table of contents

Table of contents

- **TRUST/TrioCFD historic**
- Modeling flow with TRUST/TrioCFD
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- Parallel calculation
- Mesh generators: Salomé/Gmsh
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

TRUST/TrioCFD historic (1/4)

- **TRUST** platform:
  - CFD code
  - 3D/2D incompressible monophasic flows
  - diphasic flows through the Front-Tracking module of TrioCFD
  - developed at the CEA/DEN/DANS/DM2S/STMF service
  - platform used by other CEA applications (baltik projects)

- **TRUST** platform was born on June 2015
  - from the division of **Trio_U** software version 1.7.1 in two parts:

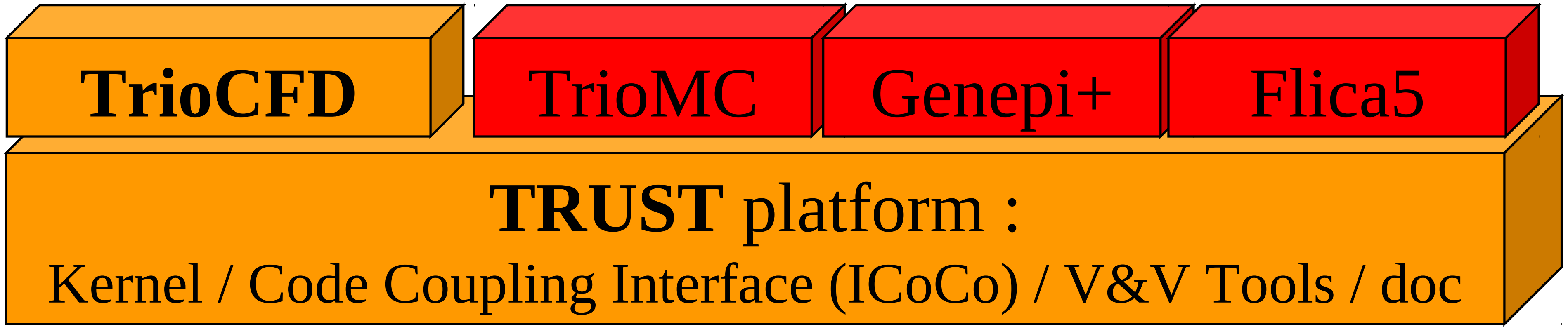
    Trio_U = TRUST + TrioCFD (FT, Radiation, LES, ...)
  - & switch to open source of **TRUST** and **TrioCFD**

- **TRUST** stands for: “**TR**io**_U S**oftware for **T**hermohydraulics”

TRUST/TrioCFD historic (2/4)

- **TrioCFD** is a **BALTIK** project of **TRUST**

- **BALTIK** stands for: “**B**uild an **A**pplication **L**inked to **Trio**_U **K**ernel”



TrioCFD

TrioMC

Genepi+

Flica5

TRUST platform :

Kernel / Code Coupling Interface (ICoCo) / V&V Tools / doc

- The **kernel** contains the equations, space discretizations, numerical schemes, parallelism...

TRUST/TrioCFD historic (3/4)

- 1994: start of the project **Trio_U**
- 01/1997 : v1.0 (VDF only)
- 06/1998 : v1.1 (VEF version)
- 04/2000 : v1.2 (parallel version)
- 07/2001 : v1.3 (radiation model)
- 11/2002 : v1.4 (new LES turbulence models)
- 02/2006 : v1.5 (VDF/VEF Front Tracking)
- 10/2009 : v1.6 (data structure revamped)
- 06/2015 : v1.7 (cut into **TRUST** & **TrioCFD** + switch to open source)

TRUST/TrioCFD historic (4/4)

- Main CEA goals:
  - R&D platform for fluid mechanics
  - To advocate LES turbulence models or RANS-LES coupling for nuclear safety studies
  - Codes coupling (fluid, structure, neutronic…)
  - Alternative to commercial CFD software
  - To base on advanced technology (C++, COO, //…)

Table of contents

- TRUST/TrioCFD historic
- **Modeling flow with TRUST/TrioCFD**
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- Parallel calculation
- Mesh generators: Internal tools & Salomé & Gmsh
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

Modeling flow with TRUST/TrioCFD (1/6)

Incompressible single phase flow

- Laminar or Turbulent flow
- Navier Stokes with or without energy equation
- Incompressible fluid or with low variation for volumic mass
  - Boussinesq hypothesis:

    $\rho=\rho(T)\sim \rho_0-\beta(T-T0)$
  - Quasi-compressible model:

    $\rho=\rho(P,T)$ for low mach numbers

$$\nabla\cdot\vec{u}=0$$

$$\frac{\partial\vec{u}}{\partial t}+\nabla\cdot\left(\vec{u}\otimes\vec{u}\right)=\nabla\cdot\left(\nu\nabla\vec{u}\right)-\nabla P^{*}$$

$$\frac{\partial T}{\partial t}+\vec{u}\nabla T=\nabla\cdot\left(\alpha\nabla T\right)+\frac{Q}{\rho C_{p}}$$

$$P^{*}=\frac{P}{\rho}+g\,z$$

Modeling flow with TRUST/TrioCFD (2/6)

Description of the Quasi Compressible model used

- Accounts for space and time variations of density due to high changes of temperature:

$$Ideal\ gas\ law : \rho(\vec{x},t) = \frac{P_0(t)}{RT(\vec{x},t)}$$

- Filters acoustic waves in order to avoid too small time-step:

$$Total\ pressure : P(\vec{x},t) = P_0(t) + P_1(\vec{x},t) \quad with\ \ P_1 \approx M^2 P_0 \ \ and\ M = Mach \ll 1$$

$$Thermodynamic\ \ pressure\ :\ P_0(t)$$
$$Hydrodynamic\ \ pressure\ :\ P_1(x,t)$$

- Set of equations solved:

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\vec{u}) = 0$$

$$\frac{\partial(\rho\vec{u})}{\partial t} + \nabla\cdot(\rho\vec{u}\vec{u}) = \nabla\cdot(\mu\nabla\vec{u}) - \nabla P_1^* - \rho\vec{g}$$

$$\rho C_p\left(\frac{\partial T}{\partial t} + \vec{u}\nabla T\right) = \nabla\cdot(\lambda\nabla T) + \frac{dP_0}{dt} + Q$$

$$P_0 = \rho RT$$

$$P_1^* = P_1 + \frac{2}{3}\mu\, div(\vec{u})$$

Modeling flow with TRUST/TrioCFD (3/6)

- Heat exchange
  - Conduction
  - Radiation in transparent medium
  - Radiation in semi-transparent medium
- Transport of passive scalars $\frac{\partial C_i}{\partial t} + \vec{u} \nabla C_i = Div\left(D_i \nabla C_i\right)$
- Porous Media
  - Surface or volume porosities
  - Singular or regular pressure loss

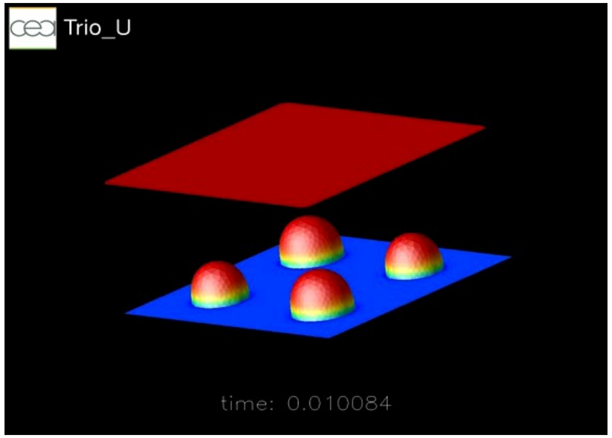
Modeling flow with TRUST/TrioCFD (4/6)

- Particles transport model:
  - One way coupling
    - Particle motion affected by the flow
  - Two way coupling
    - As above but particle disturbances also affect the flow
  - Possible to convert droplet/bubble below a given size into particles during a Front Tracking calculation

Modeling flow with TRUST/TrioCFD (5/6)

Front tracking model

- Two phases flow
  - Eulerian mesh where Navier Stokes equations are solved
  - Lagrangian moving mesh for the interface locations
  - Coalescence or breakup models for bubbles and drops



Boiling bubbles

- Can be declined in TrioCFD to use an Immersed Boundary Method using IBC (Immersed Boundary Conditions)

Modeling flow with TRUST/TrioCFD (6/6)

- 2D calculation
  - Plane, Cartesian coordinates (x,y)
  - Axi-symmetric, coordinates (r,z) (VDF only, k-eps OK)

- 3D calculation
  - Cartesian coordinates (x,y,z)

- Transient flow calculation calculated by:
  - Explicit, semi-implicit or implicit time schemes

- Steady state calculation obtained:
  - By convergence of the transient flow

Table of contents

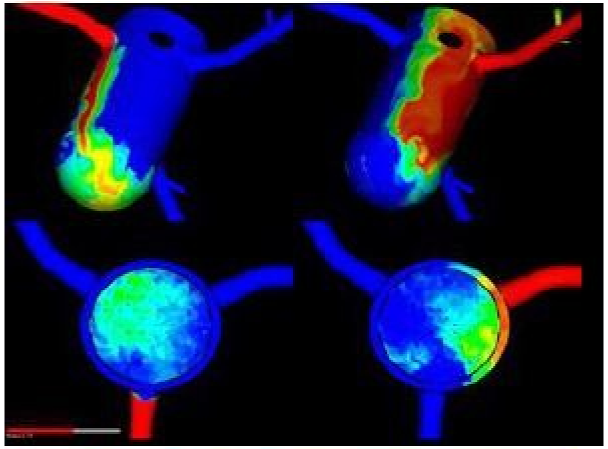
- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioCFD
- **Examples of performed calculations**
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- Parallel calculation
- Mesh generators: Internal tools & Salomé & Gmsh
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

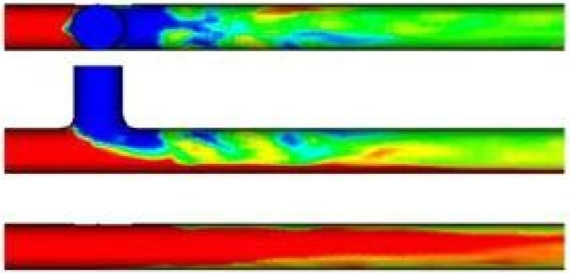
Examples of performed calculations (1/2)

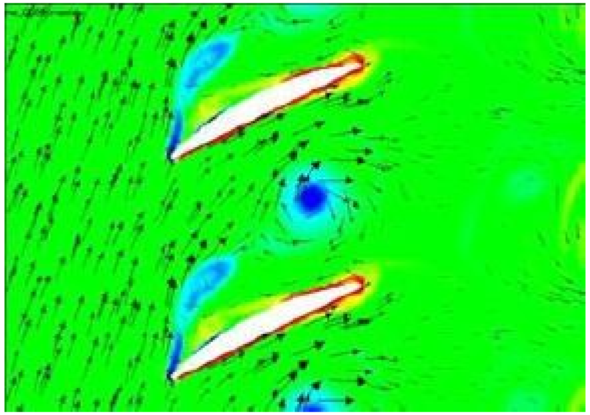
- Academic calculations
  - Plane channel with conduction coupling at the wall
  - Flow around obstacle
  - Pipe flow
  - Impinging jet
  - Isotropic turbulence

Examples of performed calculations (2/2)

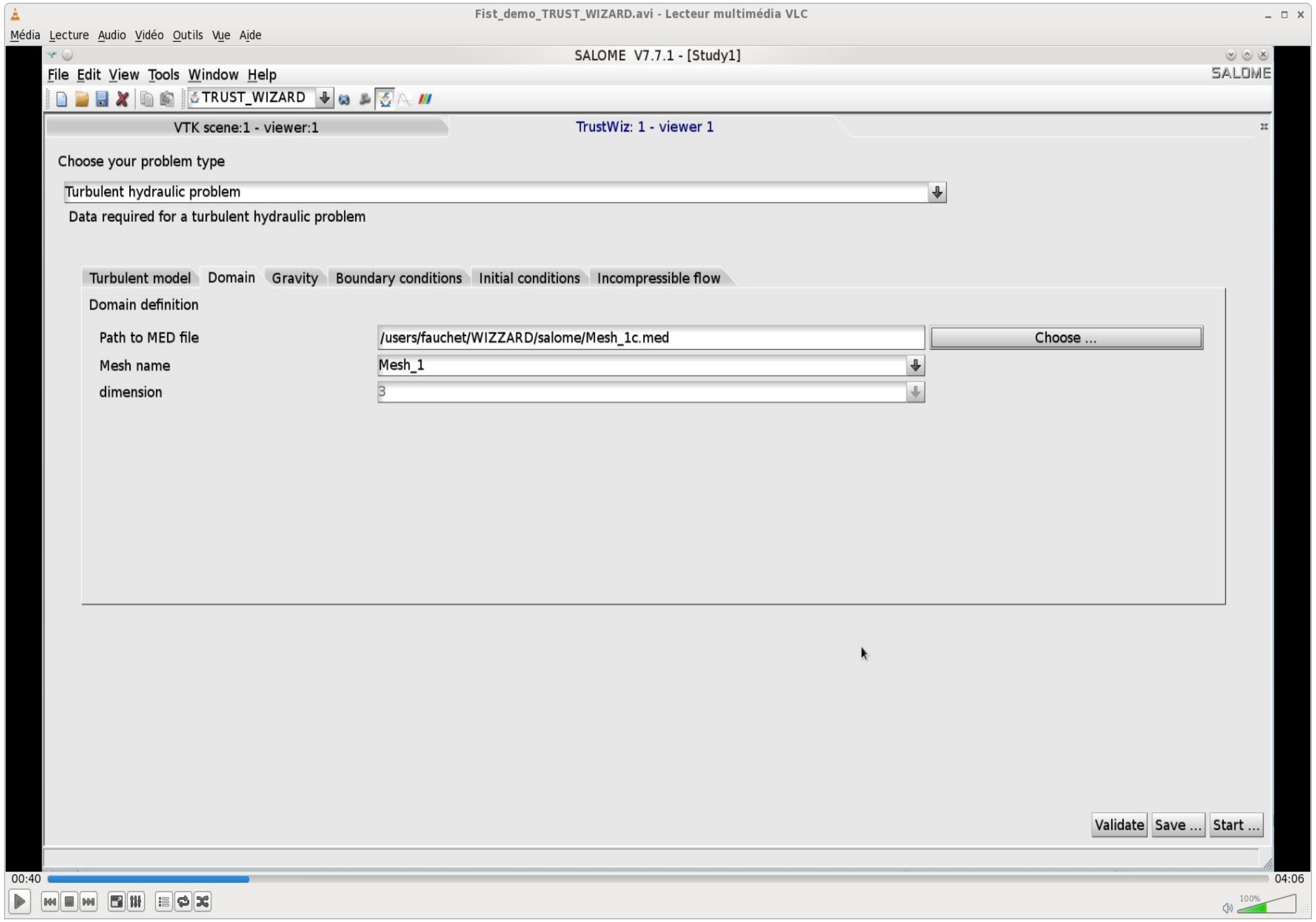
- Industrial calculations
  - Various studies about the core of a reactor
  - Thermal stress in a T-shaped mixing pipe
  - Natural convection in a storage room of waste
  - Atmospheric dispersion (polluting or radio-nucleid)
  - Compressor blades in GFR (Gas Fast Reactor)







IHM TRUST



Fist_demo_TRUST_WIZARD.avi - Lecteur multimédia VLC
Média Lecture Audio Vidéo Outils Vue Aide
SALOME V7.7.1 - [Study1]
File Edit View Tools Window Help
TRUST_WIZARD
VTK scene:1 - viewer:1
TrustWiz: 1 - viewer 1

Choose your problem type

Turbulent hydraulic problem

Data required for a turbulent hydraulic problem

Domain definition

Path to MED file: /users/fauchet/WIZZARD/salome/Mesh_1c.med

Mesh name: Mesh_1

dimension: 3

Practice

Using TRUST PLOT2D tools:

```
> source /home/triou/env_TRUST_X.Y.Z.sh
> echo $TRUST_ROOT
> mkdir -p ~/Formation_TRUST/yourname
> cd ~/Formation_TRUST/yourname
> trust -copy Obstacle
> cd Obstacle
> trust -evol Obstacle
```

- “Edit data”
- Substitute “lml” keyword to “lata”
- Save & close the file
- “Start computation!”: Wait until 100%
- Visualize a probe: select a probe and click on “Plot”
- Visualize a field with VisIt: “Visualisation”

Table of contents

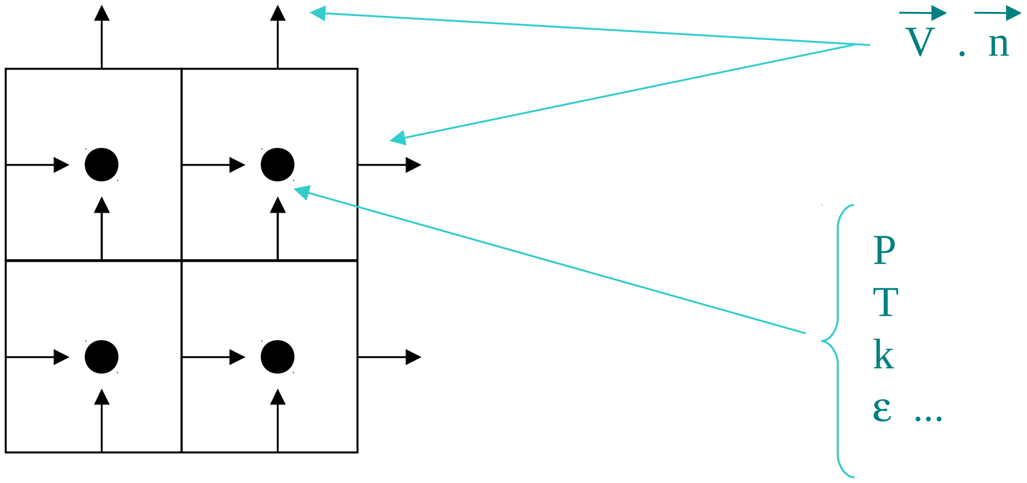
- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioCFD
- Examples of performed calculations
- **Models, schemes, numerical methods**
- Data files & calculation
- Command lines
- Parallel calculation
- Mesh generators: Internal tools & Salomé & Gmsh
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

Models, schemes, numerical methods (1/6)

- **Discretizations (VDF/VEF)..........**p22
- Time and space schemes.............p28
- Boundaries conditions...................p35
- Source terms................................p39
- Solvers for linear systems.............p43
- Turbulence models.......................p47

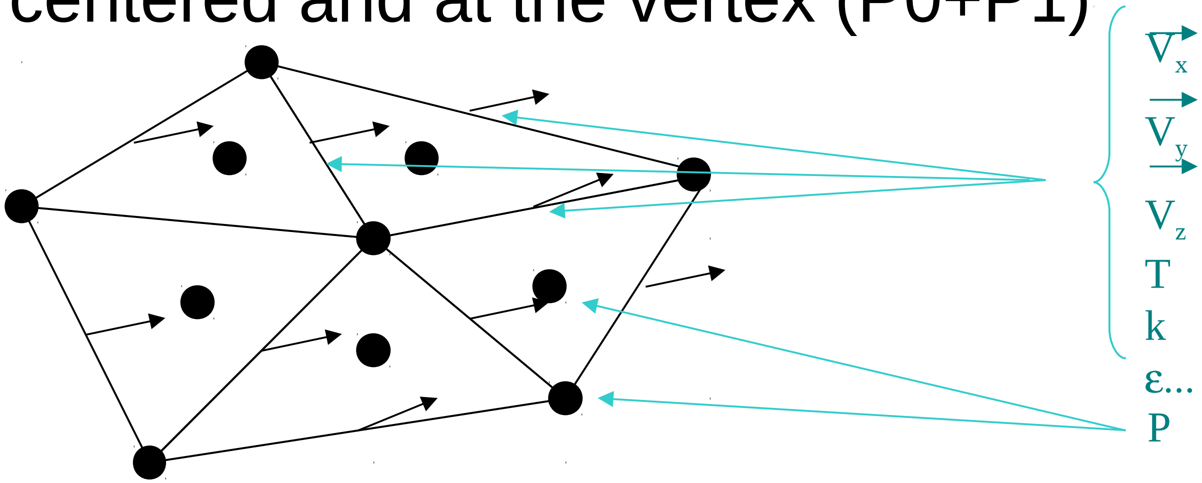
Discretizations (1/5)

- **F**inite **D**ifferences **V**olumes (**VDF**)
  - Structured meshing ; velocity and pressure are shifted
  - Scalar unknown is mesh centred
  - Velocity is normal at the face



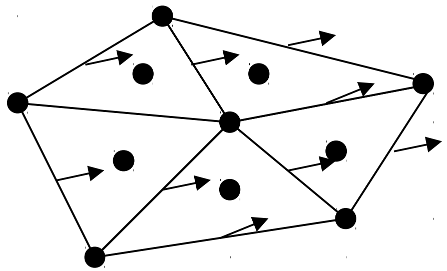
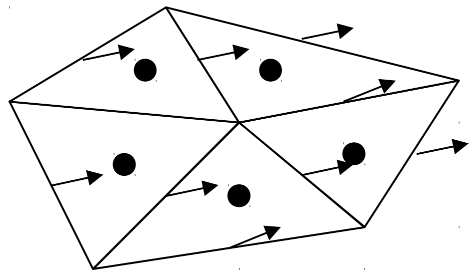
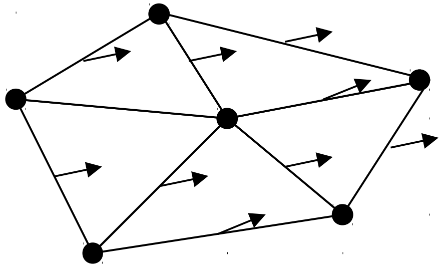
Discretizations (2/5)

