
Trio_U 1.7.0 user's training session

Trio_U 1.7.0 user's training session

- Presentations
 - Trio_U (1st day AM)
 - Mesh with tools (Xprepro, Salomé or Gmsh) (2nd day AM)
 - Automated test case with Trio_U (2nd day AM)
- Practice:
 - Trio_U (1st day PM/2nd day PM)
 - Xprepro/Salomé/Gmsh (2nd day AM)
- We provide also:
 - Advanced Trio_U user sessions (optional, 3rd day)
 - Trio_U developer sessions (optional, 4th and 5th days)



Céline CAPITAINÉ (CS) - Marthe ROUX (CS)
Trio_U support team
triou@cea.fr



Table of contents













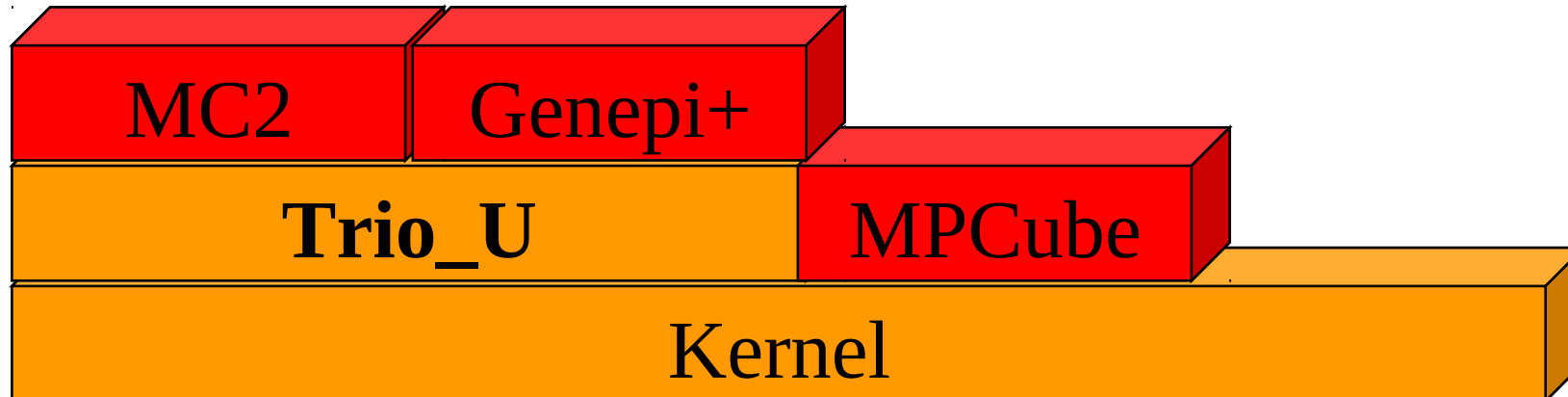
• Trio_U historic.....	p4	
• Modeling flow with Trio_U.....	p8	
• Examples of performed calculations.....	p16	
• Models, schemes, numerical methods.....	p19	
• Data files & calculation.....	p42	
• Command lines.....	p65	
• Parallel calculation.....	p71	
• Recommendations.....	p89	
• Mesh generators: Xprepro/Salomé/Gmsh.....	p106	
• Examples of data files.....	p134	
• Automating validation test case.....	p150	
• Trio_U support.....	p162	

Table of contents

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

Trio_U

- CFD code for incompressible monophasic/diphasic flow
- Developed at the CEA/DEN/DANS/DM2S/STMF/LMSF laboratory
 - Lab's chief: anne.burbeau@cea.fr
 - Project leader: gauthier.fauchet@cea.fr
- Trio_U, a software brick used by other CEA apps:



Trio_U project

- 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, //...)

Trio_U project

- 1994: start of the project
- 01/1997 : v1.0 (VDF only)
- 06/1998 : v1.1 (VEF version)
- 04/2000 : v1.2 (parallel version)
- 07/2001 : v1.3 (compressible version)
- 11/2002 : v1.4 (new LES turbulence models)
- 02/2006 : v1.5 (VDF/VEF Front tracking)
- 10/2009 : v1.6 (Data structure revamped)
- 12/2014 : v1.7

Table of contents

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

Trio_U : Modeling flow (1/7)

- 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:
 $\rho = \rho(T) \sim \rho_0 - \beta(T - T_0)$
 - Quasi-compressible model:
 $\rho = \rho(P, T)$ for low mach numbers

$$\text{Div}(\vec{u}) = 0$$

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

$$P^* = \frac{P}{\rho} + gz$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T = \text{Div}(\alpha \nabla T)$$

Trio_U : Modeling flow (2/7)

Description of the Quasi Compressible model used

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

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

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

$$\text{Total pressure: } P(\vec{x}, t) = P_0(t) + P_1(\vec{x}, t) \quad \begin{array}{l} \text{Thermodynamic pressure : } P_0(t) \\ \text{Hydrodynamic pressure : } P_1(x, t) \end{array}$$

$$\text{With } P_1 \approx M^2 P_0 \text{ and } M = \text{Mach} \ll 1$$

- Set of equations solved:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \text{div}(\rho \vec{u}) &= 0 \\ \frac{\partial}{\partial t}(\rho u_i) + \text{div}(\rho u u_i) - \sum_{j=1}^N \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + \frac{\partial P_1^*}{\partial x_i} &= -\rho g_i \\ P_1^* &= P_1 + \frac{2}{3} \mu \text{div}(\vec{u}) \end{aligned}$$

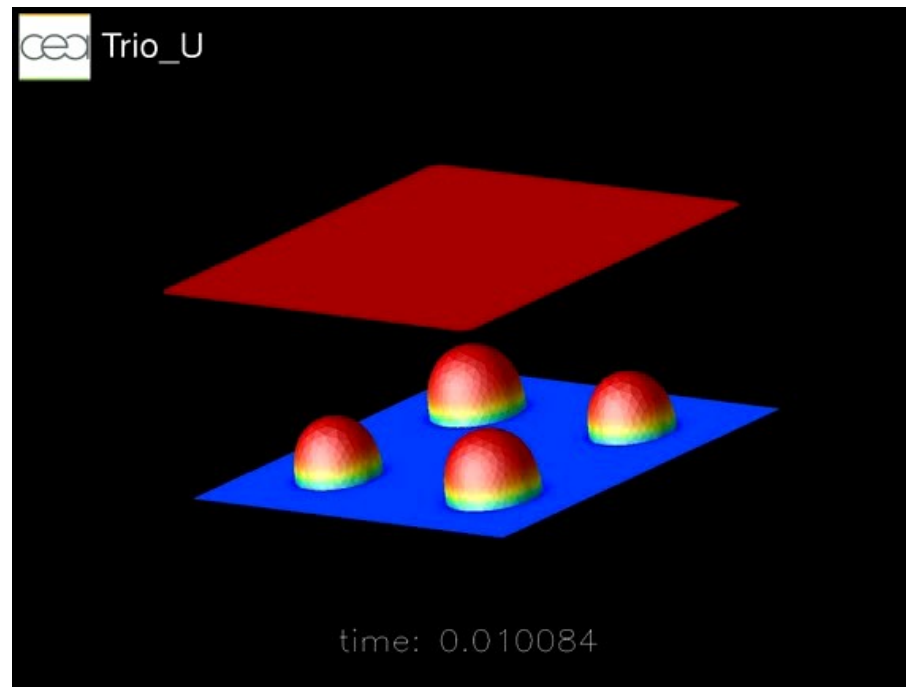
$$P_0 = \rho R T$$

$$\rho C_p \frac{dT}{dt} - \sum_{j=1}^N \frac{\partial}{\partial x_j} \left(K \frac{\partial T}{\partial x_j} \right) = Q + \frac{dP_0}{dt}$$

Trio_U : Modeling flow (3/7)

Two phases flow (Front tracking model)

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

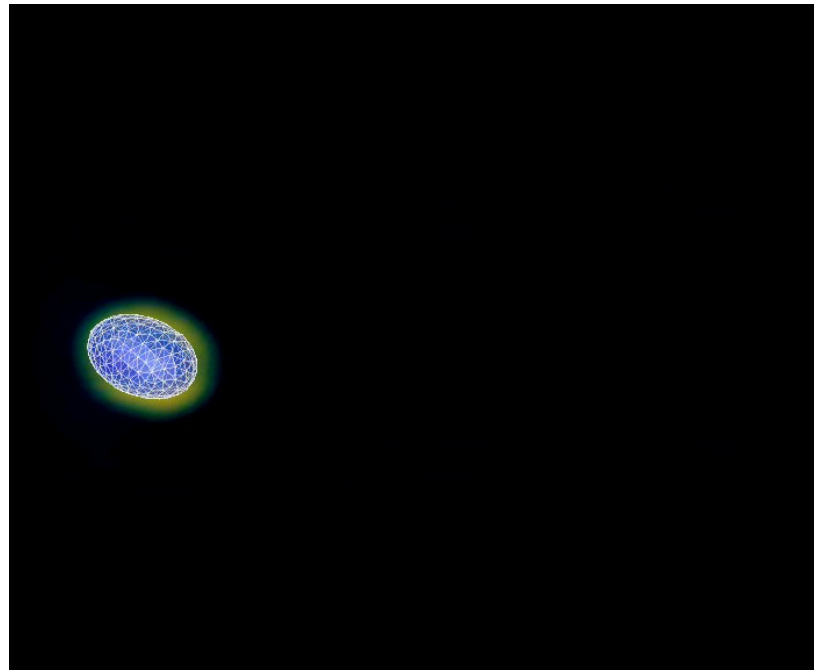


Boiling bubbles

Trio_U : Modeling flow (4/7)

Front tracking model

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



Example of a flow around a rotating body defined by an IBC

Trio_U : Modeling flow (5/7)

- 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

Trio_U : Modeling flow (6/7)

- Heat exchange
 - Conduction
 - Radiation in transparent medium
 - Radiation in semi-transparent medium

- Transport of passive scalars $\frac{\partial C_i}{\partial t} + \vec{u} \nabla C_i = \text{Div} (D_i \nabla C_i)$

- Porous Media
 - Surface or volume porosities
 - Singular or regular pressure loss

Trio_U : Modeling flow (7/7)

- 2D calculation
 - Plane, Cartesian coordinates (x,y)
 - Axi-symmetric, coordinates (r,z) (VDF only)
- 3D calculation
 - Cartesian coordinates (x,y,z)
 - Cylindrical coordinates (r,θ,z) (VDF only)
- Transient flow calculation calculated by:
 - Explicit or semi implicit time schemes
- Steady state calculation obtained:
 - Either by convergence of the transient flow
 - Or by using an implicit time scheme

Table of contents

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

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)

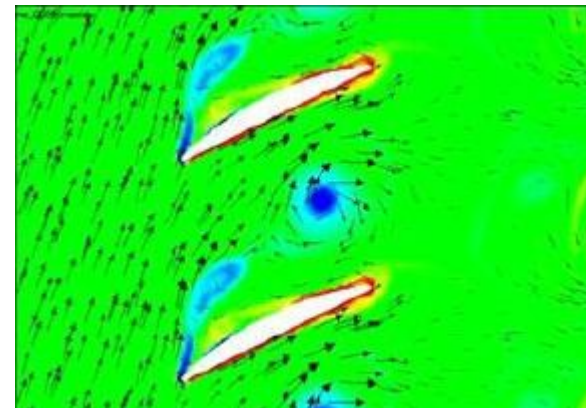
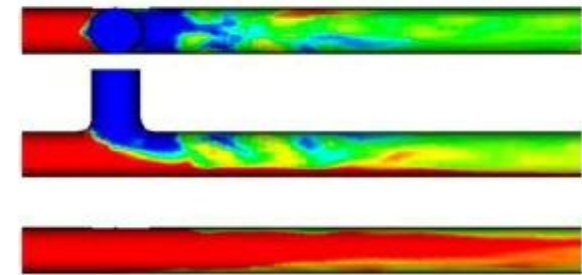
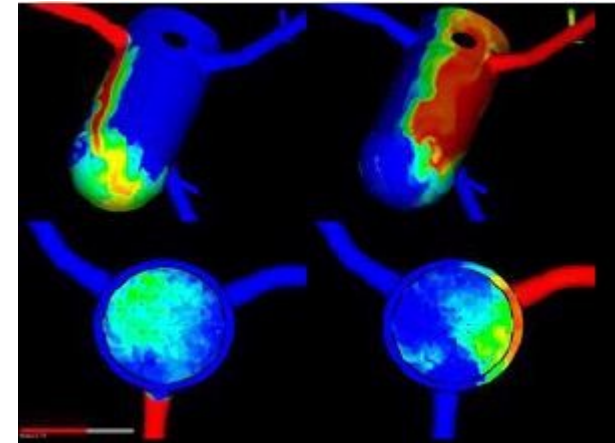