- **F**inite **E**lements **V**olumes (**VEF**)
  - Unstructured meshing triangles (2D) or tetrahedrons (3D)
  - Unknown fields are face centered (P1NC)
  - Physical characteristics are cell centered
  - Pressure :
    - Mesh centered and at the vertex (P0+P1)

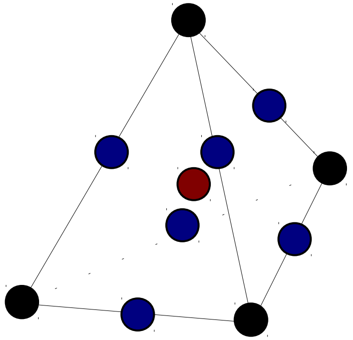


Discretizations (3/5)

- **Finite Elements Volumes (VEF)**
  - By default, P0+P1 for pressure but less/more pressure nodes is possible:

		
P0+P1	P0	P1

Plus in 3D: P0+P1+Pa



11 pressure nodes per tetra:

-1 in center (P0)

-4 on vertexes (P1)

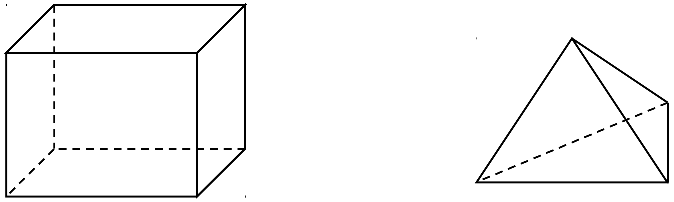
-6 on edges (Pa)

-> **Useful** for flow with a strong source term & a low velocity field where P0+P1 pressure gradient P0+P1 has trouble to match the source term

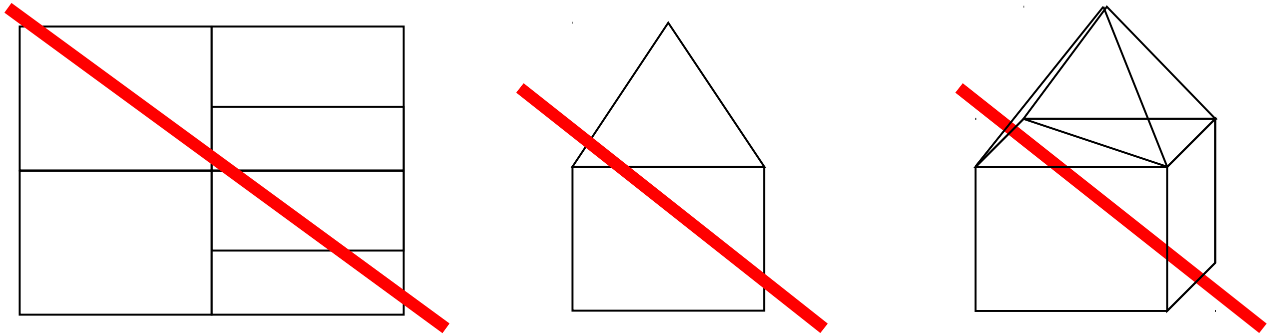
Discretizations (4/5)

- Kind of mesh permitted
  - Quadrangular or triangular meshing for 2D cases

  - Hexahedral or tetrahedral meshing for 3D cases

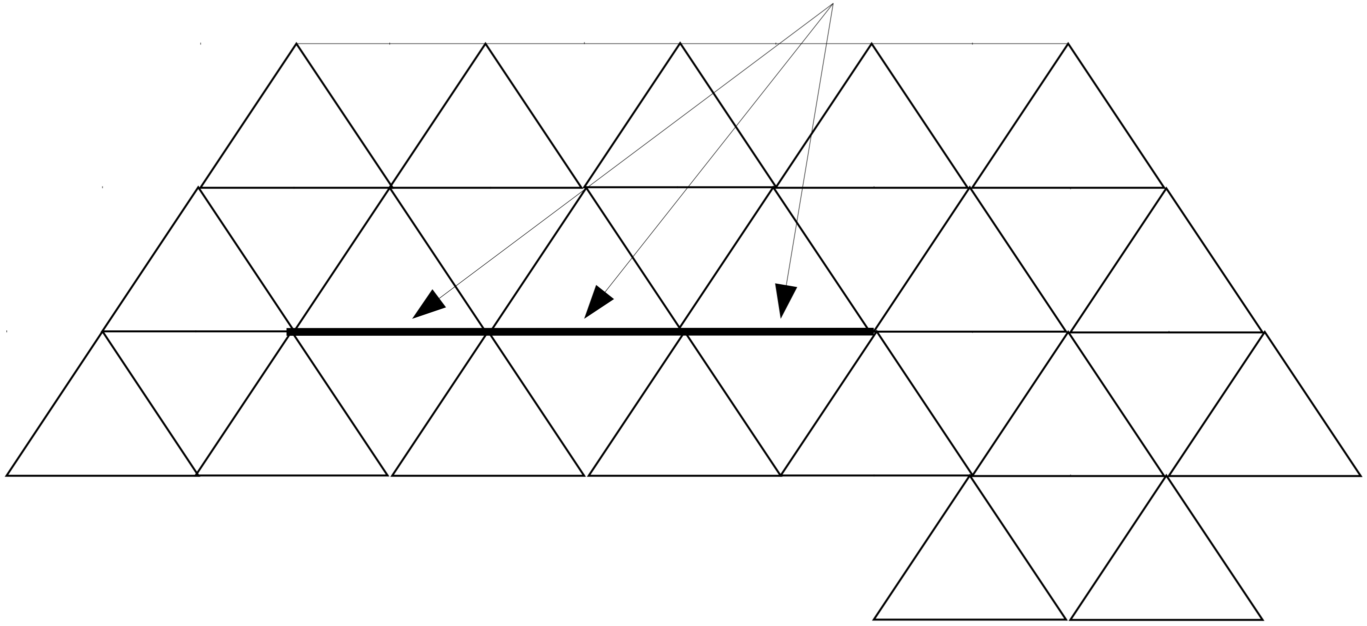


  - Non standard or hybrid meshing not supported



Discretizations (5/5)

- Kind of mesh permitted
  - Internal boundaries partially supported
    - with ICEM, spit faces in two and define boundaries
    - TRUST will differentiate the two faces



Models, schemes, numerical methods (2/6)

- Discretizations (VDF/VEF)
- **Time and space schemes**
- Boundaries conditions
- Source terms
- Solvers for linear systems
- Turbulence models

Some time and space schemes (1/6)

- Explicit time schemes:

$$\frac{\partial I^{n+1}}{\partial t}+\vec{u}^{n}\nabla I^{n}=Div\left(\alpha\nabla I^{n}\right)$$

  - Euler explicit (order 1)
  - Runge Kutta (order 2 or 3)

- Semi-implicit time scheme:

$$\frac{\partial I^{n+1}}{\partial t}+\vec{u}^{n}\nabla I^{n}=Div\left(\alpha\nabla I^{n+1}\right)$$

  - Euler semi-implicit (diffusion implicited)

- Implicit time schemes (not unconditionally stable):

$$\frac{\partial I^{n+1}}{\partial t}+\vec{u}^{n+1}\nabla I^{n+1}=Div\left(\alpha\nabla I^{n+1}\right)$$

  - Implicite, Piso, Simple (dynamic time step)

Some time and space schemes (2/6)

- Explicit time schemes:
  - For Navier-Stokes equations and incompressible flows:

$$(1)\nabla\cdot\vec{u}=0$$

$$(2)\frac{\partial\vec{u}}{\partial t}=\frac{-1}{\rho}\nabla P-\nabla\cdot(\vec{u}\otimes\vec{u})+\nabla\cdot(\nu\nabla\vec{u})$$

Convective term

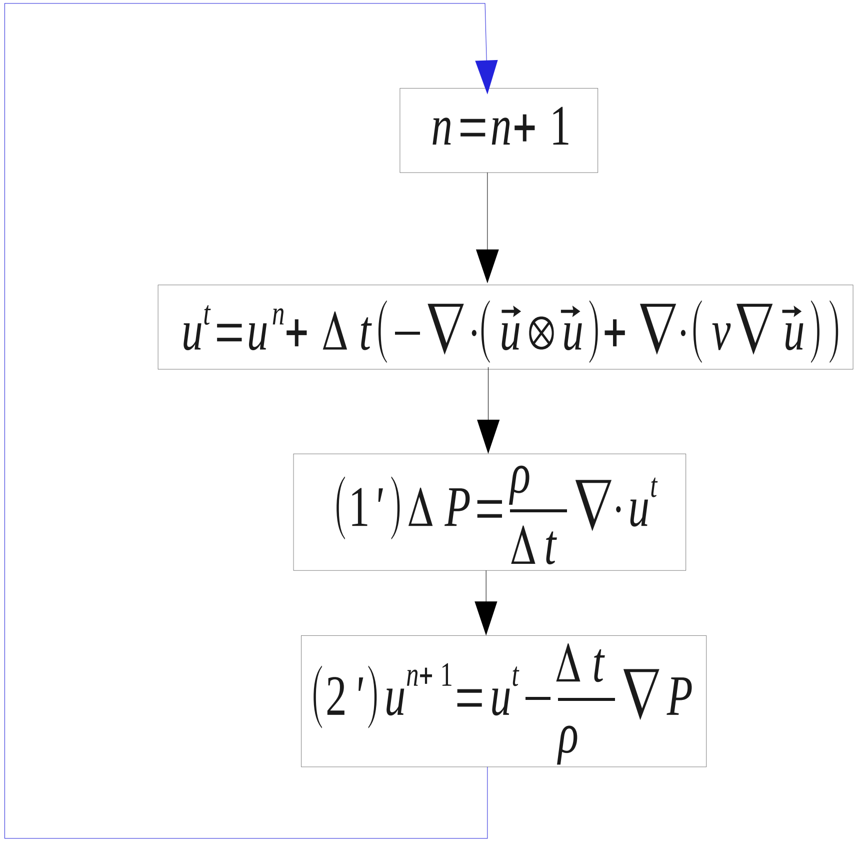
Diffusive term

  - Equation (2) leads to: $u^{n+1}=u^{n}+\Delta t\left(-\frac{1}{\rho}\nabla P-\nabla\cdot(\vec{u}\otimes\vec{u})+\nabla\cdot(\nu\nabla\vec{u})\right)$
  - We set: $u^{t}=u^{n}+\Delta t\left(-\nabla\cdot(\vec{u}\otimes\vec{u})+\nabla\cdot(\nu\nabla\vec{u})\right)$
  - So (2) becomes: $(2')\,u^{n+1}=u^{t}-\frac{\Delta t}{\rho}\nabla P$
  - Now using (1) into (2): $0=\nabla\cdot\left(\rho\frac{\partial\vec{u}}{\partial t}\right)=\nabla\cdot\left(-\nabla P-\rho\nabla\cdot(\vec{u}\otimes\vec{u})+\rho\nabla\cdot(\nu\nabla\vec{u})\right)$
  - So: $\Delta P=\rho\nabla\cdot\left(-\nabla\cdot(\vec{u}\otimes\vec{u})+\nabla\cdot(\nu\nabla\vec{u})\right)$
  - And we get:

$$(1')\,\Delta P=\frac{\rho}{\Delta t}\nabla\cdot u^{t}$$

Some time and space schemes (3/6)

- Explicit scheme: projection method



$$n=n+1$$

$$u^t=u^n+\Delta t\left(-\nabla\cdot\left(\vec{u}\otimes\vec{u}\right)+\nabla\cdot\left(\nu\nabla\vec{u}\right)\right)$$

Integrate with out pressure

$$(1')\,\Delta P=\frac{\rho}{\Delta t}\nabla\cdot u^t$$

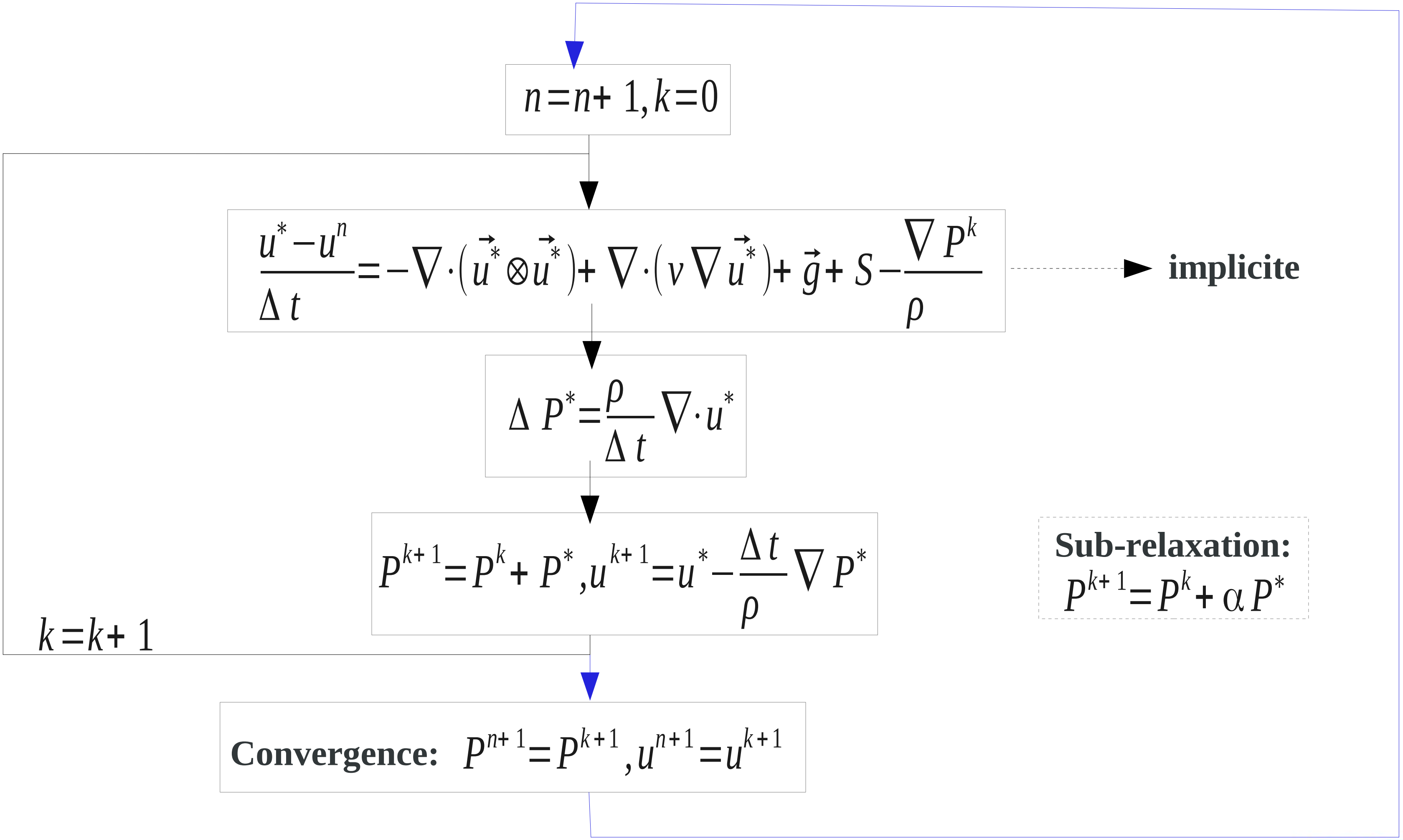
Poisson's equation

$$(2')\,u^{n+1}=u^t-\frac{\Delta t}{\rho}\nabla P$$

Projection on incompressible fields

Some time and space schemes (4/6)

- Semi-implicit scheme: SIMPLE



$$n=n+1, k=0$$

$$\frac{u^*-u^n}{\Delta t}=-\nabla\cdot\left(\vec{u}^*\otimes\vec{u}^*\right)+\nabla\cdot\left(\nu\nabla\vec{u}^*\right)+\vec{g}+S-\frac{\nabla P^k}{\rho}$$

implicite

$$\Delta P^*=\frac{\rho}{\Delta t}\nabla\cdot u^*$$

$$P^{k+1}=P^k+P^*, u^{k+1}=u^*-\frac{\Delta t}{\rho}\nabla P^*$$

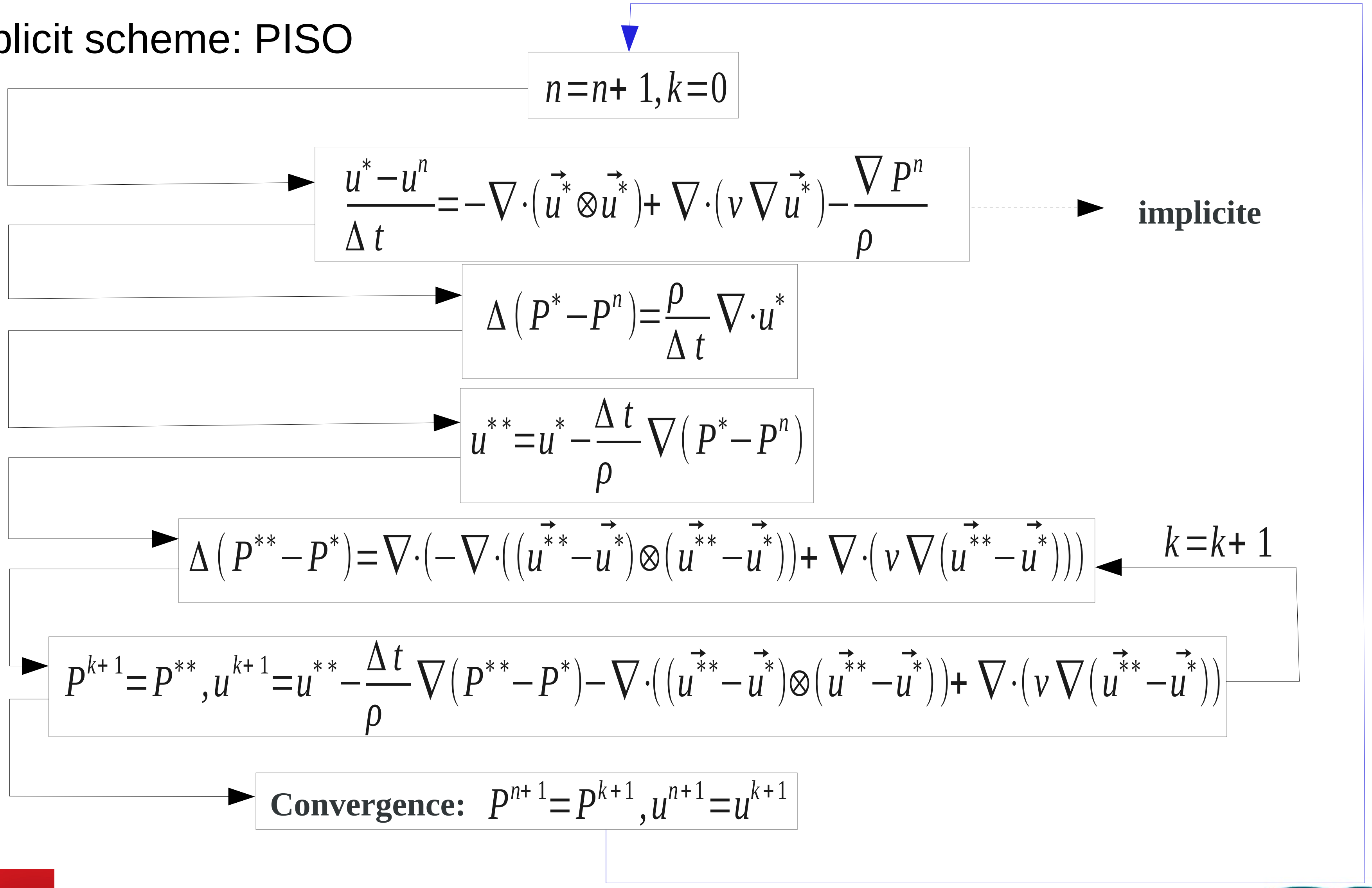
$$k=k+1$$

Sub-relaxation: $P^{k+1}=P^k+\alpha P^*$

Convergence: $P^{n+1}=P^{k+1}, u^{n+1}=u^{k+1}$

Some time and space schemes (5/6)

- Implicit scheme: PISO



$n=n+1, k=0$

$$\frac{u^*-u^n}{\Delta t}=-\nabla\cdot\left(\vec{u}^*\otimes\vec{u}^*\right)+\nabla\cdot\left(\nu\nabla\vec{u}^*\right)-\frac{\nabla P^n}{\rho}$$

implicite

$$\Delta\left(P^*-P^n\right)=\frac{\rho}{\Delta t}\nabla\cdot u^*$$

$$u^{**}=u^*-\frac{\Delta t}{\rho}\nabla\left(P^*-P^n\right)$$

$$\Delta\left(P^{**}-P^*\right)=\nabla\cdot\left(-\nabla\cdot\left(\left(\vec{u}^{**}-\vec{u}^*\right)\otimes\left(\vec{u}^{**}-\vec{u}^*\right)\right)+\nabla\cdot\left(\nu\nabla\left(\vec{u}^{**}-\vec{u}^*\right)\right)\right)$$

$k=k+1$

$$P^{k+1}=P^{**}, u^{k+1}=u^{**}-\frac{\Delta t}{\rho}\nabla\left(P^{**}-P^*\right)-\nabla\cdot\left(\left(\vec{u}^{**}-\vec{u}^*\right)\otimes\left(\vec{u}^{**}-\vec{u}^*\right)\right)+\nabla\cdot\left(\nu\nabla\left(\vec{u}^{**}-\vec{u}^*\right)\right)$$

Convergence: $P^{n+1}=P^{k+1}, u^{n+1}=u^{k+1}$

Some time and space schemes (6/6)

- convection schemes with VDF discretization
  - Quick (order 2-3)
  - Centre (order 2 or 4) « centered »
  - Amont (order 1) « upwind »

- convection schemes with VEF discretization
  - EF_stab (order 2) « centered stabilized »
  - Muscl (order 2) « quick like »
  - Amont (order 1) « upwind »

Models, schemes, numerical methods (3/6)

- Discretizations (VDF/VEF)
- Time and space schemes
- **Boundaries conditions**
- Source terms
- Solvers for linear systems
- Turbulence models

Boundaries conditions (1/3)

Available BC for momentum equation

- Wall:
  - No slip (u=0)
  - Slipping at the wall (u.n=0)
  - Imposed tangential velocity
- Fluid boundary:
  - Imposed velocity
  - Imposed pressure or pressure gradient
  - Periodic

Boundaries conditions (2/3)

Available BC for energy equation

- Wall:
  - Imposed temperature
  - Adiabatic or imposed flux
  - Imposed exchange (coefficient)
  - Contact resistance possible between two walls
- Fluid boundary:
  - Imposed temperature
  - No flux
  - Periodic

Boundaries conditions (3/3)

- Boundary condition values may be:
  - Uniform on the boundary
  - Space dependent
  - Time dependent
  - Read in a file

Models, schemes, numerical methods (4/6)

- Discretizations (VDF/VEF)
- Time and space schemes
- Boundaries conditions
- **Source terms**
- Solvers for linear systems
- Turbulence models

Source terms (1/3)

- Navier Stokes equation:
  - Boussinesq $S=\boldsymbol{\rho}_0 g \boldsymbol{\beta}(T-T_0)$
    - Useful for small variation of volumic mass
  - Flow rate $S=Q_m$
  - Pressure loss $S=-0.5\boldsymbol{\rho} C_f U|U|/D$
    - Regular pressure loss (Blasius or Cf given by the user)
  - Periodic channel $S=Q_m$
    - Useful to keep constant flow rate into a periodic channel
  - …

Source terms (2/3)

- Navier Stokes equation:

Calculation into a non Galilean referential R'. Coriolis and inertial forces, the user specifies:

- Acceleration and velocity of R' referential into the Galilean referential R

- Ω, dΩ/dt : rotation and its derivative term into the R' referential

- A centre of the rotation of R' into R with the coordinates given into the R' referential

$$\vec{F}_{ie} = -m\,\vec{a}_e = -m\,(\vec{a}(A)_{(R)} + \left(\frac{d\vec{\Omega}_{(R'/R)}}{dt}\right)_{(R)} \wedge \vec{AM} + \vec{\Omega}_{(R'/R)} \wedge (\vec{\Omega}_{(R'/R)} \wedge \vec{AM}))$$

$$\vec{F}_{ic} = -m\,\vec{a}_c = -m\;2\vec{\Omega}_{(R'/R)} \wedge \vec{v}_r$$

Source terms (3/3)

- Energy equation:
  - Volumic heat power $S=P$
    - For example into a solid media
  - ….
- Concentration equation:
  - Boussinesq $S=\boldsymbol{\rho_0} g \boldsymbol{\beta}(C-C_0)$
    - Useful to build a two miscible fluids calculation (if the volume mass is similar). Concentration C will be equal to the fraction fluid

Models, schemes, numerical methods (5/6)

- Discretizations (VDF/VEF)
- Time and space schemes
- Boundaries conditions
- Source terms
- **Solvers for linear systems**
- Turbulence models

Solvers for linear systems (1/3)

Linear systems	Sparse	Symmetric	Constant
Pressure linear system for incompressible flow	✗	✗	✗
Pressure linear system for quasi compressible flow	✗	✗	✗
Pressure linear system for diphasic flow	✗	✗	
Use of an implicit scheme	✗		
Radiation in transparent medium			✗
Radiation in semi transparent medium	✗	✗	✗

Solvers for linear systems (2/3)

- **TRUST solvers** (default choice):
  - Symmetric matrix (e.g. pressure solver)
    - GCP (Conjugate gradient with SSOR preconditioning)
  - Non symmetric matrix (e.g. implicit solver)
    - GMRES or Bi-CGSTAB with diagonal preconditioning

- Integrated **PETSc solvers** (advanced choice):
  - Symmetric matrix
    - Parallelized Cholesky:

      If enough memory available AND matrix is constant, generally the best choice under 500 cores (max 1000 cores on 10e6 cells, ~2s/timestep to solve Ax=B)
    - BI-CGSTAB with block jacobi ICC(1):

      The fastest if high scalability needed (>1000 cores), use GCP with block jacobi ICC(1) if BI-CGSTAB diverges. RCM ordering of the local matrix may accelerate also.
  - Non symmetric matrix
    - GMRES or BI-CGSTAB with diagonal preconditioning:

      Generally faster than previous TRUST versions

Solvers for linear systems (3/3)

- Iterative solvers (GC, GMRES,…)
  - Need a tolerance ε to be defined : ||Ax-B||<ε
  - Possible pitfall because it is an absolute (not a relative) value in TRUST
  - So, check the balances!
    - Exemple: Solving pressure system for an incompressible flow ⇔ Div(u)=0

      -So, check the flow rate error in .out file

- Direct solvers (PETSc Cholesky)
  - Use it if possible

Models, schemes, numerical methods (6/6)

- Discretizations (VDF/VEF)
- Time and space schemes
- Boundaries conditions
- Source terms
- Solvers for linear systems
- **Turbulence models**

Turbulence models (1/1)

- DNS (Direct numerical simulation)
  - No model

- RANS (Reynolds averaged Navier Stokes equations)
  - 0 equation
    - Mixing length
  - 2 equations
    - Standard k-ε

- LES (Large eddy simulation)
  - Wale http://www.cfd-online.com/Wiki/Wall-adapting_local_eddy-viscosity_(WALE)_model
  - Smagorinsky http://www.cfd-online.com/Wiki/Smagorinsky-Lilly_model

- Wall laws
  - Standard (logarithmic law)
  - TBLE (Turbulent Boundary Layer Equations)

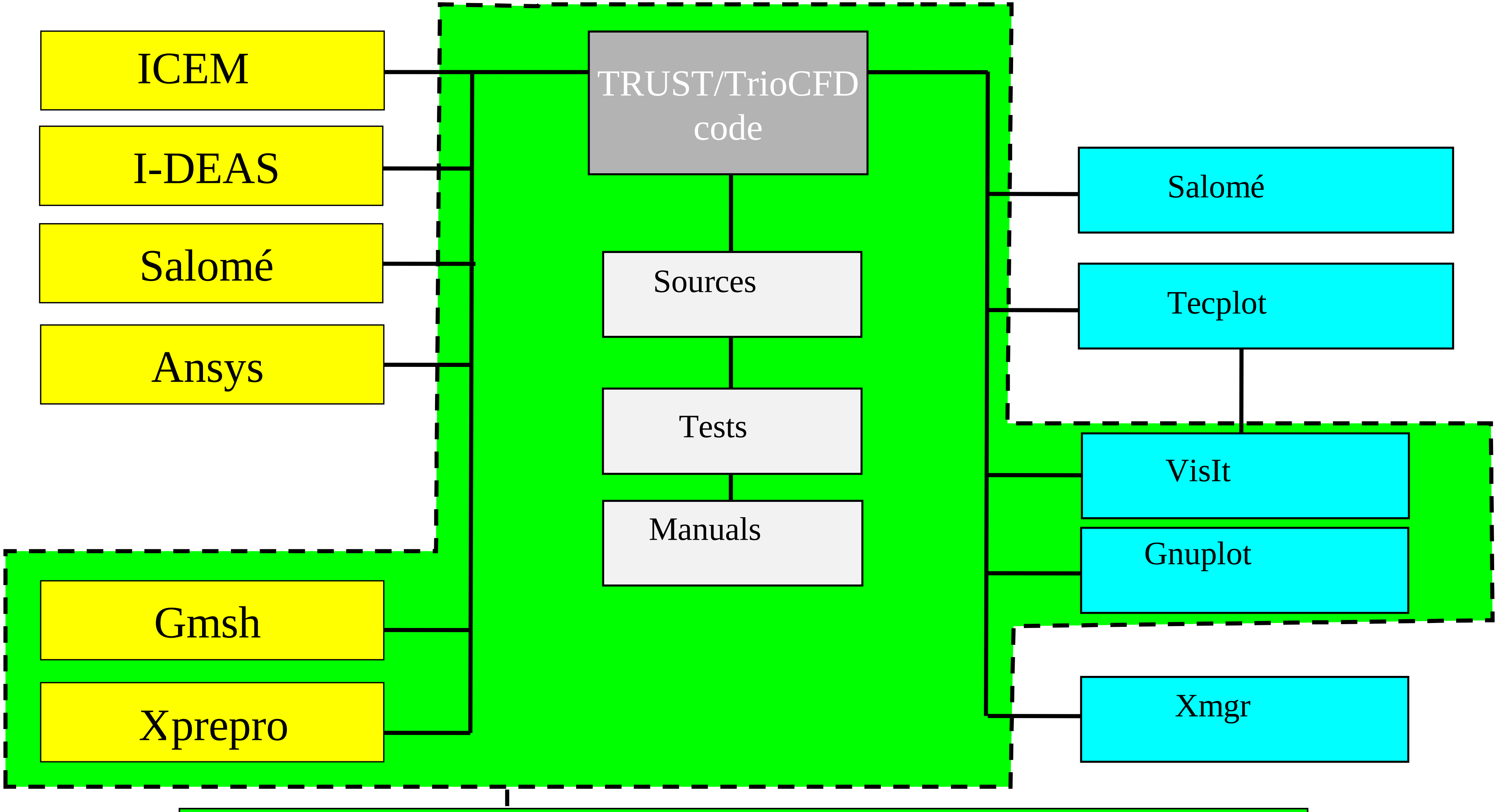
Table of contents

- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioCFD
- Examples of performed calculations
- Models, schemes, numerical methods
- **Data files & calculation**
- Command lines
- Parallel calculation
- Mesh generators: Internal tools & Salomé & Gmsh
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

Data files & calculation (1/5)

- **TRUST and tools interfaces.......**p50
- Data file description.......................p52
- Operations on meshes...................p62
- Post processing description..........p64
- Output files description..................p73

TRUST and interfaces between tools (1/1)



TRUST/TrioCFD package released and supported

Data files & calculation (2/5)

- TRUST and tools interfaces
- **Data file description**
- Operations on meshes
- Post processing description
- Output files description

Practice

Exercise: Obstacle.data + VisIt

(incompressible 2D flow)

Data file description (1/8)

- **Objects creation** with keywords (Domain, Time scheme, Problem, Fluid, …) :

# Data file objects definition #

Domaine my_domain

Schema_Euler_Explicite my_scheme

Pb_Thermohydraulique my_problem

Fluide_incompressible my_medium

…

Data file description (2/8)

- **Actions** on these objects with keywords:

```
Read_file my_domain meshing.geom  # Read a mesh file #
Read_file file.geo ;                # Read external instructions #
Read my_scheme {
        tinit  0.
        dt_min 0.001
        dt_max 0.002
        dt_impr 0.001
        …. }
Associate  my_problem  my_scheme        # Association #
Read  my_problem  { …. }          # Read (define) the problem #
Solve pb                          # The problem is solved with #
End                               # Not necessary keyword to finish #
```

Data file description (3/8)

Sequential data file example

```
# Hydraulique 2D laminar with Quick scheme #
# Dimension 2D or 3D #
dimension 2

# Domain definition #
Domaine dom

# BEGIN MESH #
/* Read mesh from ICEM */
/* Read_File_Binary dom mesh.bin */
/* Read mesh from Salome */
/* Read_MED dom Mesh_1 mesh.med */
/* Create domain and mesh from TRUST */
Read_file Obstacle.geo ;
# END MESH #

# Discretization on hexa or tetra mesh #
VDF ma_discretisation

# Time scheme explicit or implicit #
Scheme_euler_explicit mon_schema
Read mon_schema
{
        # Time step #
          # Initial time [s] #
        tinit 0
          # Min time step #
        dt_min 5.e-3
```

```
          # Max time step #
        dt_max 5.e-3 # dt_min=dt_max so dt imposed #
          # facsec such as dt = facsec * min(dt_stab , dt_max) ; #
          # Courant–Friedrichs–Lewy condition: dt_stab=1/(1/dt_convection+1/dt_diffusion) #
          # for explicit scheme facsec <= 1. By default facsec equals to 1 #
        # facsec 0.5 #
          # make the diffusion term in NS equation implicit : disable(0) or enable(1) #
        diffusion_implicite 0

        # Output criteria #
          # .out files printing period #
        dt_impr 5.e-3 # Note: small value to print at each time step #
          # .sauv files printing period #
        dt_sauv 100.
        periode_sauvegarde_securite_en_heures 23

        # Stop if one of the following criteria is checked: #
          # End time [s] #
        tmax 5.0
          # Max number of time steps #
        # nb_pas_dt_max 3 #
          # Convergence threshold (see .dt_ev file) #
        seuil_statio 1.e-8
}
```

Data file description (4/8)

```
# Problem definition #
Pb_hydraulique pb

# Physical characteristcs of medium #
Fluide_Incompressible milieu
Read milieu
{
        # hydraulic problem #
          # Dynamic viscosity [kg/m/s] #
        mu Champ_Uniforme 1 3.7e-05
          # Volumic mass [kg/m3] #
        rho Champ_Uniforme 1 2
}
```

```
# Association between the different objects #
Associate pb dom
Associate pb mon_schema
Associate pb milieu
Discretize pb ma_discretisation
```

```
# New domains for post-treatment #
# By default each boundarie condition of the domain is
          already extrated with such name
          "dom"_boundaries_"BC" #
```

```
# Problem description #
Read pb
{
        # hydraulic problem #
        Navier_Stokes_standard
        {
                # Pressure matrix solved with #
                solveur_pression GCP {
                        precond ssor { omega 1.500000 }
                        seuil 1.000000e-06
                        impr
                }
                # Two operators are defined #
                convection { quick }
                   # By default, 2nd order scheme #
                diffusion { }

                # Uniform initial condition for velocity #
                initial_conditions {
                        vitesse Champ_Uniforme 2 0. 0.
                        }
                # Boundary conditions #
                boundary_conditions {
                        Square     paroi_fixe
                        Upper      symetrie
                        Lower      symetrie
                        Outlet     frontiere_ouverte_pression_imposee Champ_front_Uniforme 1 0.
                        Inlet      frontiere_ouverte_vitesse_imposee Champ_front_Uniforme    2 1. 0.
                }
        }
...
```

$$\Delta P = \frac{\rho}{\Delta t} \nabla \cdot u^t$$

$$u^t = u^n + \Delta t \left(-\nabla \cdot \left(\vec{u} \otimes \vec{u} \right) + \nabla \cdot \left(\nu \nabla \vec{u} \right) \right)$$

TRUST/TrioCFD user's training session

Data file description (5/8)

List of possible keywords to define a field:

- Volume fields, keyword **Champ_TYPE** where TYPE may be:

  uniforme (uniform field)
  uniforme_par_morceaux (uniform field per sub-zone)
  fonc_t (uniform time dependent field)
  fonc_xyz (space dependent field)
  fonc_txyz (space and time dependent field)
  fonc_fonction (depends on another field, analytic function)
  fonc_tabule (depends on another field, tabulated function)
  fonc_MED (read a MED field)
  don_lu (field read in a file)

- Surface fields, keyword **Champ_front_TYPE** where TYPE:

  As volume fields plus:
  lu (field read in a file)
  recyclage (field extracted from a plane or a boundary of another problem)
  …

Data file description (6/8)

Formulas for a field in a data file:

Champ_front_fonc_txyz 2 t*(t_LE_100)+(t_GT_100)*100 0
Champ_fonc_xyz domain_name 2 **tanh**(4*y)*(0.95+0.1***rnd**(1)) 0.

The variables which can be used are:

x,y,z	: coordinates
t	: time

Constant or mathematical functions available:

PI, ABS, COS, SIN, TAN, ATAN, EXP, LN, SQRT, INT, ERF, RND(x), COSH, SINH, TANH
NOT(x), _AND_, _OR_, _GT_, _GE_, _LT_, _LE_

You can also use the following operations:

+	: addition
-	: substracte
/	: division
*	: multiplication
%	: modulo
$	: max
^	: power
<	: lesser than
>	: greater than
[	: less or equal to
]	: greater of equal to
( )	: test if

Data file description (7/8)

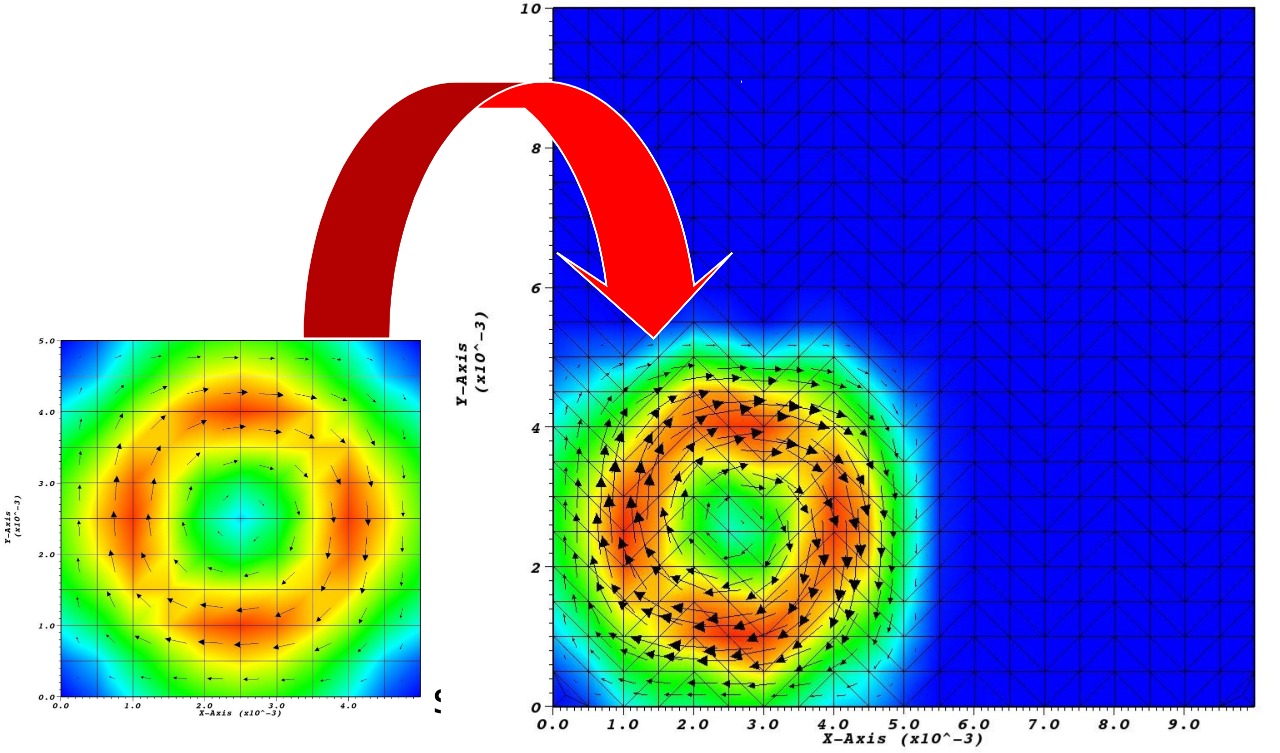
Example of Champ_Fonc_MED

First calculation on a VDF mesh:

Postraitement { **fichier** *VDF_field* **format med Champs dt_post** 0.1 { **vitesse elem** } }

Second calculation on a different refined VEF mesh with initial condition from the VDF field:

conditions_initiales { **vitesse Champ_Fonc_MED last_time** *VDF_field.med* domain **vitesse elem** 0 }

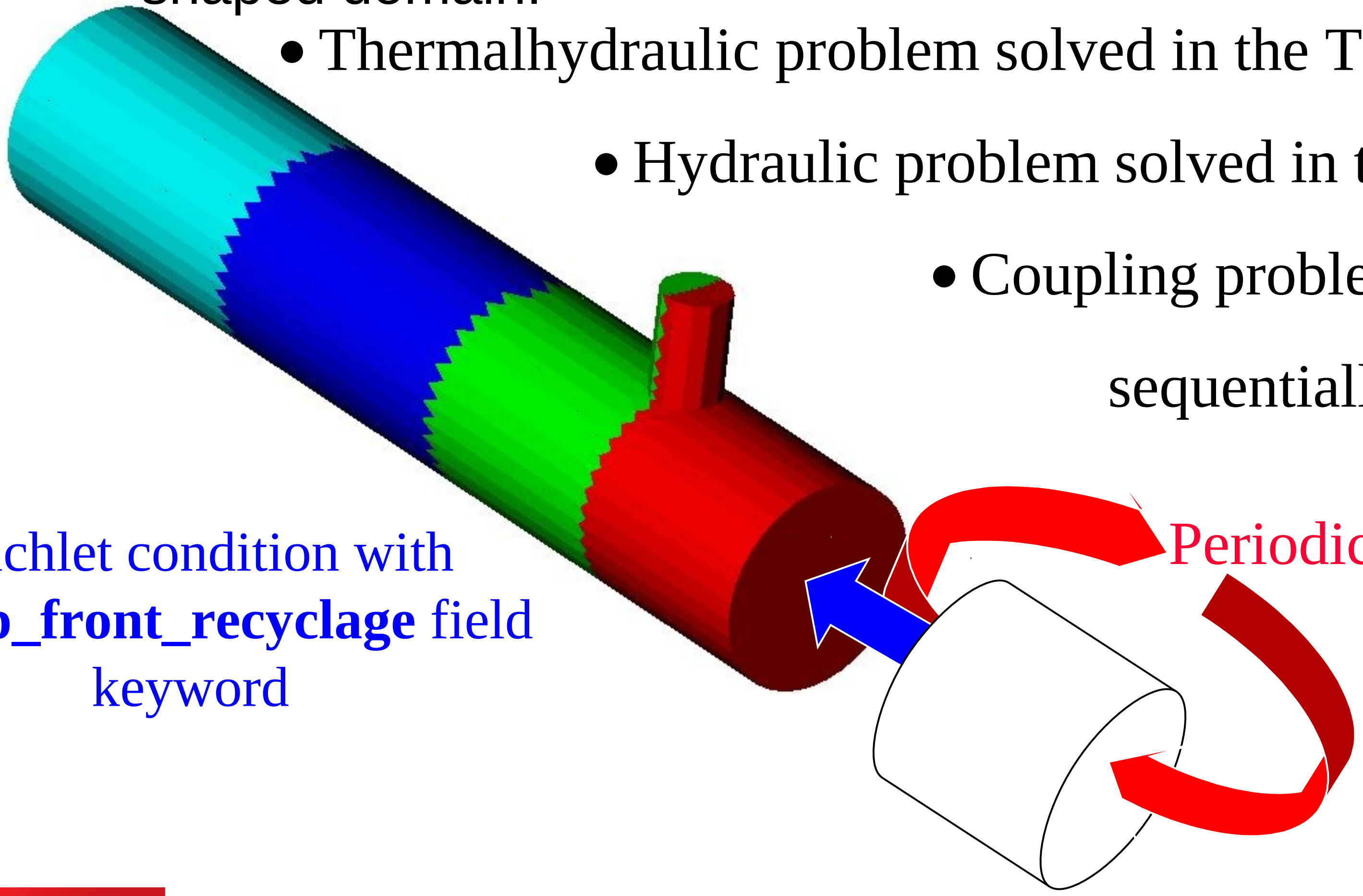


Data file description (8/8)

Example of TrioCFD coupled problems

Periodic box to provide a fully developed turbulent flow inlet for the T-shaped domain:

- Thermalhydraulic problem solved in the T-shaped domain.
- Hydraulic problem solved in the periodic box.
- Coupling problem which solves sequentially each problem.