Table of contents

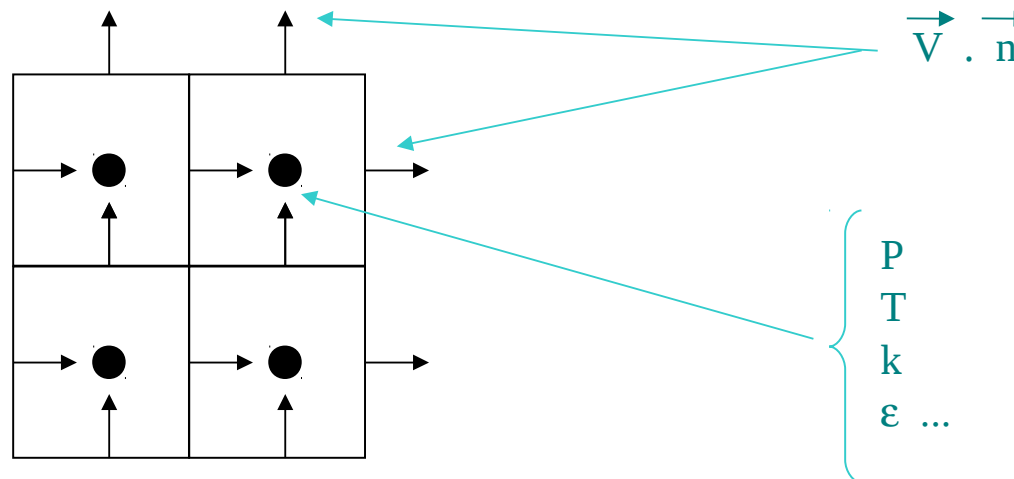
- Trio_U historic
- Modeling flow with Trio_U
- Examples of performed calculations
- **Models, schemes, numerical methods**
- Data files & calculation
- Command lines
- Parallel calculation
- Recommendations
- Mesh generators: Xprepro/Salomé/Gmsh
- Examples of data files
- Automating validation test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

Trio_U description

- **Discretizations (VDF/VEF).....p21**
- Time and space schemes.....p24
- Boundaries conditions.....p27
- Source terms.....p31
- Solvers for linear systems.....p35
- Turbulence models.....p39

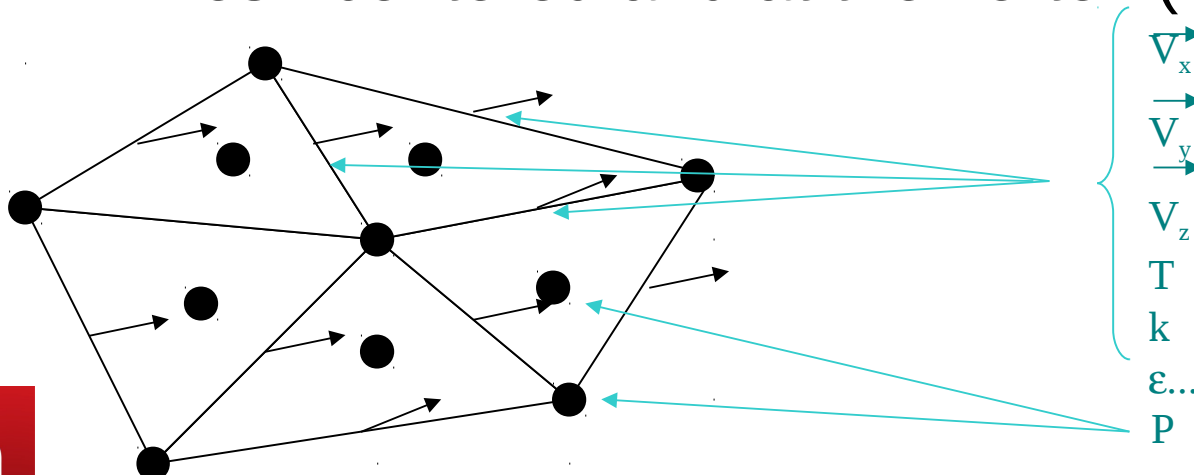
Discretizations (1/3)

- **Finite Differences Volumes (VDF)**
 - Structured meshing ; velocity and pressure are shifted
 - Scalar unknown is mesh centred
 - Velocity is normal at the face



Discretizations (2/3)

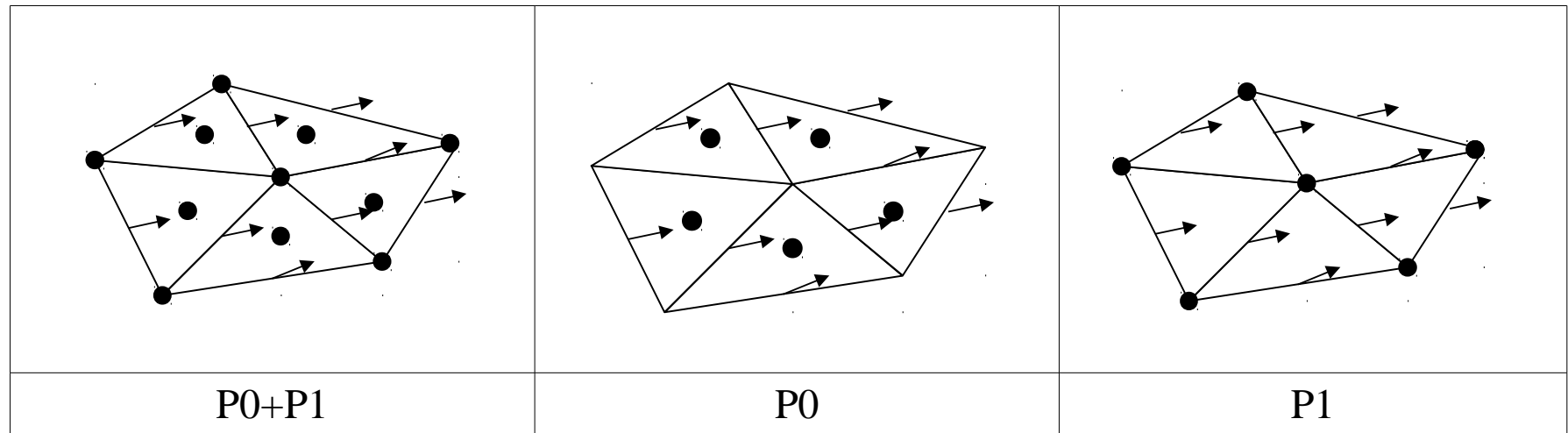
- **Finite Elements Volumes (VEF)**
 - Unstructured meshing triangles (2D) or tetrahedrons (3D)
 - Unknown fields are face centered (P1NC)
 - Physical characteristics are cell centered
 - Pressure :
 - Mesh centered and at the vertex (P0+P1)



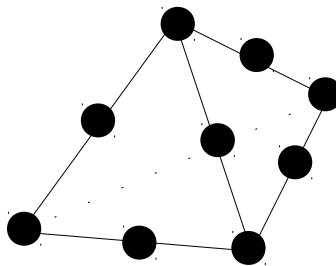
Discretizations (3/3)

- **Finite Elements Volumes (VEF)**

- By default, $P0+P1$ for pressure but less/more pressure nodes is possible:



Plus in 3D: $P0+P1+Pa$



11 pressure nodes per tetra:

-1 in center (P0)

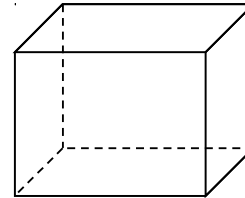
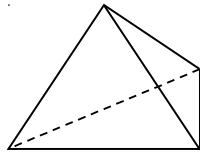
-4 on vertexes (P1)

-6 on edges (Pa)

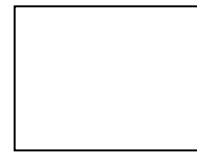
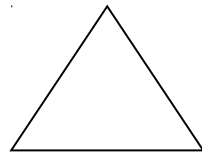
-> **Useful** for flow with a strong source term & a low velocity field where $P0+P1$ pressure gradient $P0+P1$ has trouble to match the source term

Kind of mesh permitted

- Tetrahedral or hexahedral meshing for 3D cases

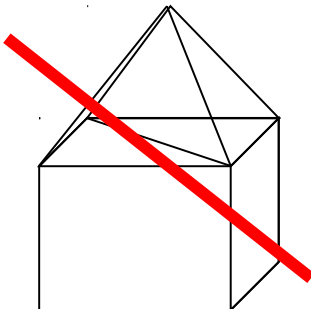
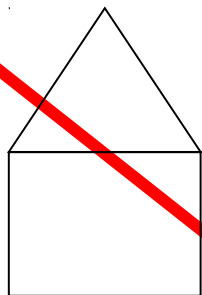


- Triangular or quadrangular meshing for 2D cases

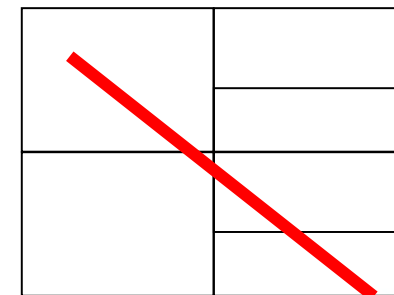


- Hybrid or non standard not supported

Hybrid

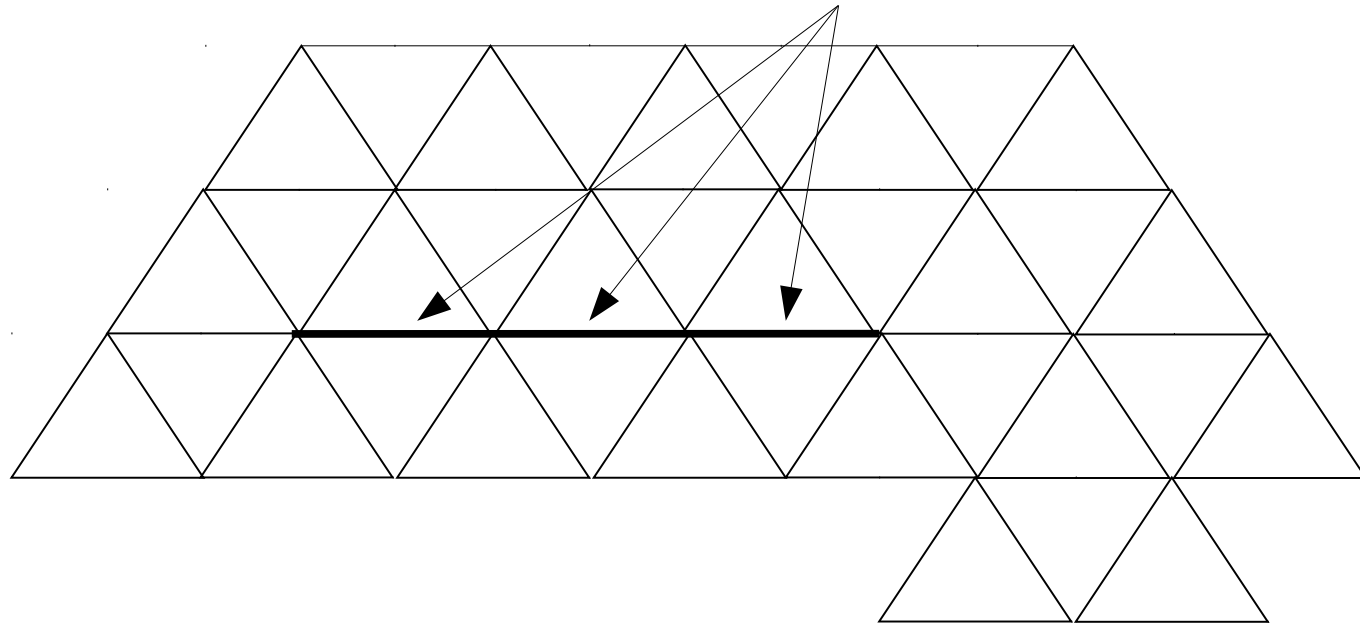


Non standard



Kind of mesh permitted

- Internal boundaries partially supported :
 - With ICEM, split faces in two and define boundaries
 - Trio_U will differentiate the two faces



Trio_U description

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

Time schemes

- Explicit schemes:

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

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

- Semi-implicit scheme:

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

- Euler semi-implicit (diffusion implicit)

- Implicit schemes (not unconditionally stable):

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

- Implicite, PISO, SIMPLE (dynamic time step)

Convection schemes

- VDF
 - Quick (order 2-3)
 - Centre (order 2 or 4) « centered »
 - Amont (order 1) « upwind »
- VEF
 - EF_stab (order 2) « centered stabilized »
 - Muscl (order 2) « quick like »
 - Amont (order 1) « upwind »

Trio_U description

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

Available boundary conditions (Momentum)

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

Available boundary conditions (Energy)

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

Available boundary conditions

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

Trio_U description

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

Source terms

- Navier Stokes equation:

- Boussinesq

$$S = \rho_0 g \beta (T - T_0)$$

- Useful for small variation of volumic mass

- Flow rate

$$S = Q_m$$

- Pressure loss

$$S = -0.5 \rho C_f U |U| / D$$

- Regular pressure loss (Blasius or C_f given by the user)

- Periodic channel

$$S = Q_m$$

- Useful to keep constant flow rate into a periodic channel

- ...

Source terms

- 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 , $d\Omega/dt$: 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

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

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

Source terms

- Energy equation:

- Volumic heat power

$$S=P$$

- For example into a solid media

-

- Concentration equation:

- Boussinesq

$$S=\rho_0 g \beta (C-C_0)$$

- Useful to build a two miscible fluids calculation (if the volume mass is similar). Concentration C will be equal to the fraction fluid

Trio_U description

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

Trio_U linear systems

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

Solvers for linear systems

- Trio_U 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 Trio_U versions

Solvers for linear systems

- Iterative solvers (GC, GMRES,...)
 - Need a tolerance ε to be defined : $\|Ax-B\| < \varepsilon$
 - Possible pitfall because it is an absolute (not a relative) value in Trio_U
 - So, check the balances!
 - Exemple: Solving pressure system for an incompressible flow $\Leftrightarrow \text{Div}(u)=0$
 - So, check the flow rate error in .out file
- Direct solvers (PETSc Cholesky)
 - Use it if possible

Trio_U description

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

Turbulence models

- 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](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

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

Trio_U data file description

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

Data file objects definition

Domaine my_domain

Pb_Thermohydraulique my_problem

Schema_Euler_Explicite my_scheme

...

Trio_U data file description

- Action 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 #

Data file example (Incompressible flow)

```
# Hydraulique 2D laminar flow with Quick scheme #
dimension 2

Pb_hydraulique pb
Domaine dom

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

# BEGIN PARTITION
Partition dom
{
    Partitionneur tranche { tranches 2 1 }
    Larg_joint 2
    Nom_Zones DOM
}
Fin
END PARTITION #

# BEGIN SCATTER
Scatter DOM.Zones dom
END SCATTER #

# A discretization is selected #
VDF ma_discretisation

Schema_Euler_explicite mon_schema
Read mon_schema
{
    tinit 0
    tmax 5.0
    nb_pas_dt_max 1000
    dt_min 5.e-3 # Trio_U stops if dt_stab<dt_min #
    dt_max 5.e-3
    dt_impr 5.e-3
    dt_sauv 1.
    facsec 0.5
    seuil_statio 1.e-8
}
# Calculation timestep dt=min(dt_stab,dt_max)*facsec #
# dt_stab=1/(1/dt(convection)+1/dt(diffusion)) #
# 1D : dt(convection)=dx/max(U) dt(diffusion)=dx^2/(2D) #
```

```
# A media is defined #
Fluide_Incompressible milieu
Read milieu
{
    mu Champ_Uniforme 1 3.7e-05 # Dynamic viscosity #
    rho Champ_Uniforme 1 2 # Volumic mass #
}
# Create links between objects #
Associate pb dom
Associate pb mon_schema
Associate pb milieu
Discretize pb ma_discretisation
Read pb
{
    Navier_Stokes_standard
    {
        convection { quick }
        diffusion { } # By default, 2nd order scheme #
        conditions_initiales { vitesse Champ_Uniforme 2 0. 0. }
        boundary_conditions {
            Square paroi_fixe # Wall U= 0 #
            Upper symetrie
            Lower symetrie
            # Neumann boundary condition P=0 #
            Outlet frontiere_ouverte_pression_imposee
                Champ_front_Uniforme 1 0.
            # Dirichlet boundary condition U=(1,0) #
            Inlet frontiere_ouverte_vitesse_imposee
                Champ_front_Uniforme 2 1. 0.
        }
        solveur_pression GCP {
            # Parameter of SSOR #
            precondition { omega 1.5 }
            # Convergence threshold which impacts flow mass rate #
            # Warning: not a dimensionless number so the flow mass #
            # rate should be checked during the first time-steps #
            seuil 1.0e-06
            impr # Print the residual error ||Ax-B|| #
        }
    }
}
```

Data file example (Incompressible flow)

```

Postraitement
{
  # Definition of probes for 1D plot #
  Sondes
  {
    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_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
  }
  format lata fichier results # Instantaneous fields and time averaged fields written into a LATA format results.lata file #
  # Definition of 2D/3D instantaneous fields for post-processing #
  Champs dt_post 1.
  {
    # Fields location #
    pression elem
    pression som
    vitesse elem
    vitesse          # By default, location is som (vertexes) #
    vitesse faces
  }
  # Definition of 2D/3D statistical (time averaged) fields for post-processing #
  Statistiques dt_post 1.
  {
    t_deb 1. t_fin 5.
    moyenne vitesse elem
    ecart_type vitesse som
    moyenne pression elem
    ecart_type pression som
  }
}
Solve pb      # Solve the problem : Start the time scheme loop #
End           # Optional keyword end #

```

Data file example (Obstacle.geo)

Mesh 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

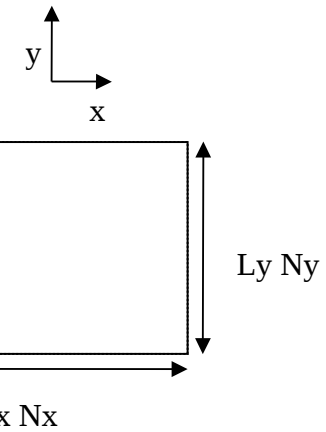
```
{
  Origine 0.1 0.12
  Nombre_de_Noeuds 2 11
  Longueurs 0.02 0.1
}
{
  bord Upper Y = 0.22 0.1 <= X <= 0.12
  bord Square Y = 0.12 0.1 <= X <= 0.12
},
```

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 O_x O_y [O_z]

Nombre_de_noeuds: Nodes number N_x N_y [N_z]

Longueurs: Lengths L_x L_y [L_z]

Bord: Boundary

Operations on meshes

Trio_U keywords:

- **Dilate** (to change the size of a mesh)
- **Mesh** (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)
- ...

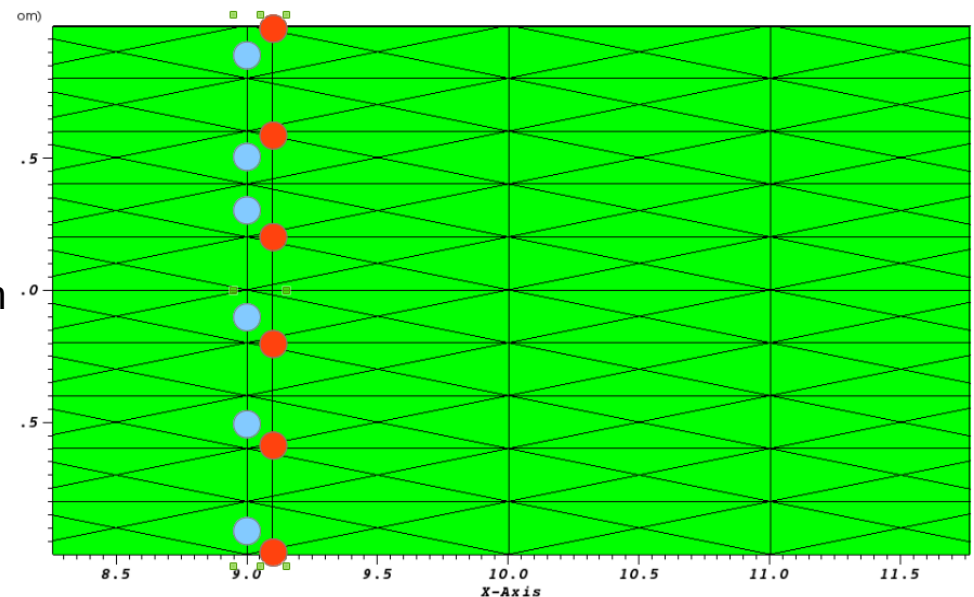
Probes : "Nodes" option

"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.

Possible post processed fields

- Basic 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**

- Advanced 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 Trio_U and an analytical solution

Maximal value of a field

```
Read pb {  
  ....  
  Postraitement {  
    Definition_champs {  
      # Creation of the 0D field: maximal temperature of the domain #  
      temperature_max Reduction_0D {  
        methode max  
        source refChamp { Pb_champ pb temperature }  
      }  
    }  
    Sondes {  
      # Print max(temperature) into the datafile_TMAX.son file #  
      tmax temperature_max periode 0.01 point 1 0. 0. 0.  
    }  
    Champs dt_post 1.0 { ... }  
  }  
}
```

Averaging a field on a boundary

Dimension 3

Domaine dom # 3D domain with a boundary named wall #

....

Domaine surface # 2D domain object created for use by the Extraction keyword #

Read problem {

...

Postraitement {

Definition_champs {

Creation of the 0D field: mean temperature on the boundary wall

 wall_mean_temperature **Reduction_0D** {

methode moyenne

source Extraction {

domaine surface **nom_frontiere** wall **methode** trace

source refChamp { **Pb_champ** problem **temperature** }

 }

 }

 }

Print into the datafile_TWALL.son file

Sondes { twall wall_mean_temperature **periode** 0.01 **point** 1 0. 0. 0. }

Champs dt_post 1.0 { ... }

} Trio_U 1.7.0 user's training session

Calculating an error between fields

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

Trio_U data file description

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)
 - don_lu** (field read in a file)
 - fonc_MED** (read a MED field)
- 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)
 - ...