Dirichlet condition with **Champ_front_recyclage** field keyword

Periodic boundary condition

cf "**Periodic 3D channel flow**" TRUST tutorial exercise

Data files & calculation (3/5)

- TRUST and tools interfaces
- Data file description
- **Operations on meshes**
- Post processing description
- Output files description

Operations on meshes (1/1)

- Keywords exist to modify your mesh after reading it (*.med, *.bin, *.geo, ...)
- List of possible keywords to adjust a mesh:
  - **Dilate** (to change the size of a mesh)
  - **Mailler** (to mesh a block or merge several meshes)
  - **Transformer** (to transform a mesh with a function)
  - **Rotation** (to rotate a mesh according to an axis)
  - **Extruder** (to extrude a 2D mesh into a 3D mesh)
  - **Trianguler**/**Tetraedriser** (to triangulate, to tetraedrise)
  - **Raffiner_(an)isotrope** (to refine a mesh)
  - **RegroupeBord** (to merge or rename boundaries)
  - **Supprime_Bord** (to suppress boundaries)
  - **Remove_Elem** (to create holes in a VDF mesh)
  - …

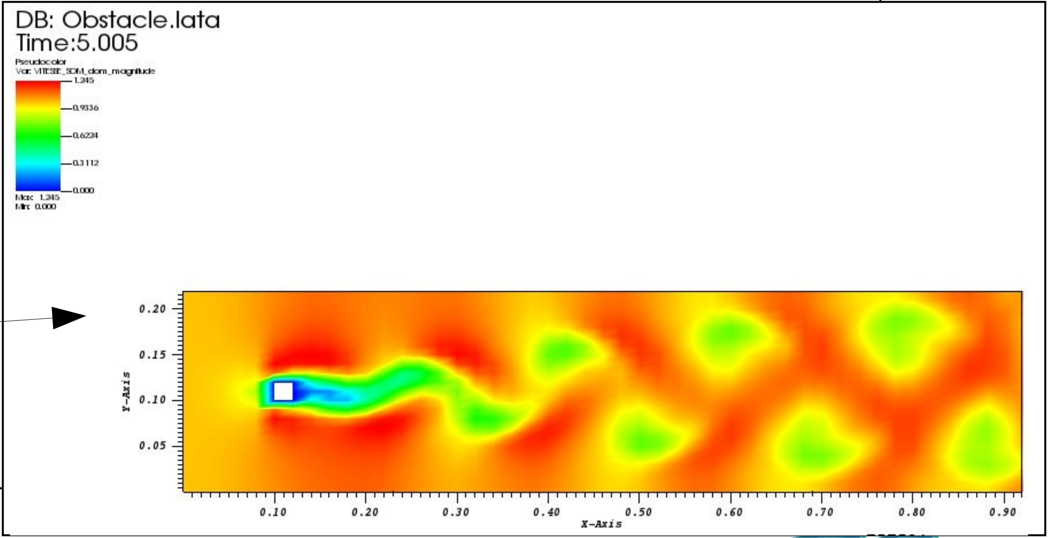
Data files & calculation (4/5)

- TRUST and tools interfaces
- Data file description
- Operations on meshes
- **Post processing description**
- Output files description

Data file description (1/8)

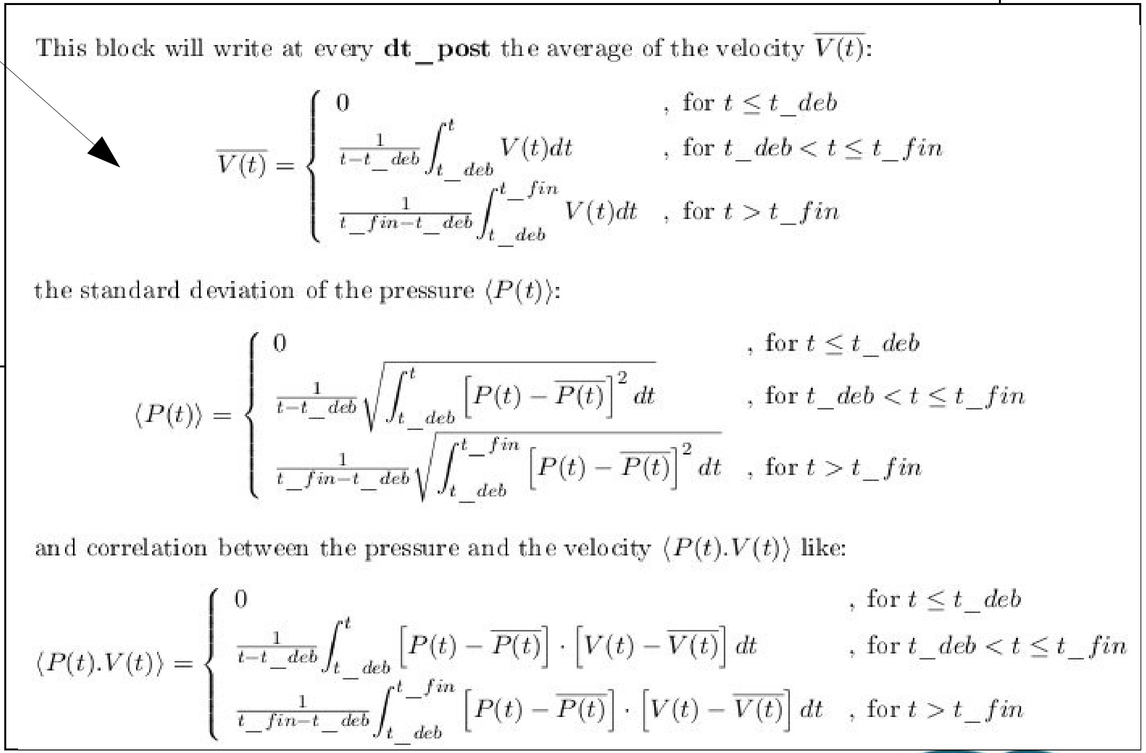
```
...
   # Post_processing description #
     /* To know domains that can be treated directly, search in .err output file:
        "Creating a surface domain named" */
     /* To know fields  that can be treated directly, search in .err output file:
        "Reading of fields to be postprocessed" */
     Post_processing
     {
          # Probes #
          Probes
          {
               # Note: periode with small value to print at each time step (necessary for spectral analysis) #
               sonde_pression          pression             periode 0.005  points 2    0.13 0.105   0.13 0.115
               sonde_vitesse           vitesse              periode 0.005  points 2    0.14 0.105   0.14 0.115
               sonde_vit        nodes  vitesse              periode 0.005  segment 22  0.14 0.0     0.14 0.22
               sonde_P                 pression             periode 0.01   plan 23 11  0.01 0.005   0.91 0.005   0.01 0.21
               sonde_Pmoy              Moyenne_pression     periode 0.005  points 2    0.13 0.105   0.13 0.115
               sonde_Pect              Ecart_type_pression  periode 0.005  points 2    0.13 0.105   0.13 0.115
          }


          # Fields #
          format lata # lata for VisIt tool #
          # Note: Warning to memory space if dt_post too small #
          fields dt_post 1.
          {
               pression elem
               pression som
               vitesse elem
               vitesse som
          }
```

```
# Obstacle_SONDE_PRESSION.son
# Temps x= 1.30000000e-01 y= 1.05000000e-01 x= 1.30000000e-01 y= 1.15000000e-01
# Champ PRESSION
# Type POINTS
0.00000000e+00 0.00000000e+00 0.00000000e+00
5.00000000e-03 -1.73139435e-01 -1.73139502e-01
1.00000000e-02 -1.80250693e-01 -1.80250216e-01
1.50000000e-02 -1.86369497e-01 -1.86371144e-01
2.00000000e-02 -1.86756785e-01 -1.86754184e-01
2.50000000e-02 -1.88558992e-01 -1.88562414e-01
```



Data file description (2/8)

```
                # Statistical fields #
                Statistiques dt_post 1.
                {
                        t_deb 1. t_fin 5.
                        moyenne vitesse
                        ecart_type vitesse
                        moyenne pression
                        ecart_type pression
                }
        }

        # Saving and restarting process #
        /* sauvegarde_simple binaire datafile.sauv */
        # Note: last time step only saved #
        /* resume_last_time binaire datafile.sauv */
}

# The problem is solved with #
Solve pb
# Not necessary keyword to finish #
End
```



This block will write at every **dt_post** the average of the velocity $\overline{V(t)}$:

$$\overline{V(t)} = \begin{cases} 0 & , \text{ for } t \le t\_deb \\ \frac{1}{t-t\_deb}\int_{t\_deb}^{t} V(t)dt & , \text{ for } t\_deb < t \le t\_fin \\ \frac{1}{t\_fin-t\_deb}\int_{t\_deb}^{t\_fin} V(t)dt & , \text{ for } t > t\_fin \end{cases}$$

the standard deviation of the pressure $\langle P(t) \rangle$:

$$\langle P(t) \rangle = \begin{cases} 0 & , \text{ for } t \le t\_deb \\ \frac{1}{t-t\_deb}\sqrt{\int_{t\_deb}^{t} \left[P(t)-\overline{P(t)}\right]^2 dt} & , \text{ for } t\_deb < t \le t\_fin \\ \frac{1}{t\_fin-t\_deb}\sqrt{\int_{t\_deb}^{t\_fin} \left[P(t)-\overline{P(t)}\right]^2 dt} & , \text{ for } t > t\_fin \end{cases}$$

and correlation between the pressure and the velocity $\langle P(t).V(t) \rangle$ like:

$$\langle P(t).V(t) \rangle = \begin{cases} 0 & , \text{ for } t \le t\_deb \\ \frac{1}{t-t\_deb}\int_{t\_deb}^{t} \left[P(t)-\overline{P(t)}\right] \cdot \left[V(t)-\overline{V(t)}\right] dt & , \text{ for } t\_deb < t \le t\_fin \\ \frac{1}{t\_fin-t\_deb}\int_{t\_deb}^{t\_fin} \left[P(t)-\overline{P(t)}\right] \cdot \left[V(t)-\overline{V(t)}\right] dt & , \text{ for } t > t\_fin \end{cases}$$

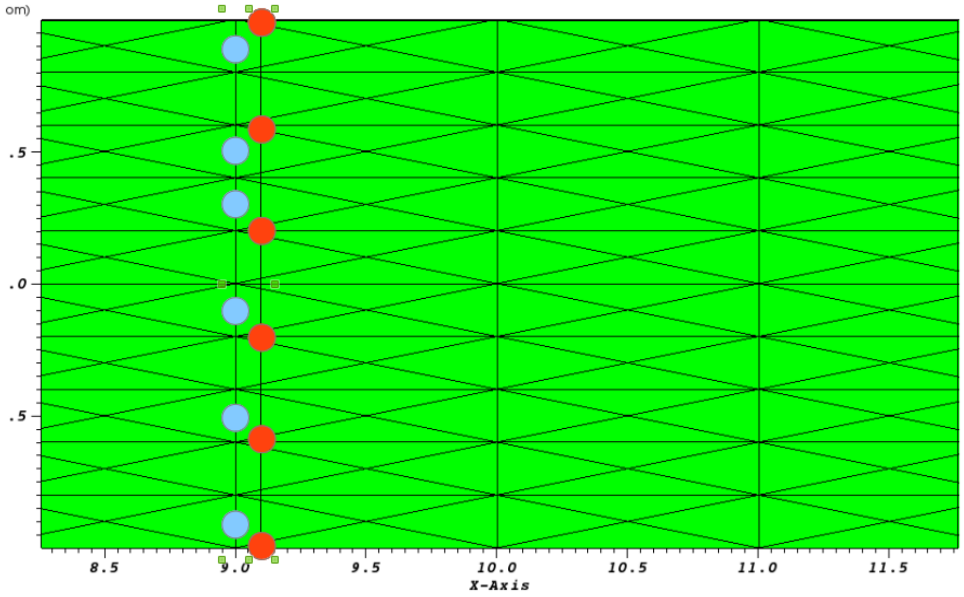
Post processing description (3/8)

Probes : “Nodes” option

"**Nodes**" option moves the probes to the nearest faces, but take care of stretched meshes!

```
sonde_vit   nodes   vitesse   periode   0.005   segment   22   0.14   0.0   0.14   0.22
```

In the following 2D example, the initial probes in **red** are defined along a segment from boundary to boundary, but when applying option "**nodes**" it moves the probes (in **blue**) to the nearest face and sometimes near the boundary the nearest face **IS NOT** a boundary face.



Since several version it is possible to visualize the .son files containing the probes in Visit. It provides you a point MESH which is the localization of the probes. We will try in the 1.6.9 version to help user by improving the messages in the .err file when probes are moved according to the "nodes" option and also to run VisIt on .son files directly from the triou script.

So, if you want extreme probes of the segment on the boundaries, try to move slightly the segment.

Post processing description (4/8)

TRUST results export

2D/3D results files are readable:

- Either directly by :
  - VisIt (use *lata* format <u>in the data file</u> “**format lata**”)*
  - Salomé (use *med* format <u>in the data file</u> “**format med**”):
    - to post-process with Salomé: use ParaVis module → New study → File → Open Paraview File,
    - to open a mesh in Salomé: use Mesh module → New study → File → Import → MED File
- Or:
  - Read with VisIt and then export (VTK format, field by field): Paraview
  - Using lata2dx script (lata_to_med, lata_to_case): for example Tecplot

1D results files by :

- Gnuplot, XmGrace, Excel

*: VisIt, thanks to a new plugin released with Trio_U 1.6.9 reads also the MED format

Post processing description (5/8)

- Possible basic post processed fields
  - Equation unknowns (velocity, pressure, temperature,…)
  - Physical characteristics (dynamic viscosity, thermal conductivity,…)
  - Model fields (turbulent viscosity, friction velocity,…)

  The complete list of keywords for fields which could be post processed for the current calculation is printed into the .err file :

```
Reading of fields to be postprocessed
Milieu_base : 1 masse_volumique
Fluide_Incompressible : 2 viscosite_cinematique viscosite_dynamique
Equation_base : 1 volume_maille
Operateur_base : 0
Navier_Stokes_std : 16 divergence_U gradient_pressionY gradient_pressionX gradient_pression pression_pa
pression vitesseY vitesseX vitesse taux_cisaillement courant_maille reynolds_maille y_plus porosite_volumique
critere_Q vorticite
```

- Possible advanced post processing fields
  - Can be created in the data file with the keyword **Definition_champs**
  - 3 examples given:
    - Monitoring extreme values of a field
    - Averaging a field on a boundary
    - Error between TRUST and an analytical solution

Post processing description (6/8)

Maximal value of a field

```
Read pb {
  ….
 Post_processing {
          Definition_champs {
                    # Creation of the 0D field: maximal temperature of the domain #
                    temperature_max Reduction_0D {
                              methode max
                              source refChamp { Pb_champ pb temperature }
                    }
          }
          Probes {
                    # Print max(temperature) into the datafile_TMAX.son file #
                    tmax temperature_max periode 0.01 point 1 0.5  1. # or numero_elem_sur_maitre 0 #
                    temp temperature       periode 0.01 point 1 0.5  1.
          }
  }
}
```

Post processing description (7/8)

Averaging a field on a surface

```
Dimension 3
Domaine dom # 3D domain with a boundary named Inlet #
Domaine plan1 # 2D domain object created for use by the Interpolation keyword or use DomainName_boundaries_BoundaryName in the new field #
Extraire_surface { Domaine plan_1 Probleme pb condition_elements (x<1.) }
...
Read problem {    …
Post_processing {
        Definition_champs { # Creation of 0D fields: mean temperature on a boundary or on a new surface #
                Inlet_mean_temperature Reduction_0D {
                        methode moyenne
                        source Interpolation {   domaine dom_boundaries_Inlet localisation elem
                                                 source refChamp { Pb_champ problem temperature } } }
                Plan1_mean_temperature Reduction_0D {
                        methode moyenne
                        source Interpolation { domaine plan1 localisation elem
                                               source refChamp { Pb_champ problem temperature } } }
        }
        # Print into the datafile_probename.son file #
        Probes {  tinlet Inlet_mean_temperature periode 0.01 point 1 0. 0.  0.
                  tplan1 Plan1_mean_temperature periode 0.01 point 1 0. 0.  0.  }
        }
}
```

Post processing description (8/8)

Calculating an error between fields

```
Post_processing {
Definition_champs { # Creation of the 3D  field: error #
        error Transformation {
                methode formule expression 1 vit-sol
                 localisation faces
                 sources {
                        refChamp { Pb_champ problem vitesse nom_source vit } ,
                        Transformation { methode vecteur expression  3  x*y  x+y  z  nom_source sol localisation faces }
                  }
        }
        # Calculate the L2 norm of the error field as 0D field for each component #
        error_norm Reduction_0D { methode norme_L2  source_reference  error }
        # Print into the datafile_ERROR_NORM.son file #
        Probes { sonde_error_norm error_norm  periode 0.01 point 1 0. 0.  0.  }
        format LATA Fields dt_post 1.0 {
                error           # Post process the error field #
                velocity
        }
}
```

Data files & calculation (5/5)

- TRUST and tools interfaces
- Data file description
- Operations on meshes
- Post processing description
- **Output files description**

Output files description (1/6)

Saving process

- Unknowns (velocity, temperature,…) are saved in:

- one *.xyz* file
- one or several (parallel calculation) *.sauv* files

- By default, saving process in *.sauv* happens during the calculation:

- At the start and at the end
- Periodically (each 23 hours of CPU with **tcpumax** keyword)
- But, user may also specify a time physical period (**dt_sauv** keyword)

- By default, saving process in *.xyz* file happens during the calculation:

- At the end
- But, user may enable it with the specific keyword “**EcritureLectureSpecial 0**”

- By default, there is a default name for the *.sauv* files (testcase_000n.sauv), the format is binary, and the files are appended during successive saves but user can change the behaviour with the keywords:

sauvegarde|sauvegarde_simple binaire|xyz *filename.sauv|filename.xyz*

with: - **sauvegarde_simple** : the file is deleted before the save

- **xyz**: the *.xyz* file is written instead of the *.sauv* files

Output files description (2/6)

Restarting process

Restarting the calculation is possible :

- Either from *.sauv* file(s) (one file per process)

-> Necessary to restart the calculation with the same number of equations on the same number of processes

- Or from a *.xyz* file

-> Possible to restart a calculation by changing the number of equations solved

-> Possible to restart with a different number of processes

The mandatory syntax in the data file is:

reprise binaire|xyz *filename.sauv|filename.xyz* with **tinit** updated
or
resume_last_time binaire|xyz *filename.sauv|filename.xyz*

Output files description (3/6)

TRUST files summary

- Input:

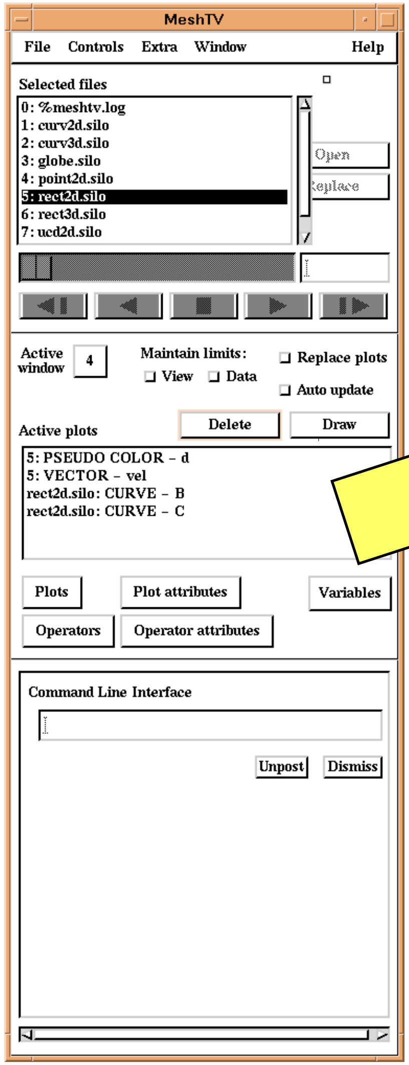
Data file:	.data
Meshing:	.geom (or .bin)
Instructions file:	.geo
Sub zones:	.ssz
Sub domains:	.Zones

- Output :