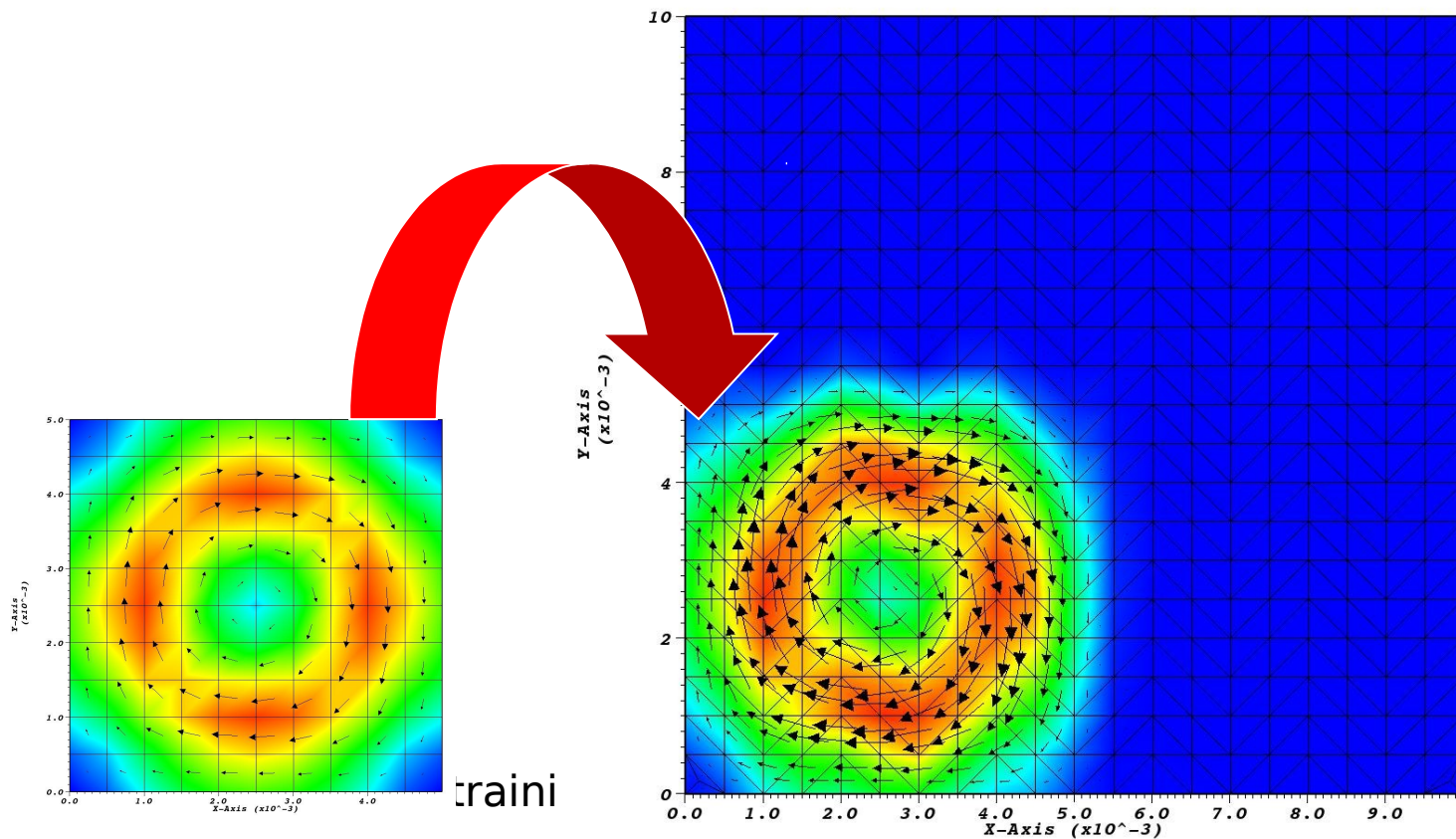
Example of Champ_Fonc_MED

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:

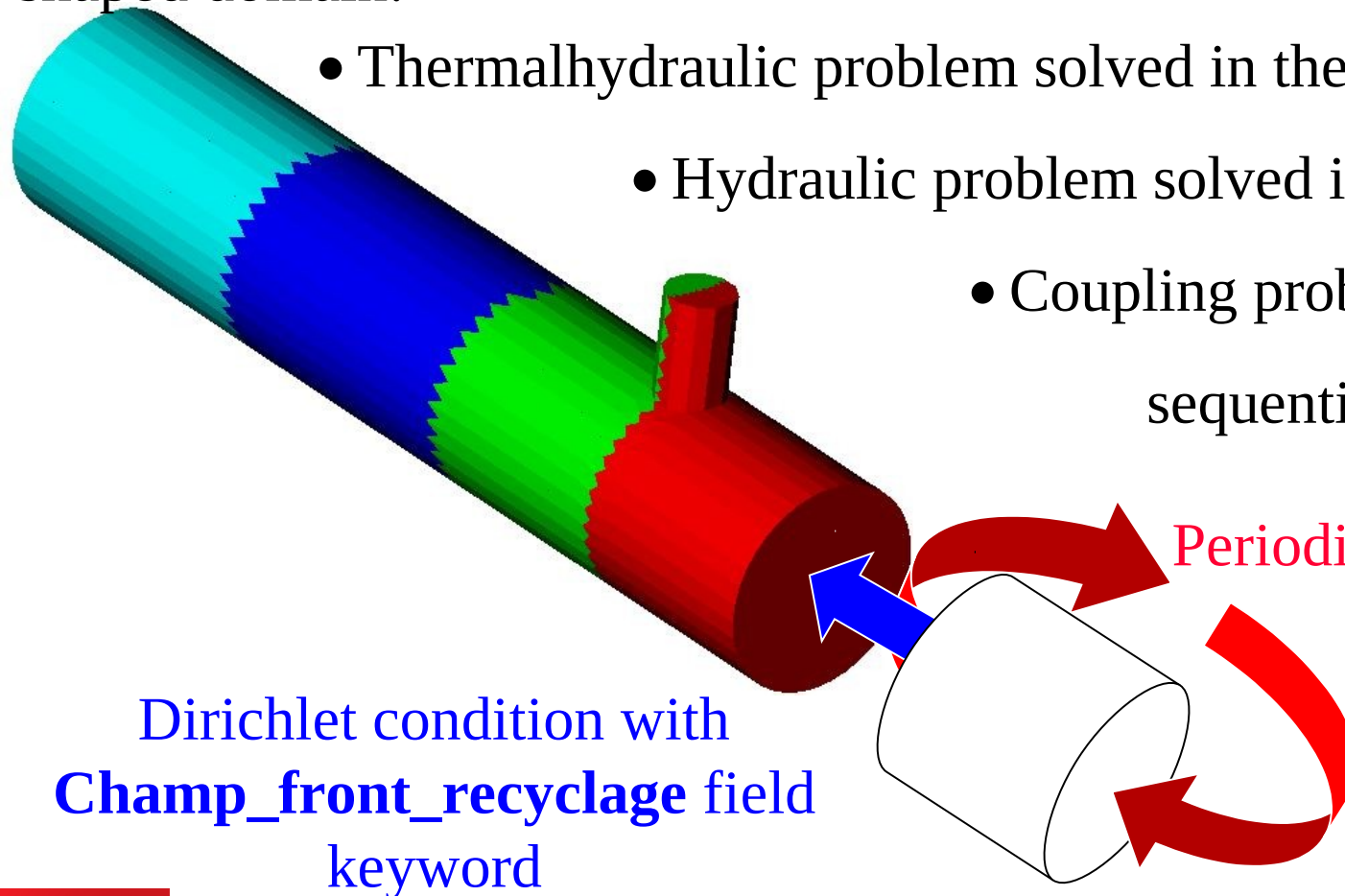
```
conditions_initiales { vitesse Champ_Fonc_MED last_time VDF_field.med domain vitesse elem 0 }
```



Example of Trio_U 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.



Trio_U data file description

Formulas for a field in a data file:

Champ_front_fonc_txyz 2 **cos**(y+x^2) t+ln(y)

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

- : subtracte

/ : division

* : multiplication

% : modulo

\$: max

^ : power

< : lesser than

> : greater than

[: less or equal to

] : greater of equal to

Trio_U 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)
 - 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 a specific keyword
- 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|formatte|xyz filename.sauv|filename.xyz

With:

- » **sauvegarde_simple** : the file is deleted before the save
- » **formatte** : the format of the file ASCII instead of binary
- » **xyz**: the .xyz file is written instead of the .sauv files

Trio_U 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|formatte|xyz *filename.sauv|filename.xyz*

Trio_U files summary

- Input :
 - Data file : .data
 - Meshing : .geom
 - 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
 - CPU performances : .TU
 - Time steps, facsec, equation residuals : .dt_ev
 - Stop file (0 or 1) : .stop

Trio_U results export

2D/3D results files are readable:

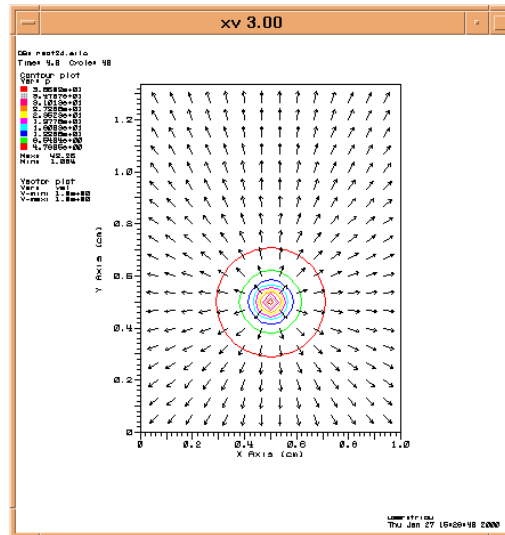
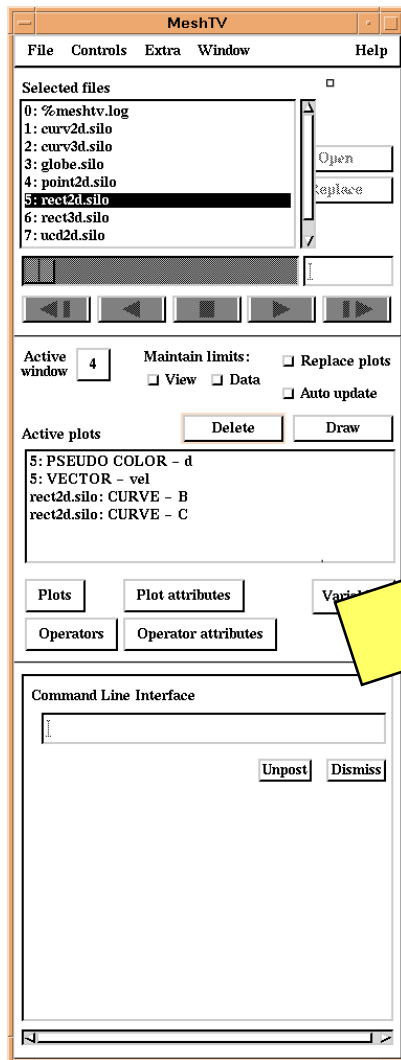
- Either directly by :
 - VisIt (use *lata* format in the data file “**format lata**”)*
 - Salomé (use *med* format in the data file “**format med**”, to post-process use “ParaVis” in Salomé and open med file with “builtin”)
- Or after read then export by VisIt (VTK format):
 - Paraview, Tecplot

1D results files by :

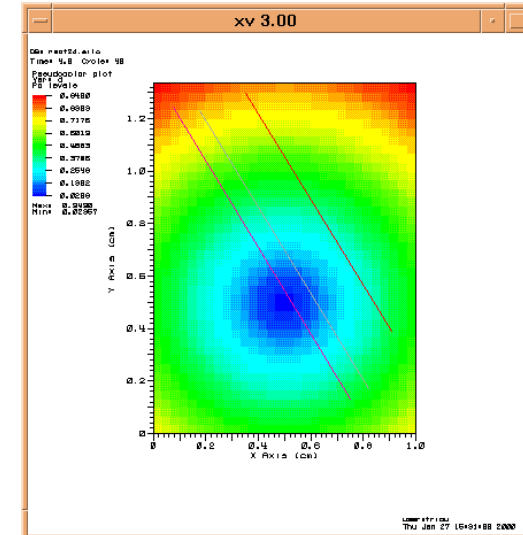
- Gnuplot, XmGrace, Excel

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

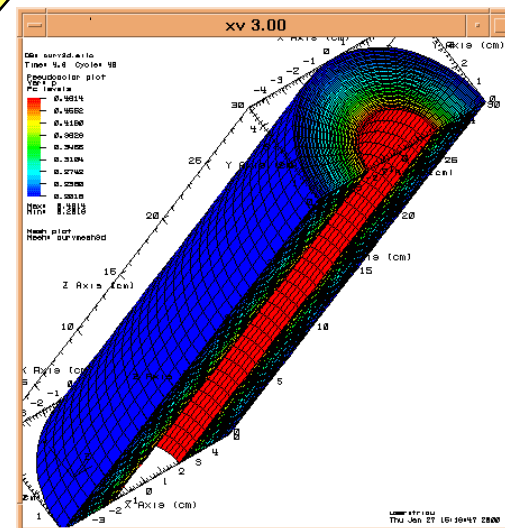
Trio_U with VisIt



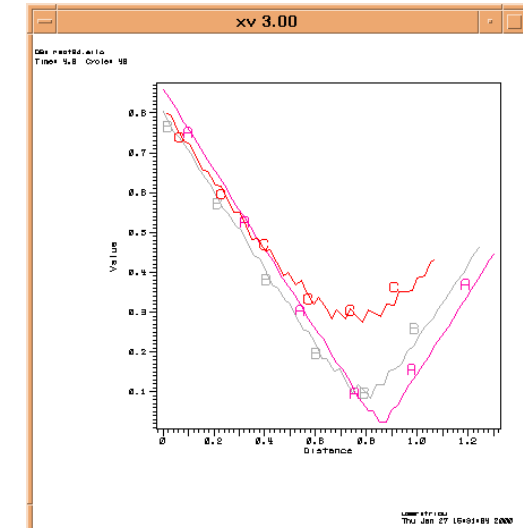
Vectors/Scalars



2D Slice

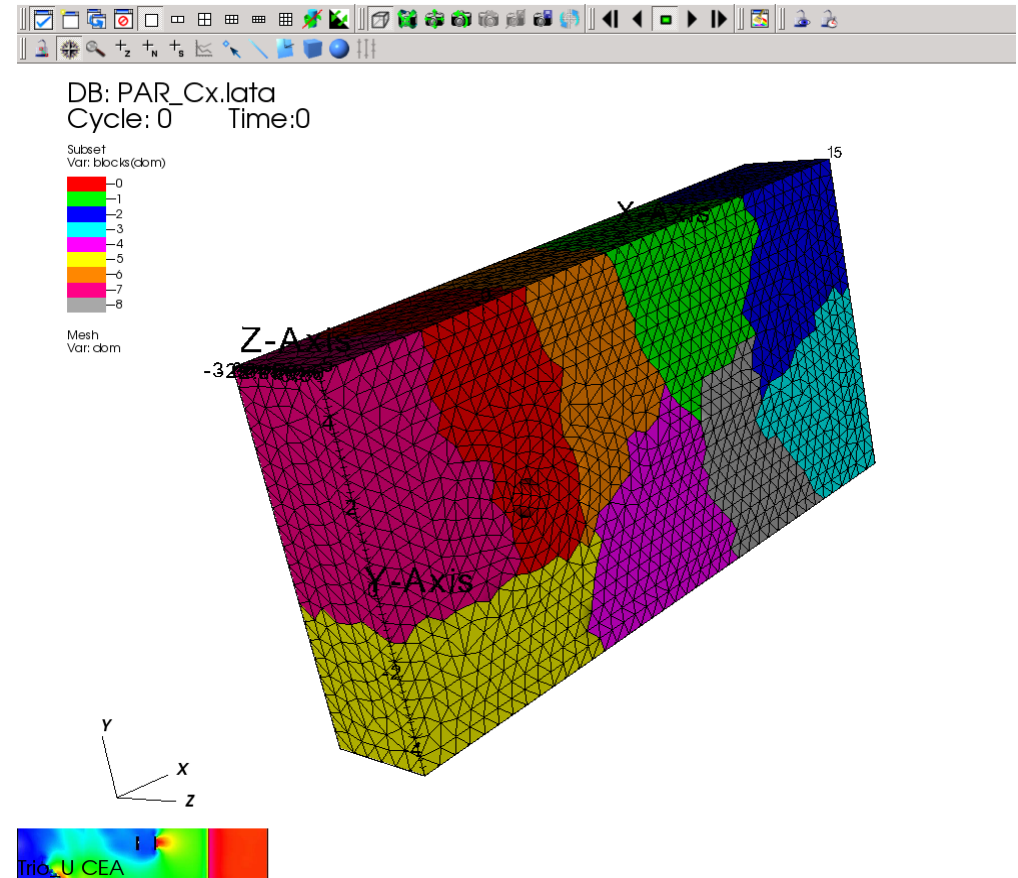
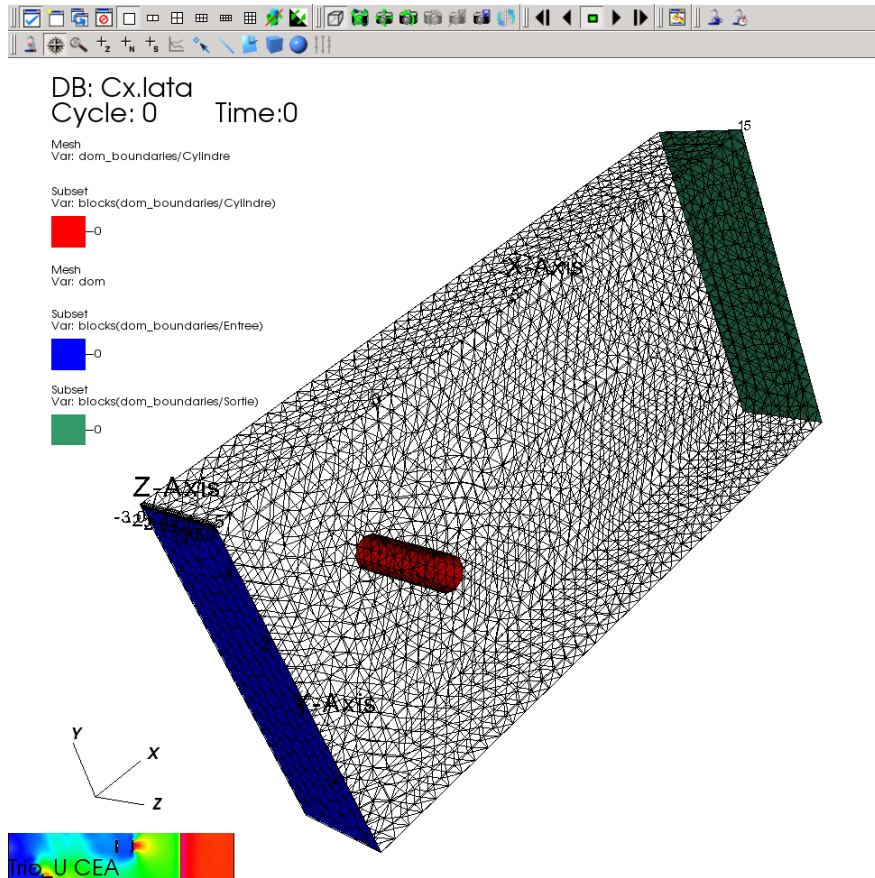


3D View



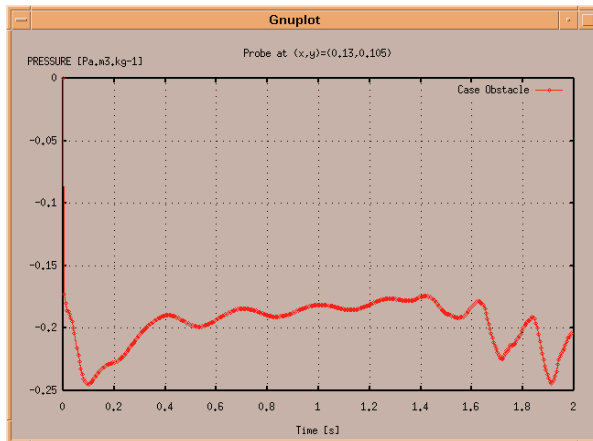
1D Cut

Trio_U with VisIt

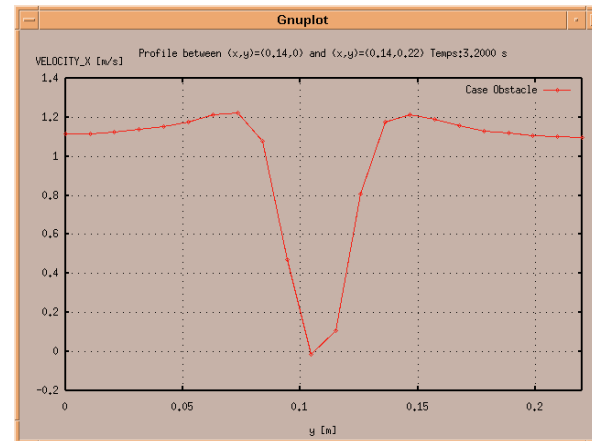


Trio_U with Gnuplot

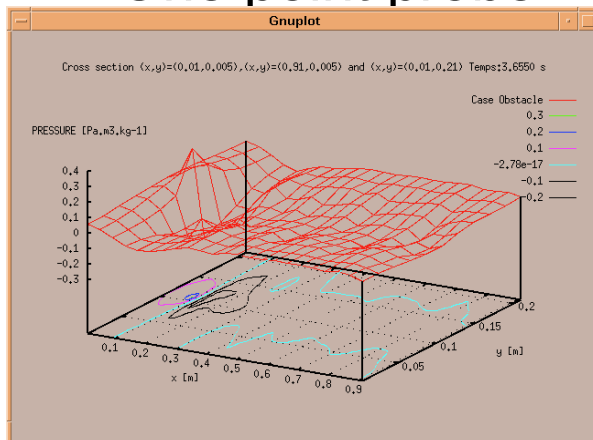
Real-time display of calculated quantities:



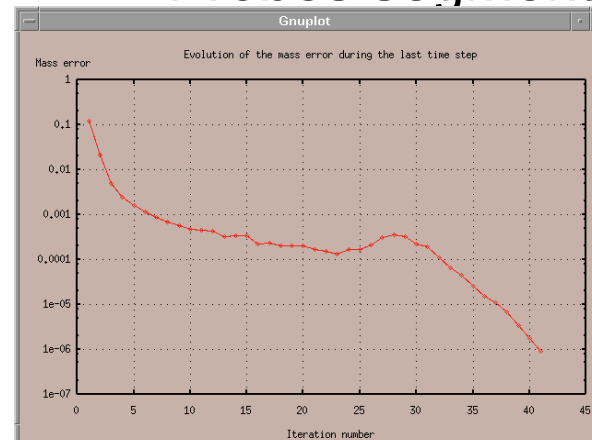
One point probe



Probes segment



Probes plane



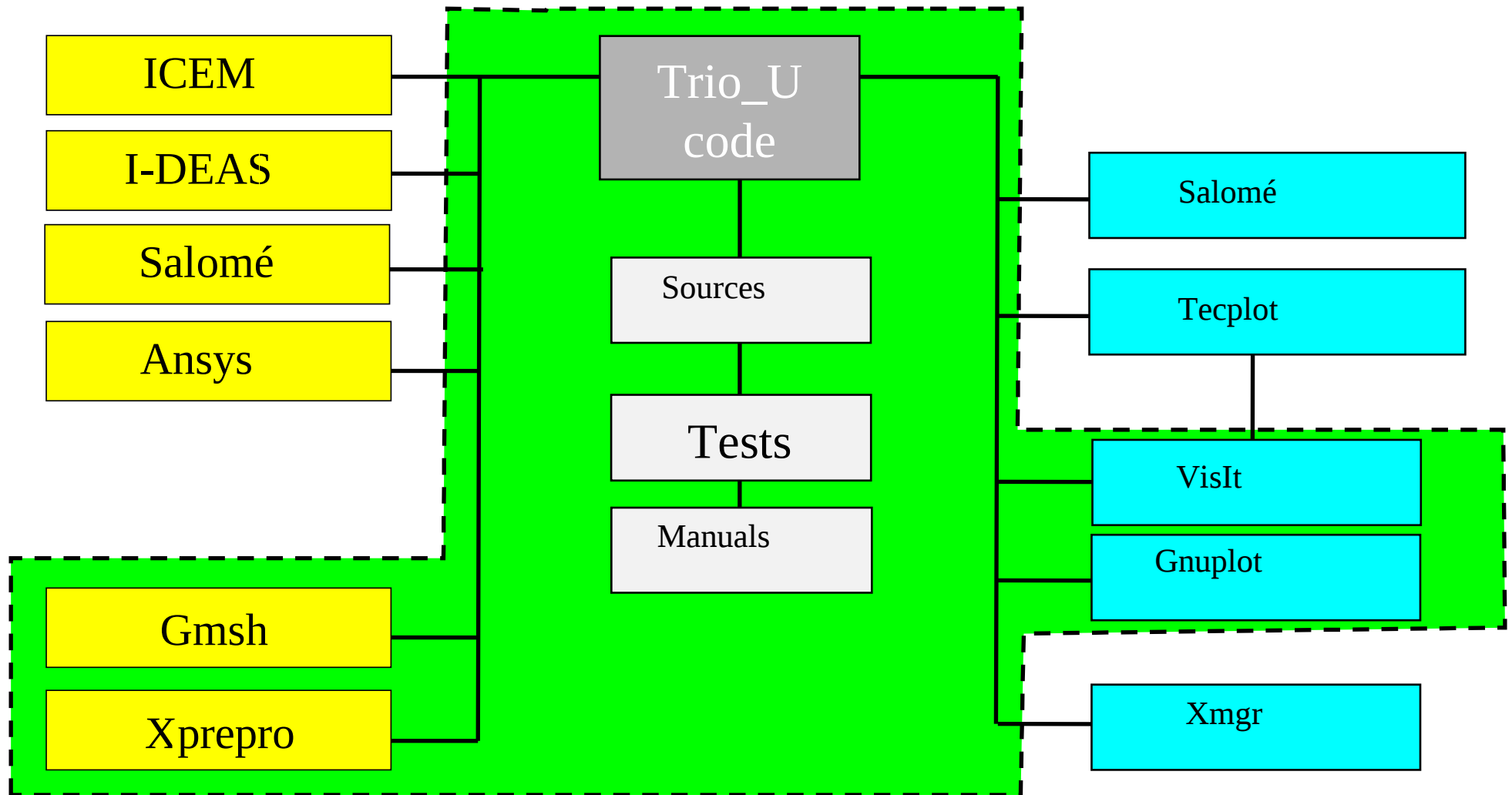
Convergence

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

- flow rate
- pressure forces
- viscous forces
- heat flux

...

Trio_U and interfaces between tools



Trio_U package released and supported

Table of contents

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

Trio_U command lines

- Trio_U environment initialization:
source \$TRIO_U_ROOT/bin/Init_Trio_U
- To run a Trio_U calculation with the triou script:
 - Sequential run:
triou datafile
 - Parallel run on N CPUs:
 - Partitioned mesh partitioned should be created sequentially, then interactively:
triou 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:
triou 1>datafile.out 2>datafile.err

Trio_U command lines

- To copy a data file from the test database:
triou -copy datafile
- To edit interactively (change/add schemes, solvers...) a data file with:
EditData datafile.data
- Check a data file without running Trio_U:
VerifData datafile.data
- To visualize the mesh and its boundaries used by a data file:
triou -mesh datafile[.data]
- To run and follow probes during your calculation (used of “Run_sonde” script):
triou -monitor datafile[.data]
- To monitor only your calculation:
triou -probes [datafile[.data]]

Trio_U command lines

- To run VisIt with a LATA results file:
visit -o datafile.lata
- To clean your calcul directory:
triou -clean
- To open the PDF documentation (User's manual):
triou -doc
- To browse some useful resources (PDF manuals, test cases, keywords, C++ classes,...) :
firefox \$TRIO_U_ROOT/index.html
- In all cases, the Trio_U binary may be changed by the **\$exec** variable and by default, `exec=$TRIO_U_ROOT/exec/Trio_U_mpi_opt`

Visit demonstration

Demonstration of the Visit tool.

For more informations and to download manuals see :

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

Practice

Exercise 1 : Incompressible 2D flow

→ Obstacle.data

Table of contents

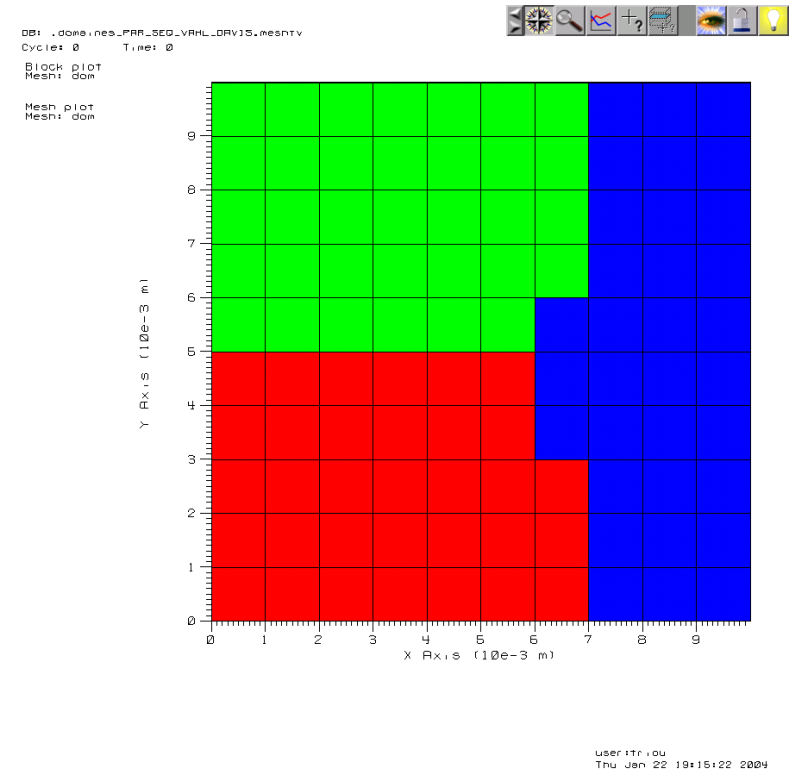
- Trio_U historic
- Modeling flow with Trio_U
- Examples of performed calculations
- Models, schemes, numerical methods
- Data files & calculation
- Command lines
- **Parallel calculation**
- Recommendations
- Mesh generators: Xprepro/Salomé/Gmsh
- Examples of data files
- Automating validation test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

Parallel calculations with Trio_U (1/4)

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

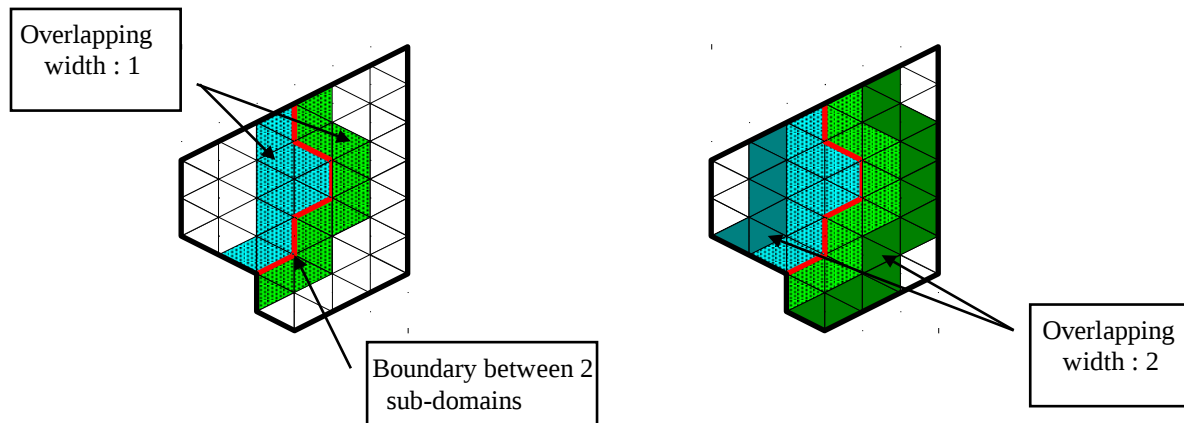
Parallel calculations with Trio_U (2/4)

- 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)
- Some rules of thumb for performance
 - If possible, use 20000-30000 cells per process
 - Look at cluster specificities:
 - L2 size cache
 - Latency network
 - ...



Parallel calculations with Trio_U (3/4)

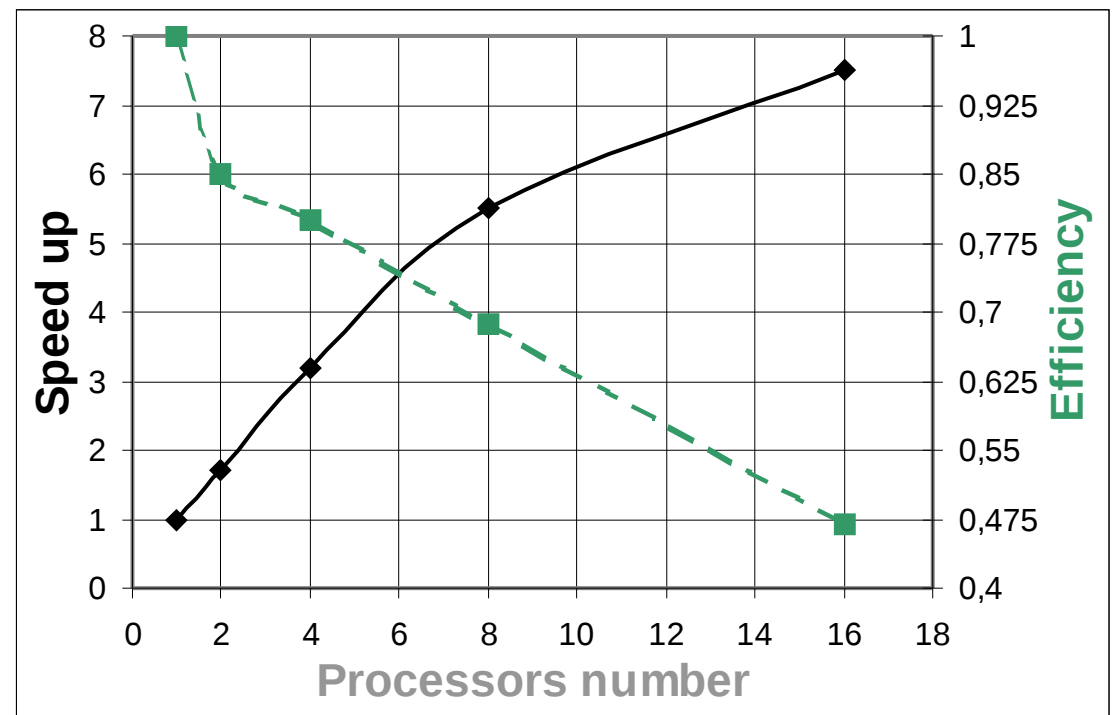
- Definition of overlapping width value
 - Number of vertexes or elements on the remote sub-domain known by the local sub-domain
 - Specified by the users during partitioning task
 - This value depends on the space scheme orders:
 - 1 if 1-2nd order
 - 2 if 3-4th order
 - In practice, use 2 except if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



Parallel calculations with Trio_U (4/4)