2D/3D results:	.lata (or .med)
1D results:	.son
Saving-restart:	.sauv ou .xyz
Listing (physical infos):	.out
Listing (warnings&errors&domain infos):	.err
Listing of boundary fluxes:	*.out
Canal_perio outputs:	*"BCname"
CPU performances:	.TU
Time steps, facsec, equation residuals:	.dt_ev
Stop file (0 or 1):	.stop

Output files description (4/6)

TRUST with VisIt



MeshTV

File Controls Extra Window Help

Selected files

0: %meshtv.log
1: curv2d.silo
2: curv3d.silo
3: globe.silo
4: point2d.silo
5: rect2d.silo
6: rect3d.silo
7: ucd2d.silo

Open

Replace

Active window 4

Maintain limits: □ View □ Data

□ Replace plots

□ Auto update

Active plots

Delete

Draw

5: PSEUDO COLOR – d
5: VECTOR – vel
rect2d.silo: CURVE – B
rect2d.silo: CURVE – C

Plots

Plot attributes

Variables

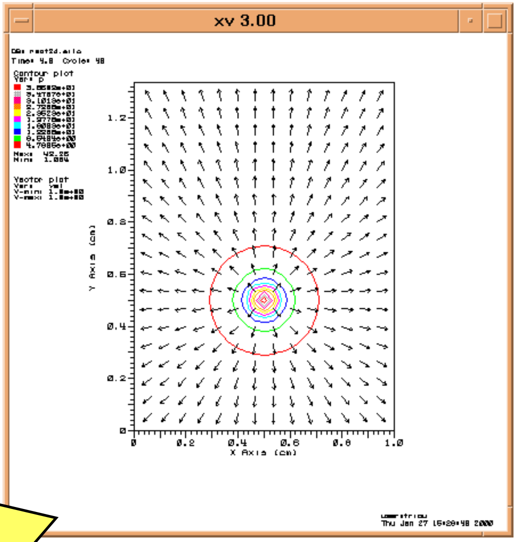
Operators

Operator attributes

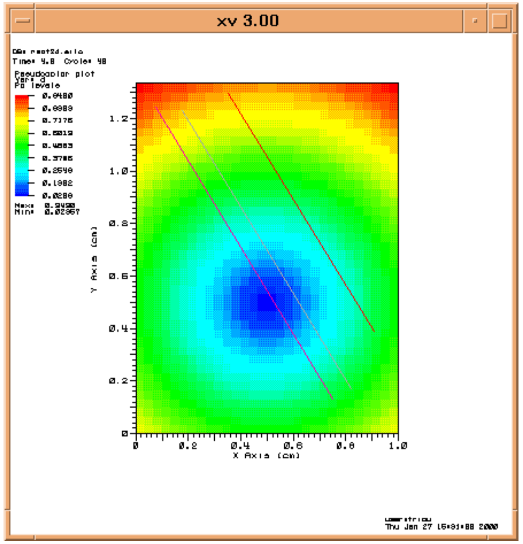
Command Line Interface

Unpost

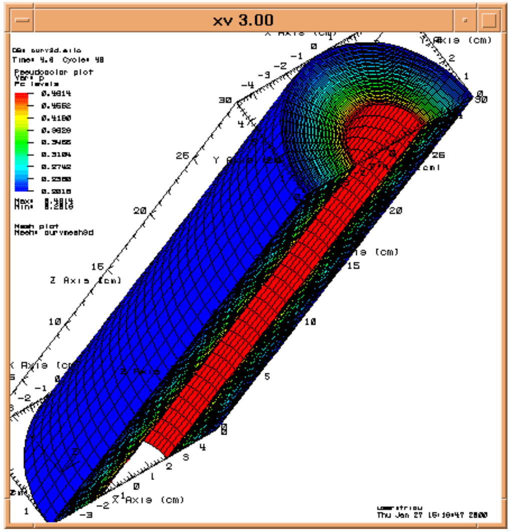
Dismiss



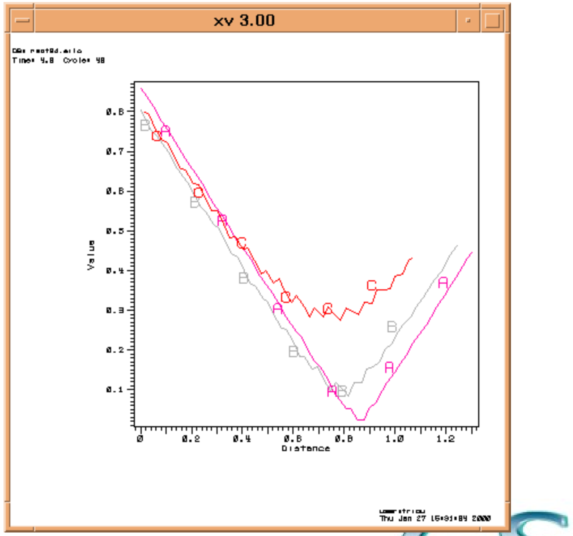
Vectors/Scalars



2D Slice



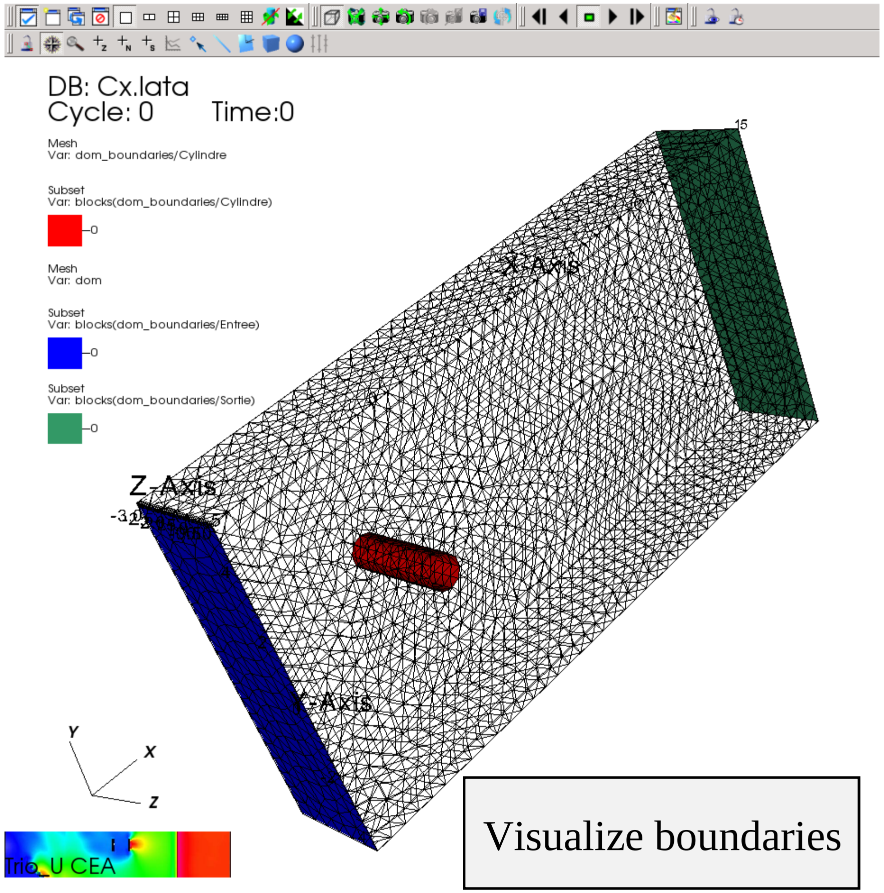
3D View



1D Cut

Output files description (5/6)

TRUST with VisIt



DB: Cx.lata
Cycle: 0 Time:0

Mesh
Var: dom_boundaries/Cylindre

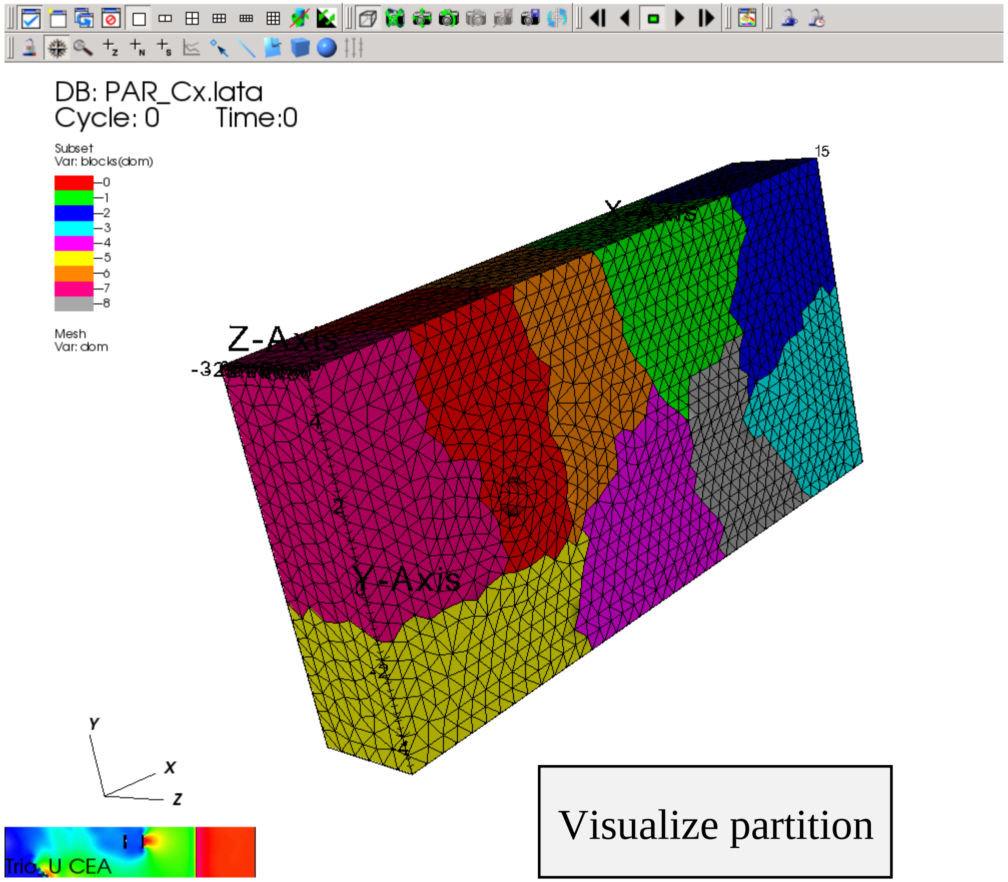
Subset
Var: blocks(dom_boundaries/Cylindre)
🟥 0

Mesh
Var: dom

Subset
Var: blocks(dom_boundaries/Entree)
🟦 0

Subset
Var: blocks(dom_boundaries/Sortie)
🟩 0

Visualize boundaries



DB: PAR_Cx.lata
Cycle: 0 Time:0

Subset
Var: blocks(dom)
🟥 0
🟩 1
🟦 2
3
4
5
6
7
8

Mesh
Var: dom

Visualize partition

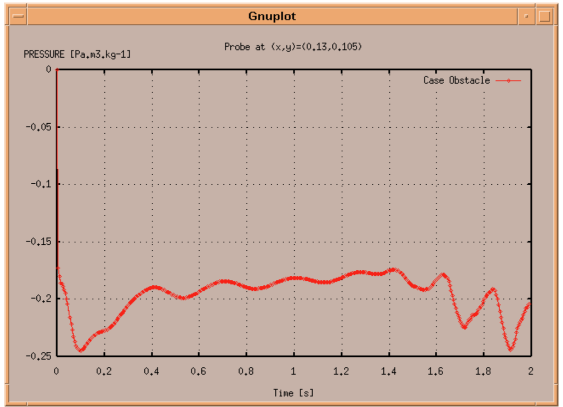
For more informations and to download manuals see :

https://wci.llnl.gov/codes/visit/manuals.html

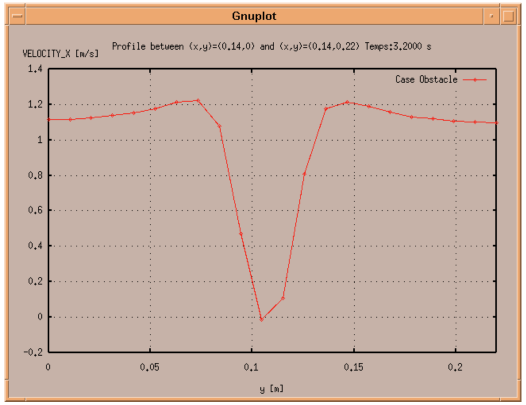
Output files description (6/6)

TRUST with gnuplot

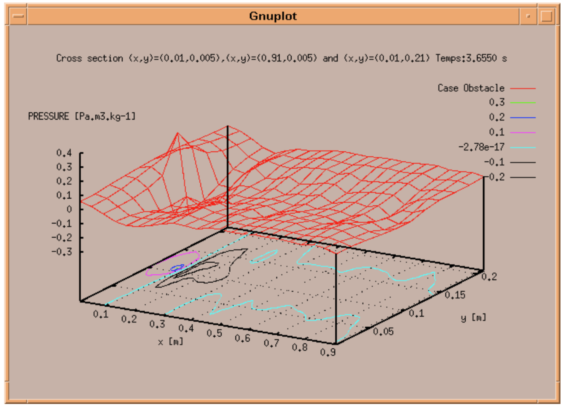
Real-time display of calculated quantities:



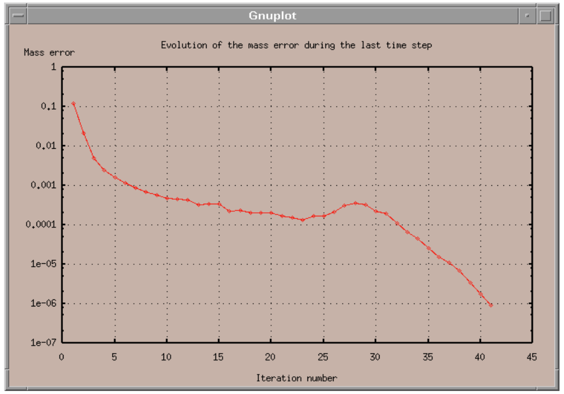
One point probe



Probes segment



Probes plane



Convergence

Instantaneous or averaged value and also, flux balance at the boundaries like:

-flow rate

-pressure forces

-viscous forces

-heat flux

…

Table of contents

- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioCFD
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- **Command lines**
- Parallel calculation
- Mesh generators: Internal tools & Salomé & Gmsh
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

Command lines (1/3)

- TRUST environment initialization:

  source $TRUST_ROOT/env_TRUST.sh

- To run a TRUST calculation with the trust script:
  - Sequential run:

    trust *datafile*

  - Parallel run on N CPUs:
    - Partitioned mesh partitioned should be created sequentially, then interactively:

      trust *datafile N*

    - Or to run on a batch-queuing system, add this line into the submission file :

      mpirun –np *N* ***$exec*** *datafile N*

- To redirect into output and error files, after the command line, add:

  trust *…. 1>datafile.out 2>datafile.err*

Command lines (2/3)

- To copy a data file from the test database:
  trust -copy datafile[.data]

- To visualize the mesh and its boundaries used by a data file:
  trust -mesh datafile[.data]

- To edit interactively (change/add schemes, solvers...) a data file with:
  trust -xedit datafile[.data]

- Check a data file without running TRUST:
  trust -xcheck datafile[.data]

- To identify all the data sets from the no regression data base which contain some specified keywords (word1 word2...wordn). The identified data set are listed in the file 'liste_cherche'.
  trust -search word1 word2 ...wordn

- To monitor only your calculation:
  trust -probes datafile[.data]
  trust -evol datafile[.data]

Command lines (3/3)

- To run VisIt with a LATA results file:
  visit -o datafile.lata &

- To clean your calcul directory:
  trust -clean

- To open the PDF documentation (User's manual):
  trust -doc

- To browse some useful resources (PDF manuals, test cases, keywords, C++ classes,…) :
  trust -index

- In all cases, the TRUST binary may be changed by the **$exec** variable and by default, exec=$TRUST_ROOT/exec/TRUST_mpi_opt

Practice

```
> source /home/triou/env_TRUST_X.Y.Z.sh
> echo $TRUST_ROOT
> cd ~/Formation_TRUST/yourname/Obstacle
> nedit Obstacle.data &
```

- Change the domain name to “truc” instead of “dom” at line 7
- Save and close the file

```
> VerifData Obstacle.data
```

- ERROR !
- Modify it to the previous value.

```
> VerifData Obstacle.data
```

- OK

Practice

> EditData Obstacle.data &

- Expand “pb”, “postraitement”
- Change “lml” to “lata” into “format”
- Close the window and say yes into the terminal.

> VerifData Obstacle.data

- OK

Practice

Exercise: Heat exchange VDF/VEF exercise

Table of contents

- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioCFD
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- **Parallel calculation**
- Mesh generators: Internal tools & Salomé & Gmsh
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

Parallel calculation (1/3)

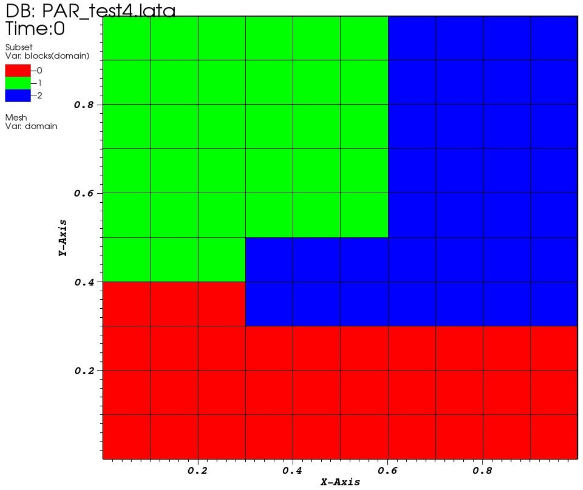
- **Parallel calculation description..**p88
- Parallel calculation on clusters...... p97
- Visualization with VisIt................... p105

Parallel calculation description (1/8)

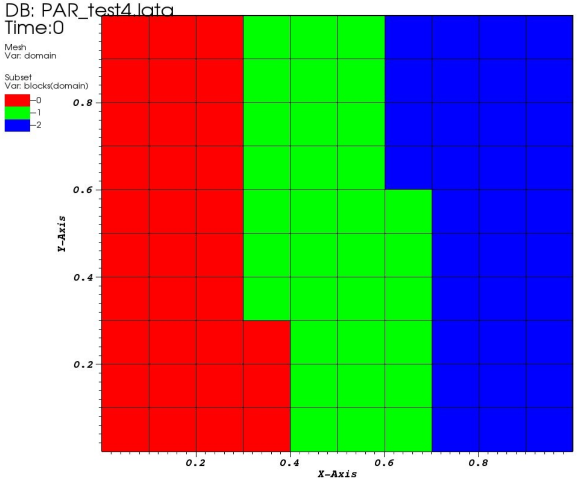
- SPMD model (Single Program Multiple Data)
- Messages exchange by MPI (Message Passing Interface)
- From PC to massively // computer, with shared or distributed memory

Parallel calculation description (2/8)

- Domain partitioning tools:
  - Metis
  - Tranche “band partitioning”



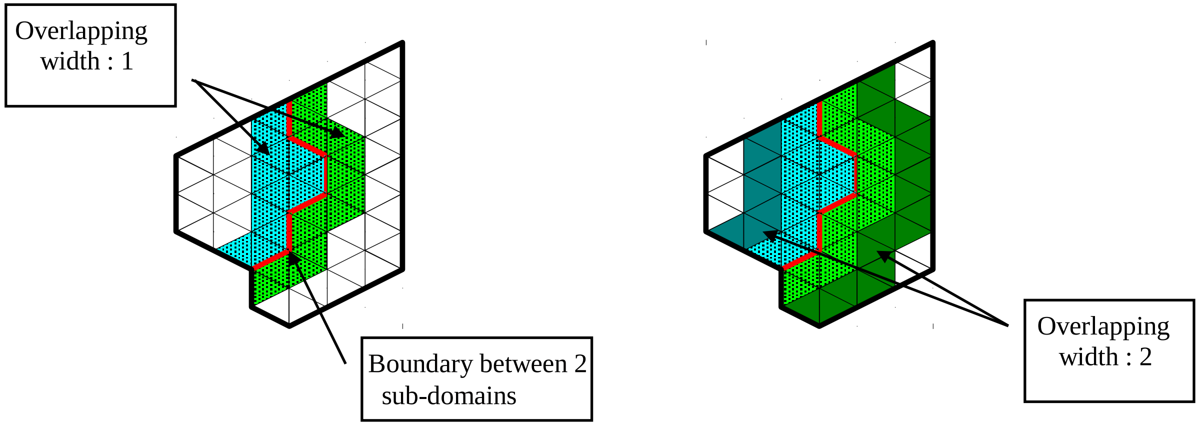
- Performances are partition dependent:
  - Same number of cells by sub-domain
  - To minimize the joints length (boundaries between sub-domains)



- If possible, use 20000-30000 cells per process.

Parallel calculation description (3/8)

- Definition of overlapping width value
  - Number of vertexes or elements on the remote sub-domain known by the local sub-domain
  - Specified by the users during partitioning task
  - This value depends on the space scheme orders:
    - 1 if 1-2nd order
    - 2 if 3-4th order
  - In practice, use 2 except if you use ***only*** upwind schemes



Overlapping width : 1

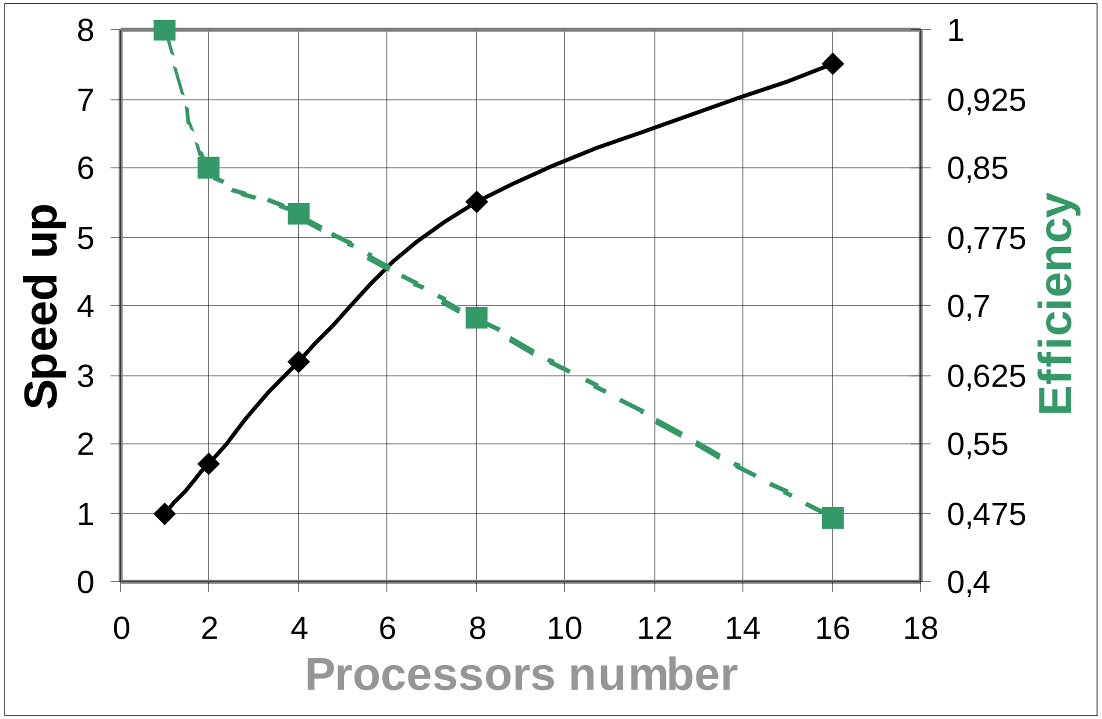
Boundary between 2 sub-domains

Overlapping width : 2

Parallel calculation description (4/8)

- Performance examples :
  - PC linux cluster (Scali network):

Processor number	Speed Up= seq_time/par_time	Efficiency= Speed_Up/nb_procs
1	1	1
2	1.7	0.86
4	3.2	0.80
8	5.5	0.69
16	7.5	0.47

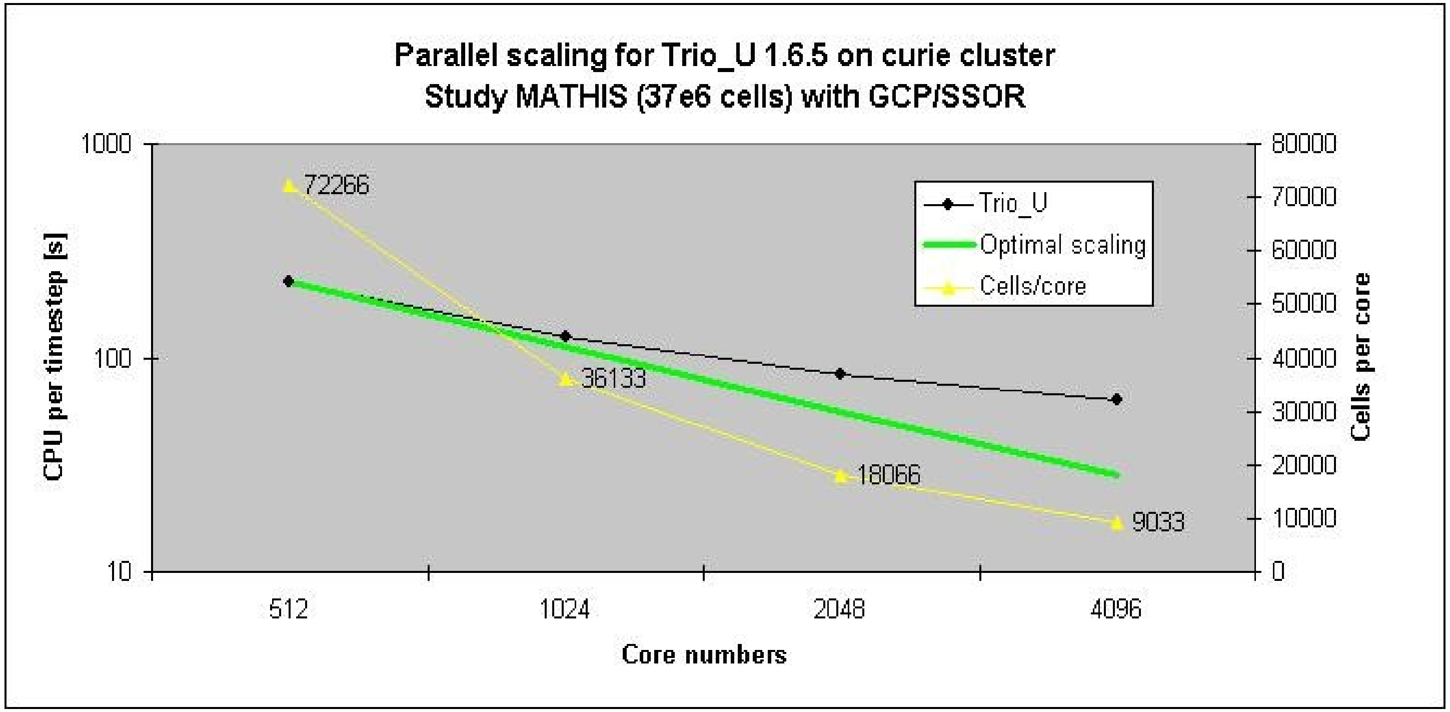


Cf datafile*.TU files

Parallel calculation description (5/8)

Some advices:

- Choose a number of elements per process between 20000 and 30000 for optimal performances.
- Below 20000 elements/process, TRUST parallel efficiency may dramatically decreases. Example on curie:



Scaling of curie from 72000 cells/core to 9000 cells/core

Parallel calculation description (6/8)

⇨ To run a parallel calculation, you must do two runs:

* the first one, to partitioning and create your 'n' sub-domains,

* the second one, to read your 'n' sub-domains and run the calculation on 'n' processors.

Data file description (7/8)

Parallel data file example for the first run

```
dimension 2

# Domain definition #
Domaine dom

# BEGIN MESH #
Read_file Obstacle.geo ;
# END MESH #

# BEGIN PARTITION #
Partition dom
{
        /* Choose Nb_parts so to have ~ 25000 cells per processor */
        Partition_tool metis { nb_parts 2 }
        Larg_joint 2
        zones_name DOM
}
End
# END PARTITION #
```

Data file description (8/8)

Parallel data file example for the second run

```
dimension 2

Domaine dom

# BEGIN SCATTER #
Scatter DOM.Zones dom
# END SCATTER #

VDF ma_discretisation

Scheme_euler_explicit mon_schema
Read mon_schema { ... }

Pb_hydraulique pb

Fluide_Incompressible milieu
Read milieu { ... }

Associate pb dom
Associate pb mon_schema
Associate pb milieu
Discretize pb ma_discretisation

Read pb
{
        Navier_Stokes_standard { ... }
        Post_processing { ... }
}
Solve pb
End
```

Parallel calculation (2/3)

- Parallel calculation description
- **Parallel calculation on clusters**
- Visualization with VisIt

Parallel calculation on clusters (1/7)

- To connect to:
  - CEA/DM2S service cluster: **callisto** (~1 000 cores)
  - CEA/Marcoule cluster: **ceres2** (~700 cores)
  - CEA/Cadarache cluster: **mezel** (~600 cores)
  - CEA/CCRT cluster: **cobalt** (~40 000 cores)
  - CEA/TGCC cluster: **curie-ccrt** (~90 000 cores)
  - CINES cluster: **occigen** (~50 000 cores)

- Ask for a login:
  - callisto: access for CEA only
  - ceres2: access for CEA only
  - mezel: access for CEA only
  - CCRT: http://www-ccrt.cea.fr
  - TGCC: http://www-hpc.cea.fr
  - occigen: http://www.cines.fr

- Once you have the login, connect from your PC to:
  - *ssh -X yourlogin@callisto-login1* or *ssh -X yourlogin@callisto-login2*
  - *ssh -X yourlogin@ceres2*
  - *ssh -X yourlogin@mezel*
  - *ssh -X yourlogin@name.ccc.cea.fr* (name=curie-ccrt | cobalt)
  - *ssh -X yourlogin@occigen.cines.fr*

Parallel calculation on clusters (2/7)

- **TRUST/TrioCFD versions located:**
  - At CEA, on PCs or on callisto:
    by “sourcing”:
    - *source /home/triou/env_TRUST_X.Y.Z.sh*
    - *source /home/triou/env_TrioCFD_X.Y.Z.sh*
  - On clusters :

– mezel	ROOT=/soft/mezel/TRIO
– ceres2	ROOT=/softs/trio_u
– cobalt & curie-ccrt	ROOT=/ccc/cont002/home/den/triou
– occigen	ROOT=/panfs/panasas/softs/applications/trio_u

by « sourcing »:

- *source $ROOT/env_TRUST-X.Y.Z.sh*
- *source $ROOT/env_TrioCFD-X.Y.Z.sh*

- You can add this command line or create **alias** in your **~/.profile** or **~/.bashrc** file:
  alias TRUST_XYZ='source /home/triou/env_TRUST_X.Y.Z.sh'

- **Check your environment**, after you reconnect to the cluster, look at TRUST_ROOT variable:
  echo $TRUST_ROOT
  echo $exec

Parallel calculation on clusters (3/7)

- First sequential run **to partition a mesh:** interactive run (but very time-limited)
  ***trust** datafile*

- Second parallel interactive run: (but very time-limited), for example to check your datafile:
  ***trust** datafile nb_processes*

- **To use the batch queuing system** (for a long time-limited run), you need to create first a submission file (named sub_file):
  ***trust –create_sub_file** datafile nb_processes*

- Then, you submit the job:
  for callisto, mezel, ceres2 & occigen: **sbatch** sub_file
  for cobalt & curie-ccrt: **ccc_msub** sub_file

Parallel calculation on clusters (4/7)

- Don't forget to read the cluster documentation before running your jobs, to be aware of it particularities.

- Before you submit the job, you can edit and change the values of the submission file **sub_file**

- The submission file describes:
  - The job name
  - The number of cores required
  - The default output files
  - The CPU time required (the CPU value selects implicitly a queue)
  - The location of the TRUST study
  - The TRUST parallel command line

Parallel calculation on clusters (5/7)

- Example on callisto

```
#SBATCH -J name_of_the_job
#SBATCH -p slim
#SBATCH --qos=normal
#SBATCH -t 2880
#SBATCH -o myjob.%J.o
#SBATCH -e myjob.%J.e
#SBATCH -n 2
cd $SLURM_BRIDGE_DIR
srun -n  $SLURM_NTASKS  $exec datafile $SLURM_NTASKS 1>jdd.out 2>jdd.err
```

- Example on CCRT/TGCC

```
#MSUB -r job_name
#MSUB -q standard
#MSUB -Q normal
#MSUB -T 86400
#MSUB -o myjob.%J.o
#MSUB -e myjob.%J.e
#MSUB -E "--no-requeue"
#MSUB -n nb_procs
# On cluster curie, add also your project (e.g. Genden):
#MSUB -A genden
cd $BRIDGE_MSUB_PWD
ccc_mprun -n $BRIDGE_MSUB_NPROC $exec datafile $BRIDGE_MSUB_NPROC 1>datafile.out 2>datafile.err
```

- Example on occigen

```
#SBATCH -J name_of_the_job
#SBATCH -t 24:00:00
#SBATCH -o myjob.%j.o
#SBATCH -e myjob.%j.e
#SBATCH --constraint=BDW28
#SBATCH --exclusive
#SBATCH -n 2
#SBATCH -N 1
cd $SLURM_SUBMIT_DIR
srun --mpi=pmi2 -K1 --resv-ports -n $SLURM_NTASKS $exec datafile $SLURM_NTASKS1>jdd.out 2>jdd.err
```

Parallel calculation on clusters (6/7)

- Description of partitions for each cluster:
  - callisto, mezel, ceres2 & occigen: ***sinfo***
  - cobalt & curie-ccrt: ***ccc_mpinfo***

- Description of queues for each cluster:
  - callisto, mezel, ceres2 & occigen: ***sacctmgr list qos***
  - cobalt & curie-ccrt: ***ccc_mqinfo***

- List of jobs and their state:
  - cobalt & curie-ccrt: ***ccc_mpp -u*** *your_login*
  - others: ***squeue -u*** *your_login* *or* ***squeue -j*** *job_number*

- Kill a job (the job_number is given by the previous command):
  - cobalt & curie-ccrt ***ccc_mdel*** *job_number*
  - others: ***scancel*** *job_number*

Parallel calculation on clusters (7/7)

- Complet informations for each cluster:
  - callisto: from cluster, “evince /cm/shared/docs/callisto.pdf”
  - mezel: see https://www-linuxcad.intra.cea.fr/doku.php
  - ceres2: see https://www-linuxmar.intra.cea.fr/dokuwiki/doku.php
  - cobalt: see https://www-tgcc.ccc.cea.fr/docs/cobalt.info.html
  - curie-ccrt: see https://www-tgcc.ccc.cea.fr/docs/curie.info.html
  - occigen: see https://www.cines.fr/calcul/materiels/occigen
    and https://www.cines.fr/tt6-comment-choisir-la-partition-doccigen

- *Space disc on each cluster:*
  - $HOME limited space for source code & binaries, backup
  - $SCRATCHDIR large space for calculation datas, outputs, no backup
  - $STOREDIR space for data archiving, file's number limited, backup

Parallel calculation (3/3)

- Parallel calculation description
- Parallel calculation on clusters
- **Visualization with VisIt**

Visualization with VisIt (1/2)

On clusters:

- On CCRT/TGCC clusters, use a **HPCDrive** or **VBridge** deported session to run VisIt without network slowness:
  - Available on curie cluster: https://visu-tgcc.ccc.cea.fr/HPCDrive/home
  - Available on cobalt cluster: https://visu-cobalt.ccc.cea.fr/HPCDrive/home
  - Ask us the HPCDrive user manual
- To visualize Occigen results, connect to **cristal** cluster:
  - https://www.cines.fr/calcul/materiels/la-machine-de-pre-post-traitement-cristal
- Or the **client/server mode**:
  - See the following description VisIt (**callisto**)
  - Unhappily, this mode DOES NOT work with CCRT/TGCC clusters
  - Fine tuning of a critical option: Rendering->Advanced->Auto (2000KPolys)
- Or local mode
  - Copy the LATA results from the cluster to your PC and run the parallel version of VisIt

Visualization with VisIt (2/2)

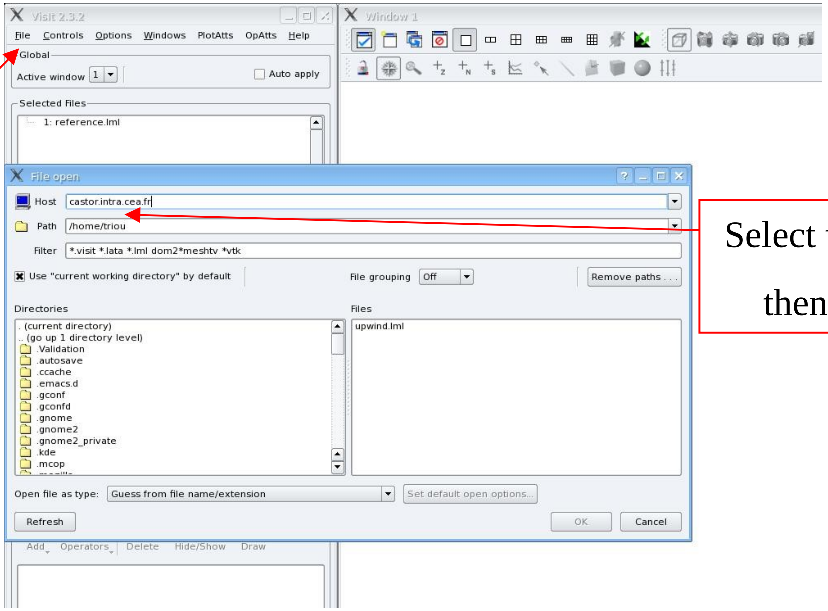
Features with VisIt

-The TRUST install builds a parallel version of VisIt:

```
visit –np 8 –o results.lata
```

-Client/server mode available by default for some clusters (callisto)

- You run the TRUST calculation on the cluster
- You visualize with VisIt your results from your Linux/Windows PC without data copy and/or network slowness


VisIt 2.3.2
File Controls Options Windows PlotAtts OpAtts Help
Global
Window 1
File open
Add Operators Delete Hide/Show Draw
Refresh
OK
Cancel
Remove paths . . .
Set default open options...

Active window 1 ▾ □ Auto apply

Selected Files: 1: reference.lml

Host: castor.intra.cea.fr

Path: /home/triou

Filter: *.visit *.lata *.lml dom2*meshtv *vtk

☒ Use "current working directory" by default

File grouping: Off

Directories:
. (current directory)
.. (go up 1 directory level)
.Validation
.autosave
.ccache
.emacs.d
.gconf
.gconfd
.gnome
.gnome2
.gnome2_private
.kde
.mcop

Files:
upwind.lml

Open file as type: Guess from file name/extension

File->Open file…

Select the host (remote host or localhost)

then browse the files to your study.

Practice

Exercise: Obstacle.data //

Exercise: Calculation on callisto

Exercise: Turbulent flow on a 3D step

Table of contents

- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioCFD
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- Parallel calculation
- **Mesh generators: Internal tools & Salomé & Gmsh**
- Automating validation test case
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

Mesh generators (1/5)

- **Possible meshing tools................**p110
- TRUST internal mesh tool..............p114
- Salomé......................................p116
- Gmsh....................................... p118
- Read a med file in TRUST.............p122

Possible meshing tools (1/3)

- TRUST internal mesh tool
  - Used by keywords in the data file
  - Limited to simple geometry (assembling of rectangle in 2D or blocks in 3D)
- Xprepro (only VDF)
  - External tool created by the Trio_U team
  - To create a complex geometry but with a regular hexahedral mesh
  - cf $TRUST_ROOT/doc/TRUST/Manuel_Xprepro.pdf
- Or use of a mesh generator tool linked with TRUST:
  - Salomé
  - Gmsh

Possible meshing tools (2/3)

- Mesh generator tools :
  - ICEM (Ansys) generates a file at TRUST format
  - TRUST reads .unv files from I-DEAS*
  - TRUST reads 2D/3D meshes from old tools of Fluent (Gambit/TGrid)
  - TRUST reads « .med » meshes from Salomé or Gmsh

- Form factors (view factors for the radiation model):
  - Link between Ansys and TRUST

*:tetrahedral meshing only

Possible meshing tools (3/3)

- Presentation of:
  - TRUST internal mesh tool,
  - Salomé,
  - Gmsh.
- Exercise with **Salomé** or **Gmsh** according to your needs

Mesh generator	Salomé	Gmsh	Other (ICEM,...)
Availability	Free	Free	License
TRUST discretization	VEF	VEF	VEF
TRUST package	No	Yes	No
TRUST tutorial	Yes	Yes	No
Support	support-salome@cea.fr	gmsh@geuz.org	

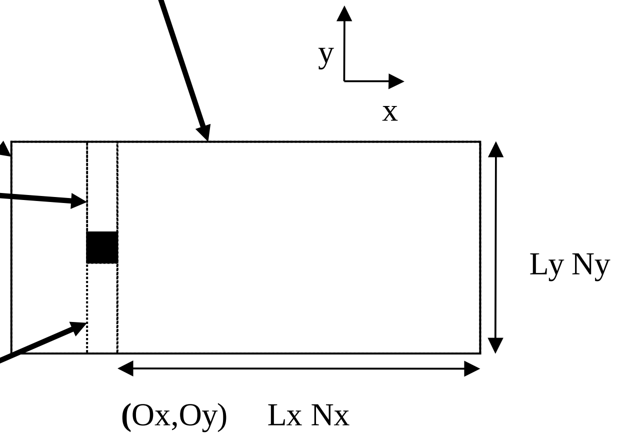
Mesh generators (2/5)

- Possible meshing tools
- **TRUST internal mesh tool**
- Salomé
- Gmsh
- Read a med file in TRUST

TRUST internal mesh tool (1/1)

```
Mailler dom {
Pave Left
              {
              Origine 0. 0.
              Nombre_de_Noeuds 6 23
              Longueurs 0.1 0.22
              }
              {
              bord Inlet X = 0. 0. <= Y <= 0.22
              bord Upper Y = 0.22 0. <= X <= 0.1
              bord Lower Y = 0. 0. <= X <= 0.1
              bord Square X = 0.1 0.1 <= Y <= 0.12
              } ,

Pave Up
              {
              Origine 0.1 0.12
              Nombre_de_Noeuds 2 11
              Longueurs 0.02 0.1
              }
              {
              bord Upper Y = 0.22 0.1 <= X <= 0.12
              bord Square Y = 0.12 0.1 <= X <= 0.12
              } ,
Pave Bottom
              {
              Origine 0.1 0.
              Nombre_de_Noeuds 2 11
              Longueurs 0.02 0.1
              }
              {
              bord Lower Y = 0. 0.1 <= X <= 0.12
              bord Square Y = 0.1 0.1 <= X <= 0.12
              } ,
```

```
        Pave Right
                      {
                      Origine 0.12 0.
                      Nombre_de_Noeuds 41 23
                      Longueurs 0.8 0.22
                      }
                      {
                      bord Outlet X = 0.92 0. <= Y <= 0.22
                      bord Lower Y = 0. 0.12 <= X <= 0.92
                      bord Upper Y = 0.22 0.12 <= X <= 0.92
                      bord Square X = 0.12 0.1 <= Y <= 0.12
                      }
}
```



Origine: Block origin Ox Oy [Oz]

Nombre_de_noeuds: Nodes number Nx Ny [Nz]