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

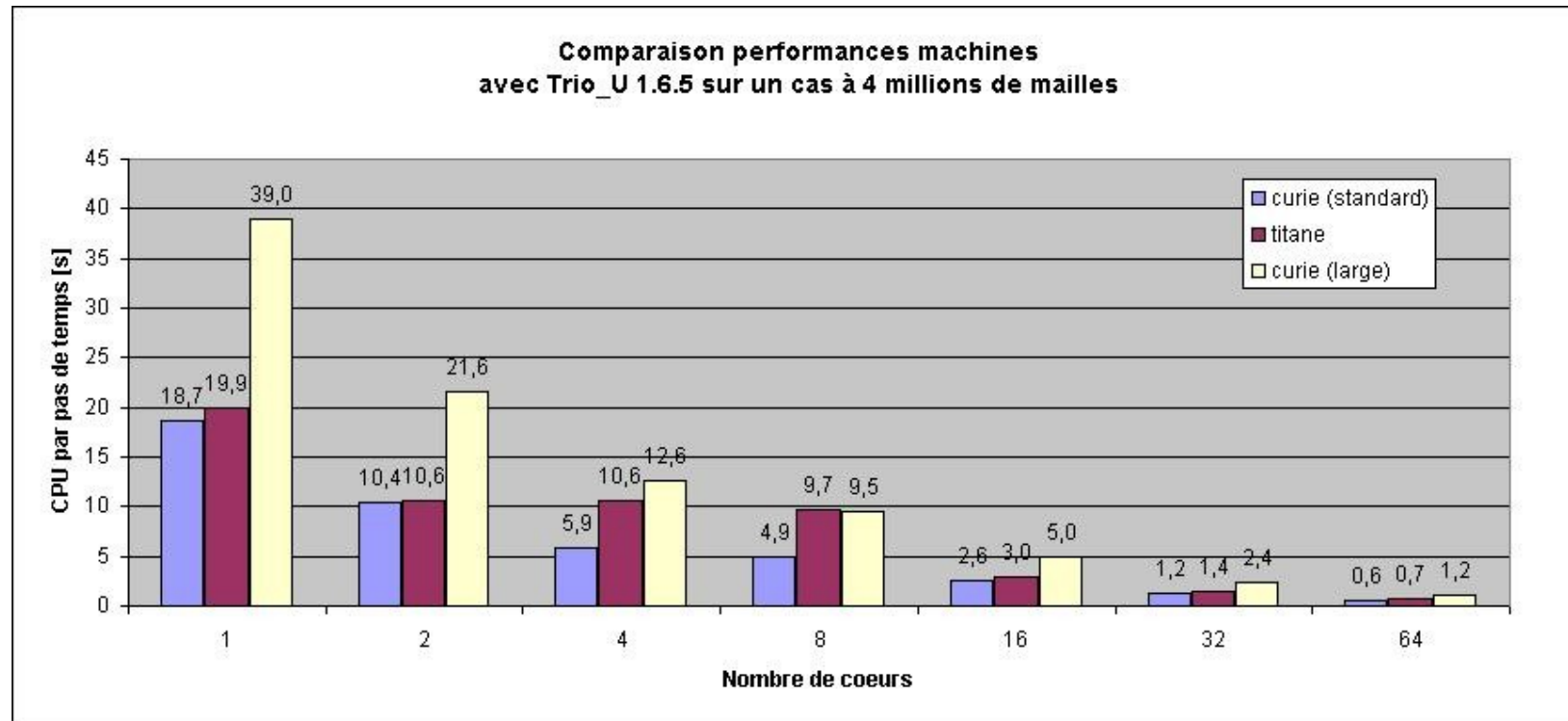
Processors	Speed Up	Efficiency
1	1	1
2	1.7	0.86
4	3.2	0.80
8	5.5	0.69
16	7.5	0.47



Parallel calculation on clusters

- To connect to:
 - CEA/DM2S service cluster: callisto (1168 cores)
 - CEA/CCRT clusters: airain (~10000 cores)
 - CEA/TGCC cluster: curie (~92000 cores)
 - CEA/Cadarache cluster: mezel2 (~600 cores)
 - CINES cluster: occigen (~50 000 cores)
 - PRACE LRZ cluster : supermuc (~155000 cores)
- Ask for a login:
 - callisto (access for CEA only)
 - CCRT <http://www-ccrt.cea.fr>
 - TGCC <http://www-hpc.cea.fr/fr/complexe/tgcc.htm>
 - jade <http://www.cines.fr/spip.php?article524>
 - occigen <https://www.cines.fr/occigen-le-nouveau-supercalculateur/>
- Once you have the login, connect from your PC to:
 - `ssh -X login@name.intra.cea.fr` (name=callisto-login1 | callisto-login2)
 - `ssh -X login@name.ccc.cea.fr` (name=curie-ccrt | airain)
 - `ssh -X login@occigen.cines.fr`

Parallel calculation on clusters



Performance of the new CEA cluster curie compared to previous cluster titane

curie (large) : 5040 nodes of 2 eight cores @ 2.27Ghz **curie (standard)** : 360 nodes of 4 eight cores @ 2.3Ghz

Parallel calculation on clusters

- **Trio_U versions located on:**

- callisto ROOT=/panfs/ixion/home/triou
- CCRT/TGCC ROOT=/ccc/scratch/cont002/den/triou
- occigen ROOT=\$HOMEDIR (home de l'utilisateur)

With Trio_U stable versions:

- \$ROOT/Trio_U-X.Y.Z/Trio_U
- \$ROOT/Trio_U-X.Y.Z_patch/Trio_U -> X.Y.Z version plus important patches

And Trio_U version in development:

- \$ROOT/Version_test_clustername/Trio_U -> future X.Y.Z+1 version

- For **Trio_U** configuration, add in your ~/.profile (or ~/.bashrc) file, the lines:

Example of Trio_U environment on callisto:

source /panfs/ixion/home/triou/Trio_U-1.6.9/Trio_U/bin/Init_Trio_U 1>/dev/null 2>&1

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

Parallel calculation on clusters

- Space discs on callisto:
 - /home (limited space but there is backup)
 - /panfs/ixion/home (local space)
 - /data/tmpdm2s or /data/tmplepp (shared spaces)
- Space discs and quotas on CCRT/TGCC:
 - /ccc/cont002/home (3 GB, slow I/O, backup) -> Data shared by all clusters
 - /ccc/work (1 TB, fast I/O, no backup) -> Source code, binary (also shared)
 - /ccc/scratch (20 TB, fastest I/O, no backup) -> Data, code output
 - /ccc/store (infinite size, backup, but data are slow to access) -> Data archiving
- Space discs on occigen:
 - /home (40 GB/user, backup)
 - /store (200 GB/group, backup, limited to 100 000 files/group)
 - /scratch (4 TB/group, no backup)

Parallel calculation on clusters

- Sequential interactive run (but very time-limited), for example to run a datafile to partition a mesh:

triu datafile

- Parallel interactive run (but very time-limited), for example to check your datafile:

triu 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):

triu -create_sub_file datafile nb_processes

- Then, you submit the job:

mezel2:	qsub	sub_file
callisto:	sbatch	sub_file
CCRT/TGCC:	ccc_msub	sub_file
occigen:	sbatch	sub_file

Parallel calculation on clusters

- 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 Trio_U study
 - The Trio_U parallel command line

Parallel calculation on clusters

- Example on callisto

```
#SBATCH -J name_of_the_job
#SBATCH -p slim
#SBATCH --qos=normal_slim
#SBATCH -t 2880
#SBATCH -n 2
cd /panfs/ixion/home/login/study
exec=absolute_path_to_the_Trio_U_binary
mpirun -np nbprocs $exec datafile nbprocs 1>jdd.out 2>jdd.err
```

- Example on CCRT/TGCC

```
#MSUB -r job_name
#MSUB -n nb_procs
#MSUB -o out.%J
#MSUB -e err.%J
#MSUB -T 86400
# On cluster curie, add also your project (e.g. genden):
#MSUB -A genden
cd /ccc/scratch/cont002/den/login/study
mpirun -np nb_procs ./Trio_U_mpi_opt datafile nb_procs 1>datafile.out 2>datafile.err
```

- Example on occigen

```
#SBATCH -J name_of_the_job
#SBATCH -t 24
#SBATCH -n 2
#SBATCH -N 1
cd directory_of_my_study
exec=absolute_path_to_the_Trio_U_binary
srun --mpi=pmi2 -K1 --resv-ports -n $SLURM_NTASKS $exec datafile $SLURM_NTASKS1>jdd.out 2>jdd.err
```

Parallel calculation on clusters

- Description of partition/queues/classes for each cluster:

callisto: **slim** (no limit of CPU, limited to 800 cores, 6.4GB/core)
 large (no limit of CPU, limited to 320 cores, 12.8GB/core)
 fat (no limit of CPU, limited to 48 cores, 21.3GB/core)

CCRT/TGCC: **test** (30 minutes of CPU, limited to 256 cores)
 prod (24 hours of CPU)
 More class informations with the command **class**

occigen: **all** (24 hours of CPU)

- List of jobs and their state:

callisto: **squeue**
CCRT/TGCC: **ccc_mpp**
occigen: **squeue -u** your_login or **squeue -j** job_number

- Kill a job (the job_number is given by the previous command):

callisto: **scancel** job_number
CCRT/TGCC: **ccc_mdel** job_number
occigen: **scancel** job_numbertrain

Parallel calculation on clusters

Another useful Trio_U script (not on occigen for the moment):

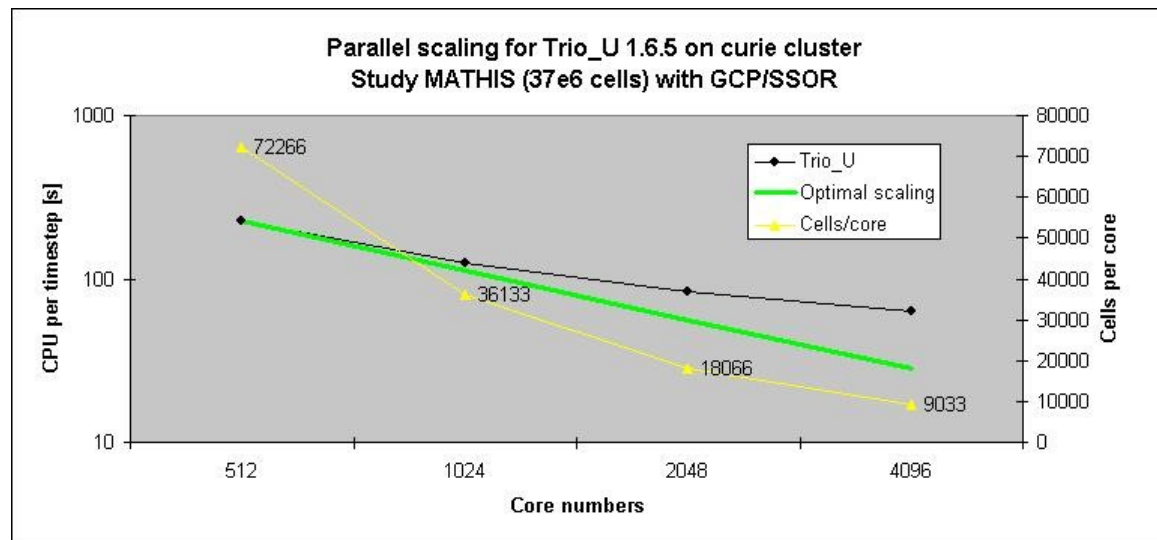
reprise_auto [-test]

- Submits a job in the current directory (a correct submission file should created before)
- Submits periodically new jobs in a way that the Trio_U calculation will restart automatically as soon as possible after the end of the previous job
- This script also backups the main files of the calculation into a dedicated directory
- The `-test` option will test (into a sub-directory named *test_reprise_auto*) this script for a first calculation restarted after 15 minutes.