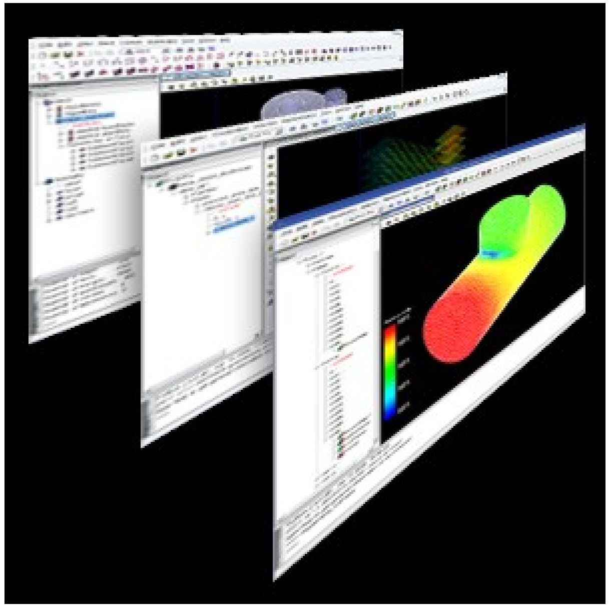
Longueurs: Lengths Lx Ly [Lz]

Bord: Boundary

Mesh generators (3/5)

- Possible meshing tools
- TRUST internal mesh tool
- **Salomé**
- Gmsh
- Read a med file in TRUST

Salomé (1/1)



Salomé: An OpenSource platform (CEA, EDF, OpenCascade,...) which provides a 2D/3D element mesh generator. It is available here for download:

http://www.salome-platform.org

-> **Salomé** is not provided in the TRUST package

-> To have a full training session, or receive support for install or use, see:

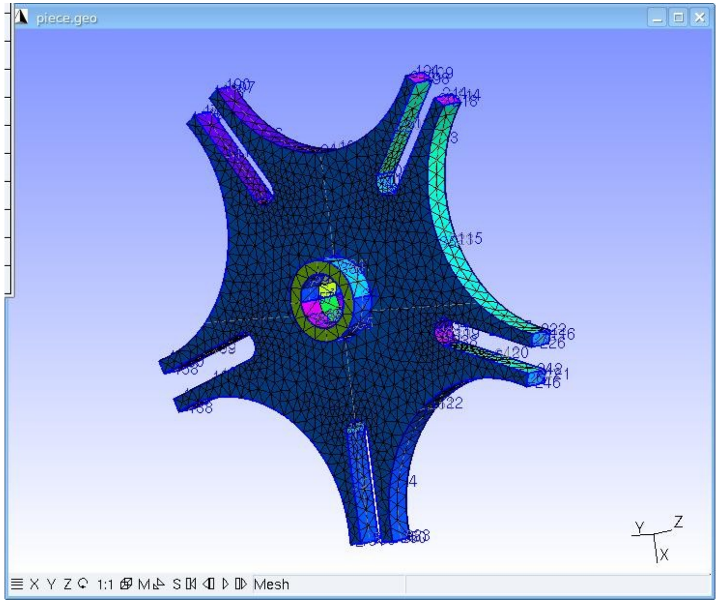
http://www.salome-platform.org/service-and-support

support-salome@cea.fr

Mesh generators (4/5)

- Possible meshing tools
- TRUST internal mesh tool
- Salomé
- **Gmsh**
- Read a med file in TRUST

Gmsh (1/3)



Gmsh: A 2D/3D finite element mesh generator available here:

http://www.geuz.org/gmsh

-> The documentation is here:

http://geuz.org/gmsh/doc/texinfo/gmsh.html

-> Gmsh is downloaded and built during the TRUST install. There are tutorials and examples under:

$TRUST_ROOT/exec/gmsh/share/doc/gmsh/tutorial

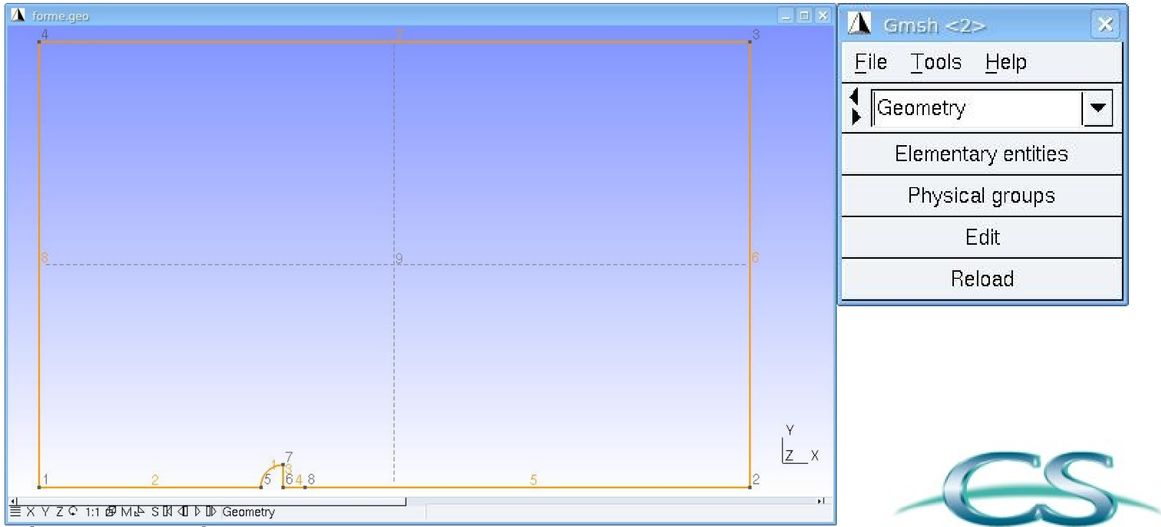
$TRUST_ROOT/exec/gmsh/share/doc/gmsh/demos

-> Support on Gmsh at gmsh@geuz.org

Best is to start from a .geo file of one of the previous examples. Run Gmsh with:

gmsh *file.geo*

Edit and change your .geo file and use the **Reload** button to update the geometry visualization.

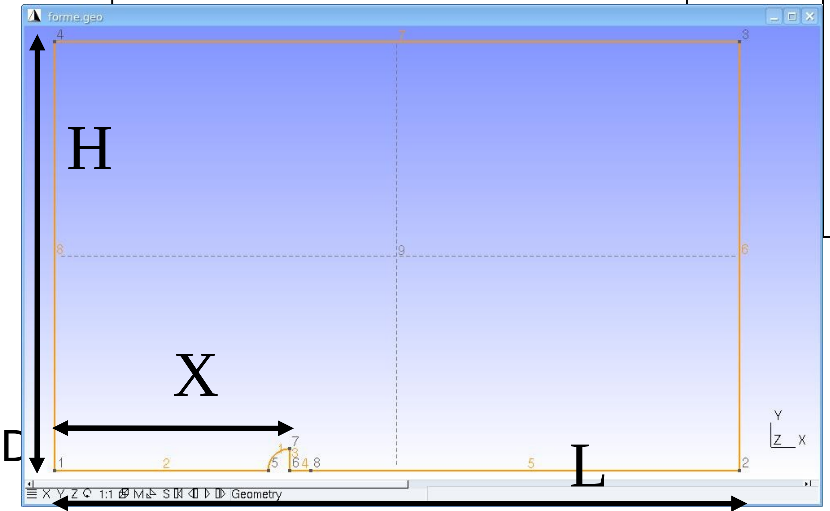


Gmsh (2/3)

Example of .geo file

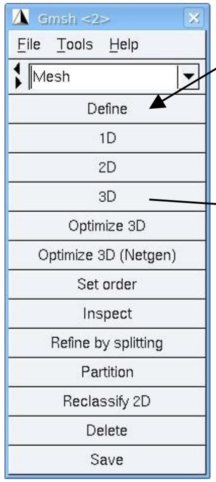
```
// Variables definition
lc = 0.02;
// First cell size (used when points
// are defined):
lc1 = lc * 8;
// Second cell size
lc2 = lc / 2;
// Circle diameter
D = 0.14 ;
E = D ;
param = 1;
H = param * 10 * D ;
X = param * 5 * D ;
L = param * 10 * D + X + E;

// Points definition
Point(1) = {0,0,0,lc1};
Point(2) = {L,0,0,lc1};
Point(3) = {L,H,0,lc1};
Point(4) = {0,H,0,lc1};
Point(5) = {X,0,0,lc2};
Point(8) = {X+E,0,0,lc2};
```

```
// Lines definition
Line(2) = {1,5};  // 2 points
Line(5) = {8,2};
Line(6) = {3,2};
Line(7) = {3,4};
Line(8) = {4,1};

// 1/4 Circle definition
Point(6) = {X+D/2,0,0,lc2}; // Center
Point(7) = {X+D/2,D/2,0,lc2};
// 3 points for the circle arc (P1,Center,P2):
Circle(1) = {5,6,7};
Line(3) = {7,6};
Line(4) = {6,8};
// A circle arc is STRICTLY smaller than Pi
```



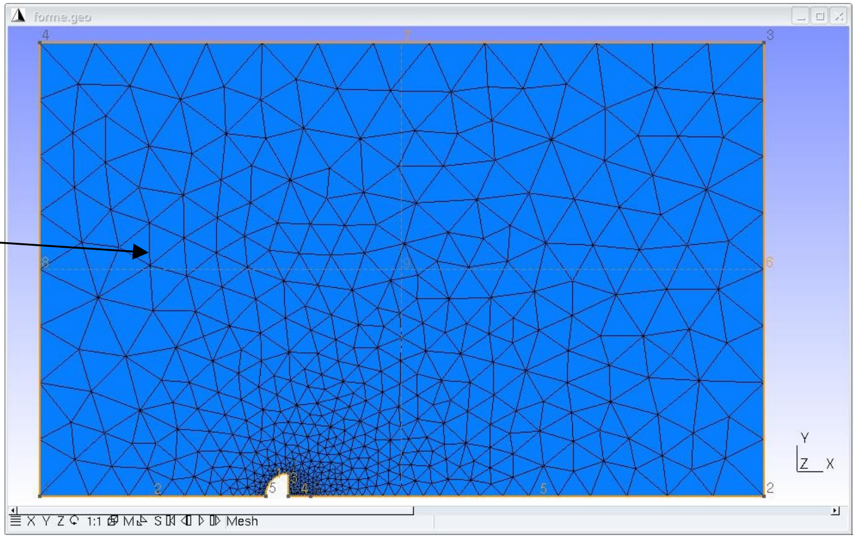
```
// Naming the boundaries is MANDATORY
// and it is thanks to the
// Physical Line (use Lines or Circle to
define it)
// DO NOT USE LINE LOOPS !!!!
Physical Line(“Shape") = {1,3};
Physical Line("Axis") = {2,4,5};
Physical Line(“Outlet") = {6};
Physical Line(“Top") = {7};
Physical Line(“Inlet") = {8};

// A lineloop is a loop on several lines
// for defining/orienting a surface
// Use negative lines to reverse the
// orientation of the line
Line Loop(1) = {2,1,3,4,5,-6,7,8};
// The surface will use the lineloop
Plane Surface(1) = {1};
// Naming the domain is MANDATORY
Physical Surface("domain") = {1};
```

Gmsh (3/3)



Switch from Geometry to Mesh

2D mesh (in 3D, 3D mesh AND do not forget to optimize 3D after that)



Then, export the mesh to a MED format file (File->Save As, format MED) and DO NOT select “Save All” because points could be saved.

Important: Check that your mesh is created with the command (if nothing appears, you forgot to name boundaries and/or the domain with the Physical keywords):

gmsh ***file.med***

Mesh generators (5/5)

- Possible meshing tools
- TRUST internal mesh tool
- Salomé
- Gmsh
- **Read a med file in TRUST**

Read a med file in TRUST (1/1)

***** To import a .med mesh file, add in the TRUST datafile:

Dimension 2

Domaine dom

*# By default with **Gmsh**, **the mesh name is the name of the file**, so there mesh_name=file #*

Read_MED family_names_from_group_names dom mesh_name *file.med*

***** If you have created several domains into the same Gmsh mesh, you will add cause the different domains in the Gmsh mesh file are seen as subzones.

Domain second_dom

Create_domain_from_sous_zone { **domain_final** second_domain **par_sous_zone** sub_zone_name **domaine_init** dom }

***** Possibility to **create a TRUST data file by opening a .med mesh**:

> **trust -wiz**

⇨ in the choice of domain set your .med file.

The wizzard will automatically find your boundary names,...

Practice

Exercise: Meshing tools

VEF calculation with TRUST

- Run Salomé exercise in the tutorial
- Run Gmsh exercise in the tutorial

Run Xprepro exercise in the tutorial if one is interested by a VDF calculation with TRUST

Table of contents

- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioCFD
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- Parallel calculation
- Mesh generators: Internal tools & Salomé & Gmsh
- **Automating validation test case**
- TRUST/TrioCFD support
- Examples of data files
- Recommendations

Automating validation test case (1/16)

- What is an automated test case ?
- How to generate an automated test case?
- How to run an automated test case?

Automating validation test case (2/16)

- **Verification of the non-regression** is done by running **.prm automated tests cases.**

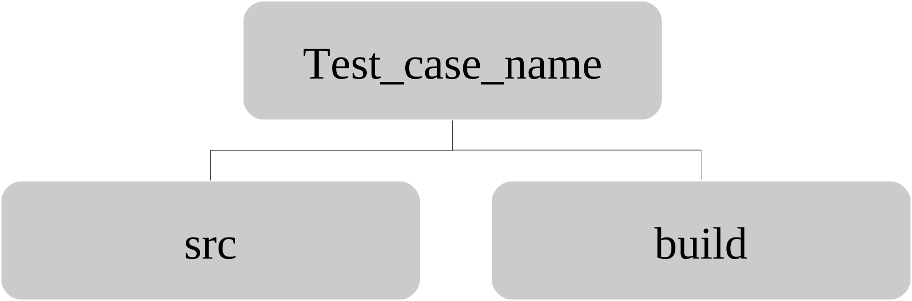
- This is a **tool to compare TRUSTs results** and experimental data and/or analytical solutions.

- Finaly it creates a **report in PDF format** containing:
  - » figures (images or gnuplot plots)
  - » tables with results
  - » visualizations (built by VisIt tool)
  - » etc…

- Useful to quickly validate a new TRUST/TrioCFD version or to compare different versions of the code

Automating validation test case (3/16)

- **How to generate an automated test case?**
  - Create a directory for example Test_case_name and under:
    - src directory will contain the elements to build the test case
  - TRUST will create a build directory which will contain the results of the building run



  - Example of a study report

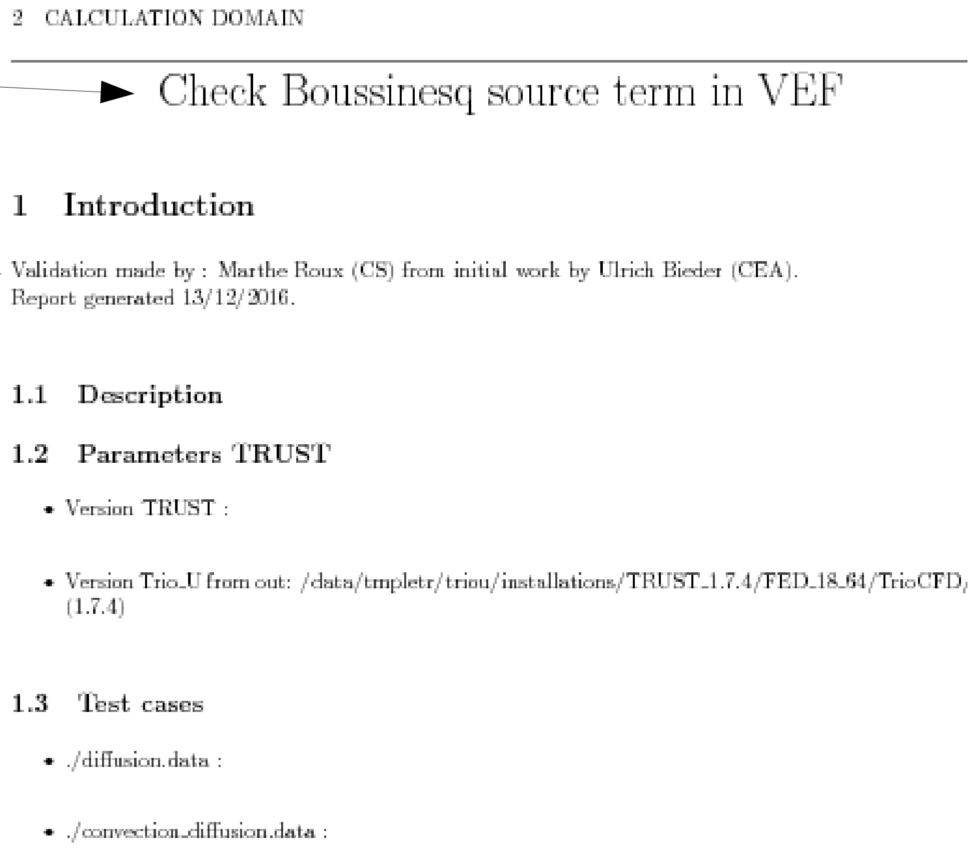
Automating validation test case (4/16)

- **How to generate an automated test case?**
  - In the src directory, you will create a:
    - **.prm** file (mandatory)
      - It contains the automated test case description
    - **prepare** script (optional)
      - It will build one (several) data file(s) in the build sub-directories
    - **pre_run** script (optional)
      - It will do operations BEFORE calculations begin, for example the mesh partition before a parallel calculation
    - **post_run** script (optional)
      - It will extract relevant information from the raw result files AFTER the calculation finishes (run python scripts,...)
    - other optional files/directories necessary to build the test case
      - data file, mesh file, experimental data, images,…

Automating validation test case (5/16)

- Example of a **prm** file and his **pdf**:

```
Parameters {
   Title "Check Boussinesq source term in VEF"
   Author "Marthe Roux (CS) from initial work by Ulrich Bieder
(CEA)"
   Testcase .  diffusion
   Testcase .  convection_diffusion
   ...
}
```



2 CALCULATION DOMAIN

Check Boussinesq source term in VEF

1 Introduction

Validation made by : Marthe Roux (CS) from initial work by Ulrich Bieder (CEA).
Report generated 13/12/2016.

1.1 Description

1.2 Parameters TRUST

- Version TRUST :
- Version Trio_U from out: /data/tmpletr/triou/installations/TRUST_1.7.4/FED_18_64/TrioCFD/
(1.7.4)

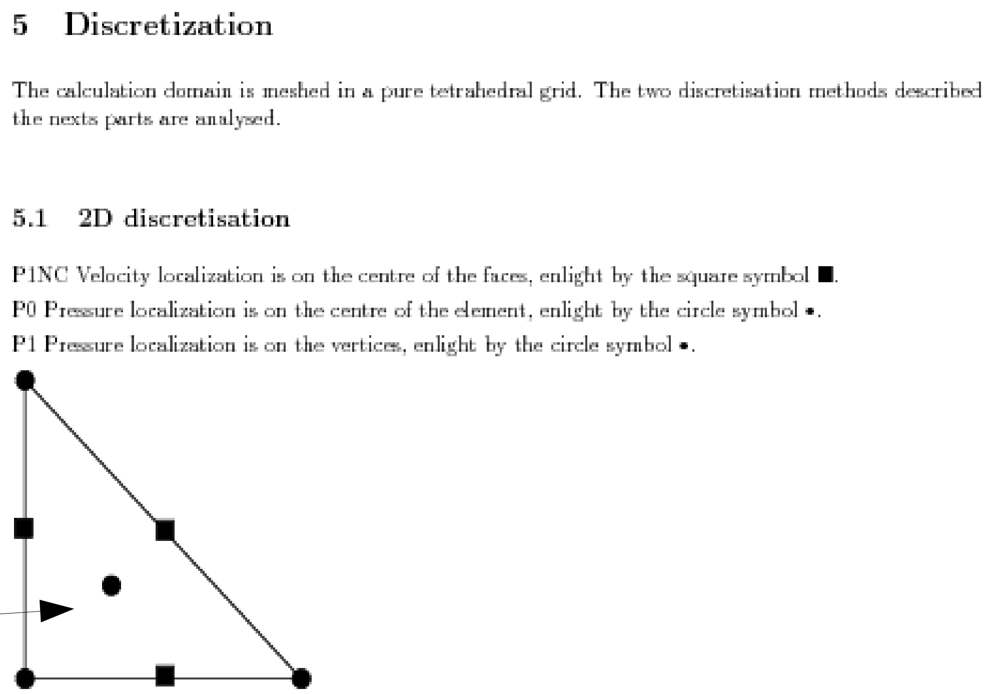
1.3 Test cases

- ./diffusion.data :
- ./convection_diffusion.data :

Automating validation test case (6/16)

- Example of a **prm** file and his **pdf**:

```
Chapter {
   Title "Discretization"
   Description "The calculation domain is meshed in a pure tetrahedral
grid. The two discretisation methods described the nexts parts are
analysed."

   Figure {
      Title "2D discretisation"
      description "P1NC Velocity localization is on the centre of the
faces, enlight by the square symbol $\blacksquare$."
      description "P0 Pressure localization is on the centre of the
element, enlight by the circle symbol $\bullet$."
      description "P1 Pressure localization is on the vertices, enlight by
the circle symbol $\bullet$."
      Width 5cm
      picture triangle.pdf
   }
...
}
```



5 Discretization

The calculation domain is meshed in a pure tetrahedral grid. The two discretisation methods described the nexts parts are analysed.

5.1 2D discretisation

P1NC Velocity localization is on the centre of the faces, enlight by the square symbol ■.

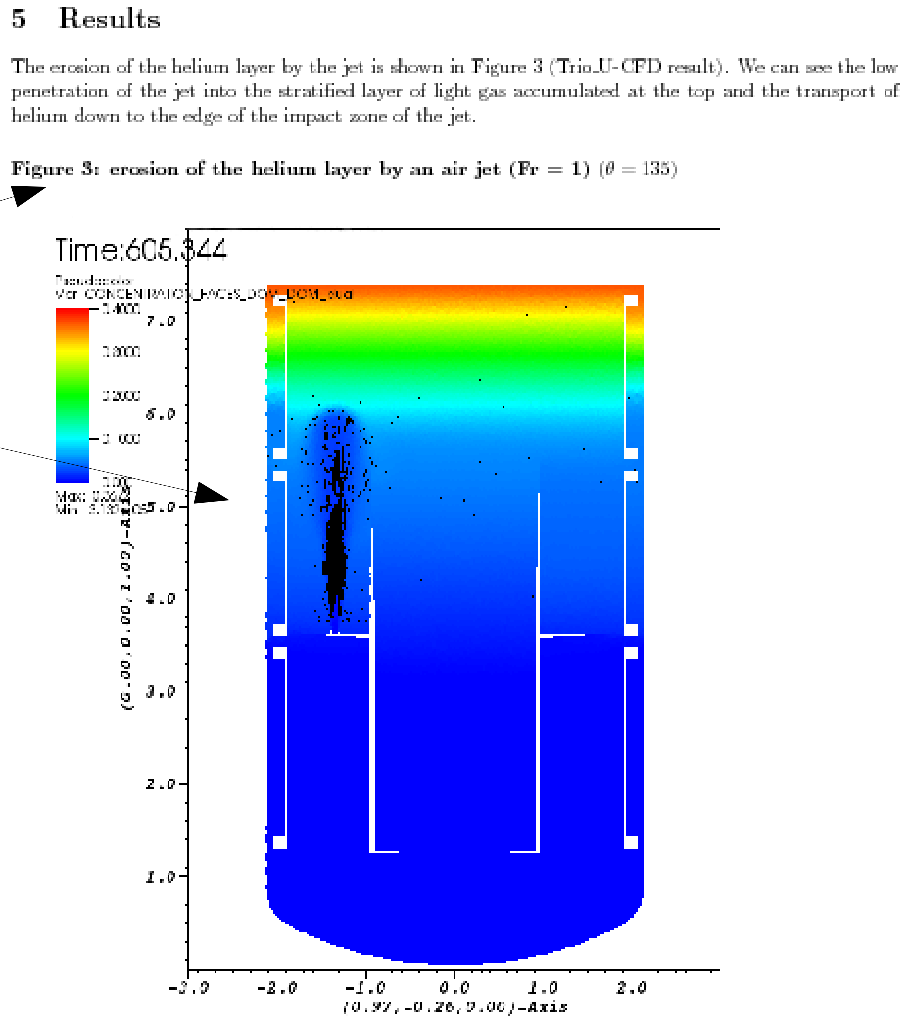
P0 Pressure localization is on the centre of the element, enlight by the circle symbol •.

P1 Pressure localization is on the vertices, enlight by the circle symbol •.

Automating validation test case (7/16)

- Example of a **prm** file and his **pdf**:

```
Chapter {
   Title "Results"
   Description "The erosion of the helium layer by the jet is shown in
Figure 3 (Trio_U-CFD result). ..."
   Description "\latex_( \vspace{0.5cm} \textbf{Figure 3: erosion of
the helium layer by an air jet (Fr = 1)} ($\theta=135$) \latex_)"
   Visu {
      Width                16cm
      pseudocolor_with_range  calculs/results.lata
                     DOM_DOM_dual CONCENTRATION FACES 0. 0.4
      operator             slice2D  0 0 0 -0.258819 -0.965926 0.
      BlackVector_with_nb     calculs/results.lata
                              DOM_DOM_dual VITESSE FACES 3000 1.4
      operator             clip_1plane 0 0 3.75 0 0 -1.
      operator             slice2D  0 0 0 -0.258819 -0.965926 0.
      zoom2D               -3. 3. 0. 8.
   }
}
```



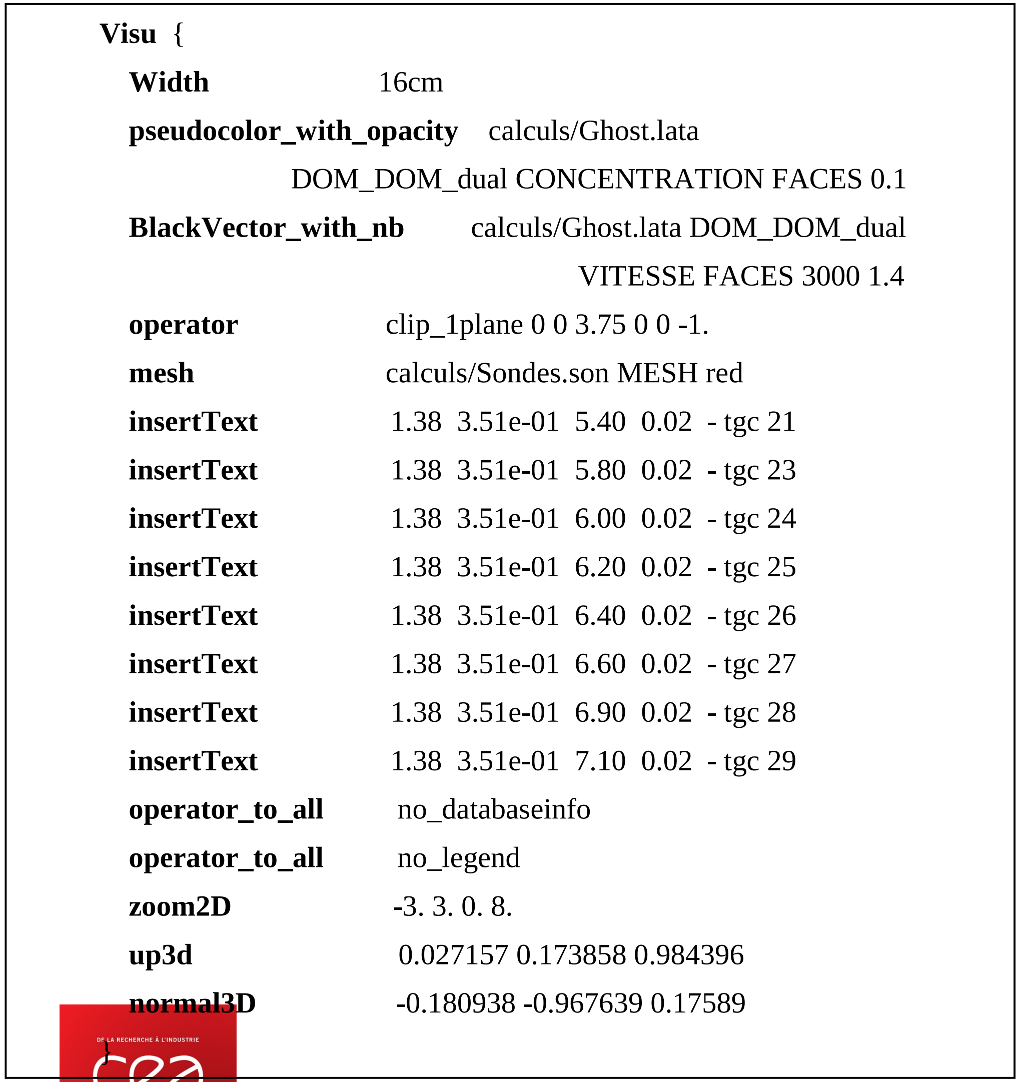
5 Results

The erosion of the helium layer by the jet is shown in Figure 3 (Trio_U-CFD result). We can see the low penetration of the jet into the stratified layer of light gas accumulated at the top and the transport of helium down to the edge of the impact zone of the jet.

Figure 3: erosion of the helium layer by an air jet (Fr = 1) ($\theta = 135$)

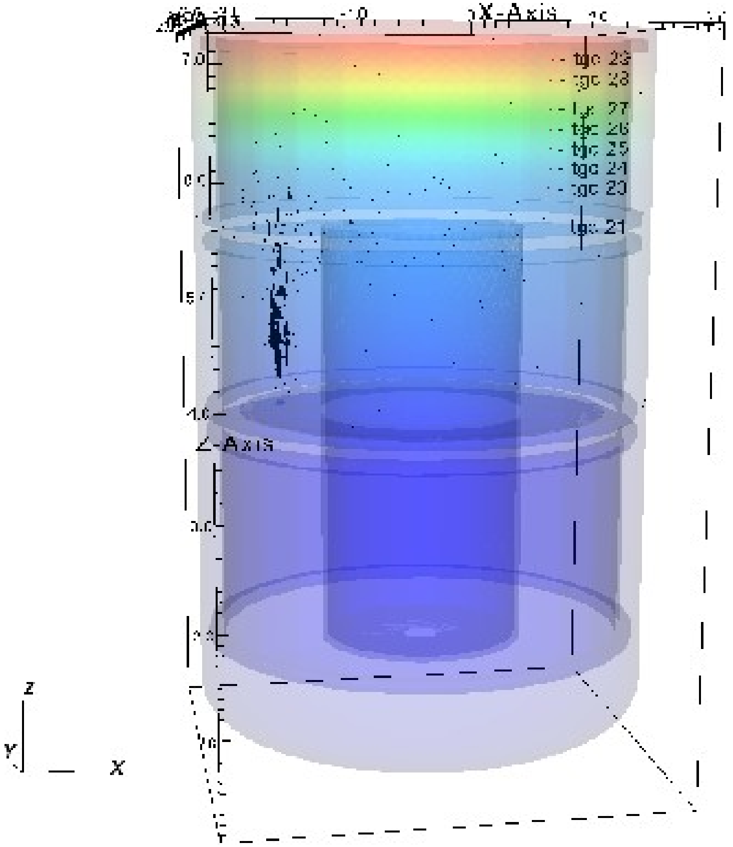
Automating validation test case (8/16)

- Example of a **prm** file and his **pdf**:



```
Visu {
   Width                     16cm
   pseudocolor_with_opacity    calculs/Ghost.lata
                       DOM_DOM_dual CONCENTRATION FACES 0.1
   BlackVector_with_nb        calculs/Ghost.lata DOM_DOM_dual
                                          VITESSE FACES 3000 1.4
   operator                  clip_1plane 0 0 3.75 0 0 -1.
   mesh                      calculs/Sondes.son MESH red
   insertText                 1.38  3.51e-01  5.40  0.02  - tgc 21
   insertText                 1.38  3.51e-01  5.80  0.02  - tgc 23
   insertText                 1.38  3.51e-01  6.00  0.02  - tgc 24
   insertText                 1.38  3.51e-01  6.20  0.02  - tgc 25
   insertText                 1.38  3.51e-01  6.40  0.02  - tgc 26
   insertText                 1.38  3.51e-01  6.60  0.02  - tgc 27
   insertText                 1.38  3.51e-01  6.90  0.02  - tgc 28
   insertText                 1.38  3.51e-01  7.10  0.02  - tgc 29
   operator_to_all            no_databaseinfo
   operator_to_all            no_legend
   zoom2D                     -3. 3. 0. 8.
   up3d                       0.027157 0.173858 0.984396
   normal3D                   -0.180938 -0.967639 0.17589
}
```

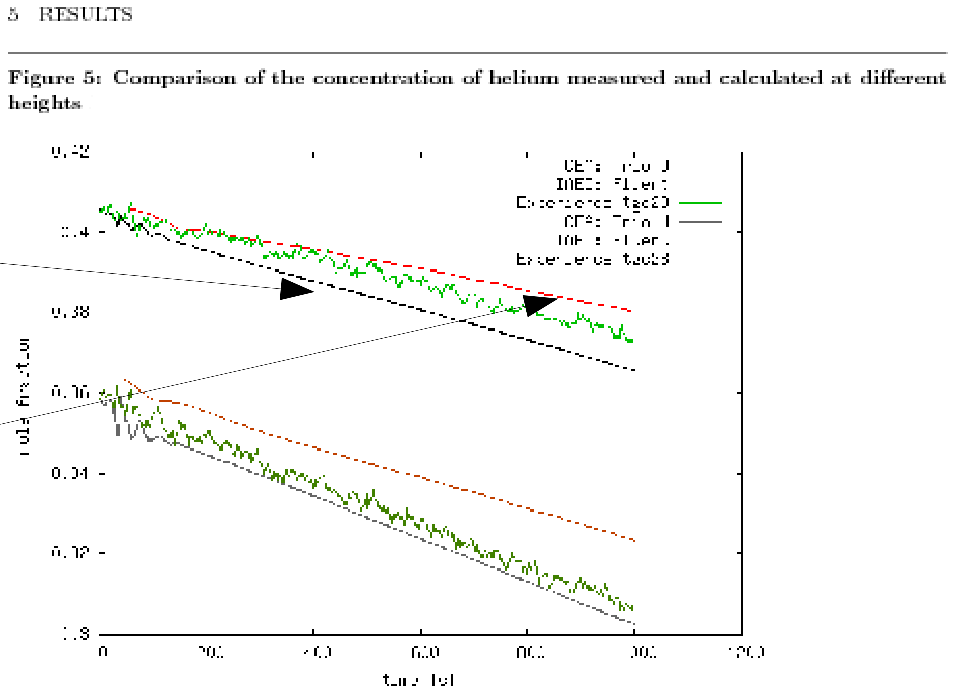
Figure 4: Location of concentration sampling probes



Automating validation test case (9/16)

- Example of a **prm** file and his **pdf**:

```
Figure {
        labelx "time [s]"
        labely "mole fraction"
        Width 13cm
        Include_Description_Curves 0
        legend top
        Curve {
            fichier calculs/results_SONDE_C_V7.son
            colonnes $1
                            $22-read_value_in_file("VAR_EXPE_11","calage")
            legende "CEA: Trio_U"
            style lines
            TypeLigne -1
        }
        Curve {
            fichier Calculs_IAEC/calc_fluent.txt
            colonnes $1 $11
            legende "IAEC: Fluent"
            style lines
            TypeLigne 1
        }
```



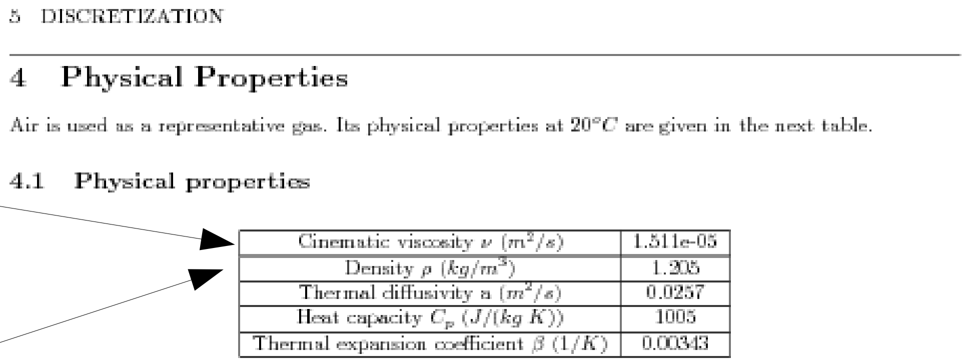
5 RESULTS

Figure 5: Comparison of the concentration of helium measured and calculated at different heights

Automating validation test case (10/16)

- Example of a **prm** file and his **pdf**:

```
Table {
        Title "Physical properties"
        nb_columns 1
        line {
            legend "Cinematic viscosity $\nu$ ($m^2/s$)"
            file propertiesGeometry.dat
            nb_columns_file 5
            columns ($1)
        }
        line {
            legend " Density $\rho$ ($kg/m^3$)"
            file propertiesGeometry.dat
            nb_columns_file 5
            columns ($2)
        }
       ...
}
```



5 DISCRETIZATION

4 Physical Properties

Air is used as a representative gas. Its physical properties at $20°C$ are given in the next table.

4.1 Physical properties

Cinematic viscosity ν (m^2/s)	1.511e-05
Density ρ (kg/m^3)	1.205
Thermal diffusivity a (m^2/s)	0.0257
Heat capacity C_p $(J/(kg\ K))$	1005
Thermal expansion coefficient β $(1/K)$	0.00343

Automating validation test case (11/16)

- Example of a **prepare** script:

```
#!/bin/bash
# Loop on several convection schemes:
for scheme in “muscl ef_stab upwind”
do
   # Create a sub directory into the build directory
   mkdir  $scheme
   # Go into the sub directory
   cd $scheme
   # Copy the flow.data into a new file named flow.data
   cp ../flow.data flow.data
   # Substitute into the the data file the SCHEME string by  the the value of  the  $scheme variable
   echo –e “1,$ s?SCHEME?$scheme?g\nw” | ed flow.data
   # Create a link with pre_run and post_run script
   ln –s –f ../pre_run .
   ln –s –f ../post_run .
   # Come back to the build directory
   cd ..
done
```

Automating validation test case (12/16)

- Example of a **pre_run** script:

```
#!/bin/bash
# Uncompress the mesh file
gunzip –c ../Channel.msh.gz  >  Channel.msh
# Partition the mesh:
trust -partition flow
```

Automating validation test case (13/16)

- Example of **post_run** script:

```
#!/bin/bash
# The first parameter is the name of the data file:
datafile=$1
file=${datafile%.data}
# Read the pressure drag on the 5th column of the last line (final time) of
# the pressure force file:
fp=`tail -1 $file"_pb_Force_pression.out" | awk '{print $5}'`
# Read the viscous drag:
fv=`tail -1 $file"_pb_Contrainte_visqueuse.out" | awk '{print $5}'`
# Calculate the total drag:
Drag=‘echo $fp $fv | awk '{print $1+$2}'‘
#Drag=`echo “$fp+$fv” | bc –l`
# Write the total drag into a file to be included into a table of the PDF file
echo $Drag > drag.dat
# You can also call python scripts...
```

Automating validation test case (14/16)

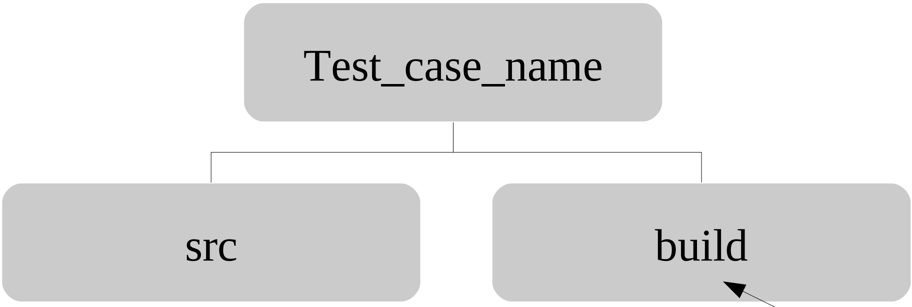
- **How to run an automated test case?**

  - “**Run_fiche**” command should be run either from the root directory of the test case, either in the src directory
  - All operations made by **Run_fiche** are in the build directory:
    - » First, it runs the **prepare** script
    - » Then for each calculation:

      -runs the **pre_run** script

      -runs the calculation

      -runs the **post_run** script
    - » Then builds the <u>**PDF report file**</u>



You have acces to the **latex/images/... files** in the directory:

build/.tmp

Automating validation test case (15/16)

- **User guide:**
  $TRUST_ROOT/doc/TRUST/HowTo_Validation.pdf

- **.prm syntax** documented in:
  $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml

- **Examples** of automated verification test case:
  $TRUST_ROOT/Validation/Rapports_automatiques/Verification/Verification_codage

- *To have **information** about scripts:*

  “*trust -index*” → “Memo scripts” link

Automating validation test case (16/16)

- **To identify all the data sets** from the non-regression data base which contain some specified keywords (word1 word2...wordn):

  trust -search *[-reference_only] word1 word2 ...wordn*

  → *results in file 'liste_cherche'*

- **List of tests cases** with little explanation:

  “*trust -index*” → “Test cases” link

- **Tests cases ending with .jdd1, .jdd2 ...** are test cases from .prm automated test cases run on 3 time steps.

- Most of the **validation tests cases** are in TrioCFD.

Practice

Exercise with “-search” option:

```
> source /home/triou/env_TRUST_X.Y.Z.sh
> trust -search Fluide_Quasi_Comp*
> more liste_cherche
> trust -search *FT_Disc*
> source /home/triou/env_TrioCFD_X.Y.Z.sh
> trust -search *FT_Disc*
> trust -search Fluide_Quasi_Comp*
```

To have access to all test cases:

```
> source .../full_env_TrioCFD.sh
```

Exercise: Validation form

Table of contents

- TRUST/TrioCFD historic
- Modeling flow with TRUST/TrioFD
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- Parallel calculation
- Mesh generators: Internal tools & Salomé & Gmsh
- Automating validation test case
- **TRUST/TrioCFD support**
- Examples of data files
- Recommendations

TRUST/TrioCFD support (1/2)

Subscribe to the TrioCFD newsletters (diffusion list):

- Users (1 mail/trimester) http://saxifrage:3500/wws/info/trio_u_annonces
- Developers (1 mail/week) http://saxifrage:3500/wws/info/trio_u_dev

Download new versions from sourceforge site:

- TRUST: http://sourceforge.net/projects/trust-platform/files/
- TrioCFD: http://sourceforge.net/projects/triocfd/files/

See recent research publications related to TrioCFD project:

- Web site: http://www-trio-u.cea.fr

Hot line

- **triou@cea.fr**

TRUST/TrioCFD support (2/2)

- A release every 6 months:
  - Linux version only
  - Installed on several CEA clusters, TGCC/CCRT and CINES
  - Installation by users or TRUST/TrioCFD support

- Documentation available under **$TRUST_ROOT/doc/TRUST** directory else ask it to CEA project leaders:

TRUST_and_TrioCFD_presentation.pdf	(these slides)
TRUST_tutorial.pdf	TRUST/TrioCFD/Meshing exercises
Models_Equations_TRUST.pdf	“Methodology for incompressible single phase flow”
Best_Practice_TRUST.pdf	“Validation of Trio_U code”
TRUST_Generic_Guide.pdf	“User Manual TRUST/TrioCFD”
HowTo_Validation.pdf	“Organisation of TrioCFD validation data base”
Manuel_Xprepro.pdf	“User Manual Xprepro”
Developer_TRUST_presentation.pdf	TRUST development Presentation