Parallel calculation on clusters

Some advices:

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



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

Visualization on clusters with VisIt

- If available, use a **VisuPortal** deported session on CCRT/TGCC clusters to run VisIt without network slowness
 - Available on curie cluster: <https://visu-tgcc.ccc.cea.fr/VisuPortal/home>
 - Available on airain cluster: <https://visu-airain.ccc.cea.fr/VisuPortal/home>
 - Ask us the VisuPortal user manual
- Or the Client/server mode
 - See the previous description here with VisIt (**callisto**, **jade**, **eris**, ...)
 - 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

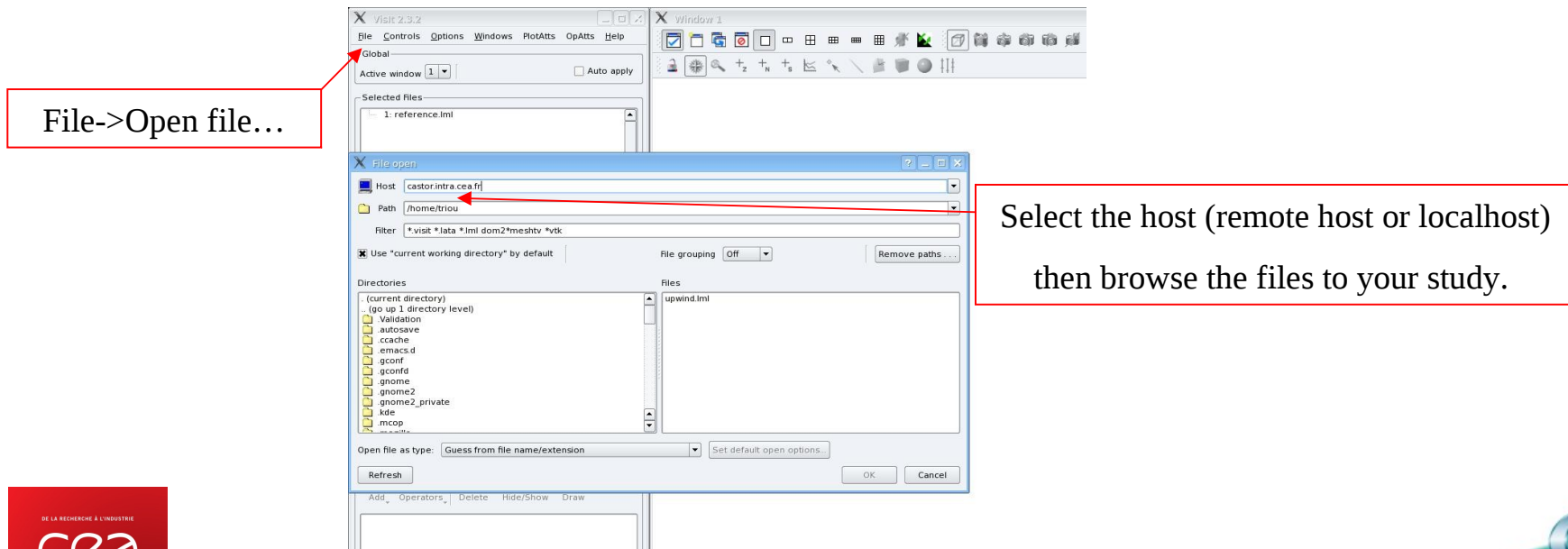
Recent features with VisIt

-The Trio_U install builds a parallel version of VisIt:

```
visit -np 8 -o results.lata
```

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

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



Practice

Exercise 2 : Obstacle.data //

Exercise 3 : Calculation on callisto

Table of contents

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

Recommendations for use of models

Which turbulence models ?

Interested in averaged quantities

- k- ϵ standard model
 - Low cost but lack of generality

Interested in fluctuating quantities

- LES model
 - Involves a refined 3D mesh
 - Recommended in the following cases

Recommendations for use of models

LES model if :

- Strongly non stationary flow
 - Access to mechanical or thermal fluctuations
 - Mixing phenomena
 - Free and impinging jets
- Flow with important secondary structures
 - Tube bends
 - Rectangular channels at reduced Reynolds numbers
- Flow with detachment and/or reattachment
 - Backward facing steps
 - Obstacles
- Flow at slow regimes
 - Natural convection

Recommendations for VEF meshing

- Mesh refinement
 - Minimum of 10 points between two walls
 - To avoid very small time steps, do not create small cells in high velocity regions
- Boundary layers
 - Quality of the mesh near boundary layer will be improved if 2 or 3 layers of regular cells is used near the wall
- Tetra general mesh recommendations
 - Avoid significant changes in the mesh size of two adjacent tetras. A propagation factor of 20% seems to be the upper limit (5% recommended)

Recommendations for VEF meshing

- Tetra general mesh recommendations:
 - Avoid the use of a great amount of obtuse cells (angles between faces greater than 90°). In the contrary, non physical phenomena are observed to the diffusion operator.
 - Use any optimization tools of mesh generators to reduce too large angles
 - Check in the .err file of Trio_U, the angles histogram of the mesh
 - Visualize the field (mesh quality) named LargestAngle_elem
 - Use isotropic cells in all directions as much as possible for a 3D flow. In the case of a flow in a given direction (1D flow), the stretching of a mesh up to a hundred times in the preferred direction nevertheless doesn't much degrade the results
 - Look the Meshing paragraph of the [Best Practice Guide](#) for more advices

Recommendations for conditions

- Boundary conditions

- Wall conditions for RANS model (no rule for the LES model):
 - The first discretization point should be in the logarithmic layer ($30 < y^+ < 500$). y^+ field may be post-processed to check this condition.
 - Look for paragraph “Boundary conditions” in Best Practice Guide

- Initial conditions

- Look for paragraph “Initial conditions” of the Best Practice Guide to optimize the flow initialization in order to improve/reach faster the convergence

Recommendations for convection

- VEF Schemes
 - EF_stab
 - Use it for Navier Stokes and scalar equations (temperature, concentration)
 - Option α is a compromise between:
 - robustness (α near 1, the default value)
 - accuracy (smaller value for α)
 - $\alpha=0.2$ gives the better results. When the mesh is stretched in one direction, this scheme may have some convergence issues.
 - $\alpha=1$ insures the stability and gives the same results for a lot of cases when the forced convection flow has a preferred direction.
 - Muscl
 - Use it for very perturbed or mixing flows
- VDF schemes
 - Quick

Recommendations for implicit time schemes

When to use it ?

- To solve a steady state calculation (e.g. k- ϵ simulation)
- To reduce the time of the transient state of a calculation before switching to an explicit time scheme

WARNING: An unsteady calculation solved with an implicit scheme may give non physical results !

How to use it?

- Define upper and lower limits of the dynamic time step algorithm
- Define a solver (**GMRES**, **BiCGSTAB**,...)
- Define the convergence criterias

Recommendations for implicit time schemes

- Dynamic time step algorithm:

Time step $dt = dt(\text{Courant-Friedrichs-Lewy condition}) * facsec(t)$

With :

$facsec(0) = \mathbf{facsec}$ (lower limit keyword)

$facsec(t^{n+1}) = a * facsec(t^n)$

$facsec(t) \leq \mathbf{facsec_max}$ (upper limit keyword)

The algorithm uses $a=1.2$ and reduces it if necessary

Recommendations for implicit time schemes

- In a first approach, use **facsec=facsec_max** with:
 - **facsec_max=20-30** for:
 - hydraulic calculation only
 - thermal hydraulic with forced convection and no or low coupling between velocity and temperature
 - **facsec_max=90-100** for:
 - thermal hydraulic with forced convection and a strong coupling between velocity and temperature
 - **facsec_max=300** for:
 - thermal hydraulic with natural convection
 - Much higher value is possible for:
 - conduction calculation
- In a second approach (no convergence):
 - Keep **facsec_max** and reduce **facsec** to 5 or 10

Recommendations for implicit time schemes

- It is possible to specify a solver for each equation:
 - Momentum equations
 - **GMRES** solver generally works well
 - Scalar equations
 - **BICGSTAB/ILU** if **GMRES** converges slowly or not at all
 - Transport k- ϵ equations
 - **GMRES** solver and if no convergence (rare), solve these equations with an explicit scheme
- Convergence criteria's
 - Threshold convergence (keyword **seuil**) of the iterative solver (to have at least 3 iterations)
 - A maximal number of iterations may be specified with **nb_it_max** keyword (5 is a good value)
 - For coupled problems, another parameter:
 - Keyword **seuil_convergence_implicite** should be set to 0.001(or less) for strongly coupled problems to guarantee the correct coupling of various equations as e.g. hydraulic and thermal phenomena

Recommendations for explicit time schemes

Time step $\Delta t = \text{facsec} * \Delta t(\text{CFL condition})$

- Euler explicit scheme:

$\text{facsec} \leq 1$

- Runge Kutta schemes (facsec limit may be increased):

$\text{facsec} = 2$ if RK2

$\text{facsec} = 3$ if RK3

$\text{facsec} = 4$ if RK4

Specific recommendations for LES

- Time schemes:
 - High order explicit schemes like:
 - Runge Kutta order 3 (but facsec 1.0 for LES)
 - Adams Bashforth order 2
- Convection schemes:
 - VEF:
 - EF_stab with $\alpha=0.2$ for Navier Stokes equation (do not use first order scheme for a LES, so if you need to set $\alpha=1.0$ to insure convergence, then something is wrong with your mesh)
 - VDF:
 - Centre (order 4) and if unstable use facsec=0.2

Specific recommendations for LES

- Use a periodic box to provide a turbulent velocity field (see page 56):
 - The length of the periodic box should be at least $8 \cdot D_h$ (D_h : hydraulic diameter)
 - Do not stretch the mesh in the axial direction of the box
- Models
 - First approach, use Wale model, then check the turbulent energy spectra
 - If Wale is unable to dissipate sufficiently the high frequencies, use the Smagorinsky model (see Post processing paragraph of the Best Practice Guide)

Recommendations for RANS calculation

- Model adapted to high Reynolds number
- First discretization point of the mesh should be located in the logarithmic zone ($y^+ \sim 30$)
- Taking care to the initial and boundary conditions for k and ε will avoid some issues and dramatically improve convergence
 - In a first approach (for example 10% turbulence rate), you can use (U , bulk velocity and L , a relevant dimension, example a pipe diameter):
$$k \sim 3/2 * (\text{turbulence_rate} * U)^2$$
$$\varepsilon \sim k^{1.5} / L$$
- Use EF_stab scheme for the Navier Stokes equation
- Use upwind scheme for k - ε equation

Recommendations for post processing

- Fields can be visualized at:
 - element, vertex, or face
- Prefer visualization without linear interpolation
- The fields visualized without linear interpolation are :
 - At elements (keyword elem)
 - pressure, turbulent viscosity, y^+ , physical properties and for VDF only, temperature, concentration, k , ϵ
 - At vertexes (keyword som)
 - pressure
 - At faces (keyword faces)
 - velocity and for VEF only, temperature, concentration, k , ϵ
- Prefer the LATA format which can be converted to any other Trio_U format (thanks to lata2dx tool)

What if ?

The calculation does not converge...

- Symptoms:
 - Time-step decreases to the **dt_min** lower limit
 - wall law does not converge (error message)
 - implicit diffusion algorithm does not converge (warning message)
 - ...
- Try:
 - Reduce the time-step (use **facsec** value 0.5 or 0.2) if using an **explicit** scheme
 - Reduce the upper limit of the time-step (reduce **facsec_max**) if using an **implicit** scheme
 - If using the **EF_stab** scheme, try to increase α value incrementally (0.2 to 0.3 or 0.5 or 1.0) to recover stability or reduce the **facsec** (if no change, switch to muscl scheme)
 - If using the **centre** scheme, reduce the **facsec** or change your scheme (**quick** scheme)
 - Contact Trio_U support

Table of contents

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

Meshing tools released with Trio_U

- Trio_U internal mesh tool
 - Used by keywords in the data file
 - Limited to simple geometry (assembling of rectangle in 2D or blocks in 3D)
- Xprepro
 - External tool
 - To create a complex geometry but with a regular hexahedral mesh
- Or use of a mesh generator tool linked with Trio_U:
 - Salomé
 - Gmsh

Meshing with Trio_U

- Mesh generator tools :
 - ICEM (Ansys) generates a file at Trio_U format
 - Trio_U reads .unv files from I-DEAS*
 - Trio_U reads 2D/3D meshes from old tools of Fluent (Gambit/TGrid)
 - Trio_U reads « .med » meshes from Salomé or Gmsh
- Form factors (view factors for the radiation model):
 - Link between Ansys and Trio_U

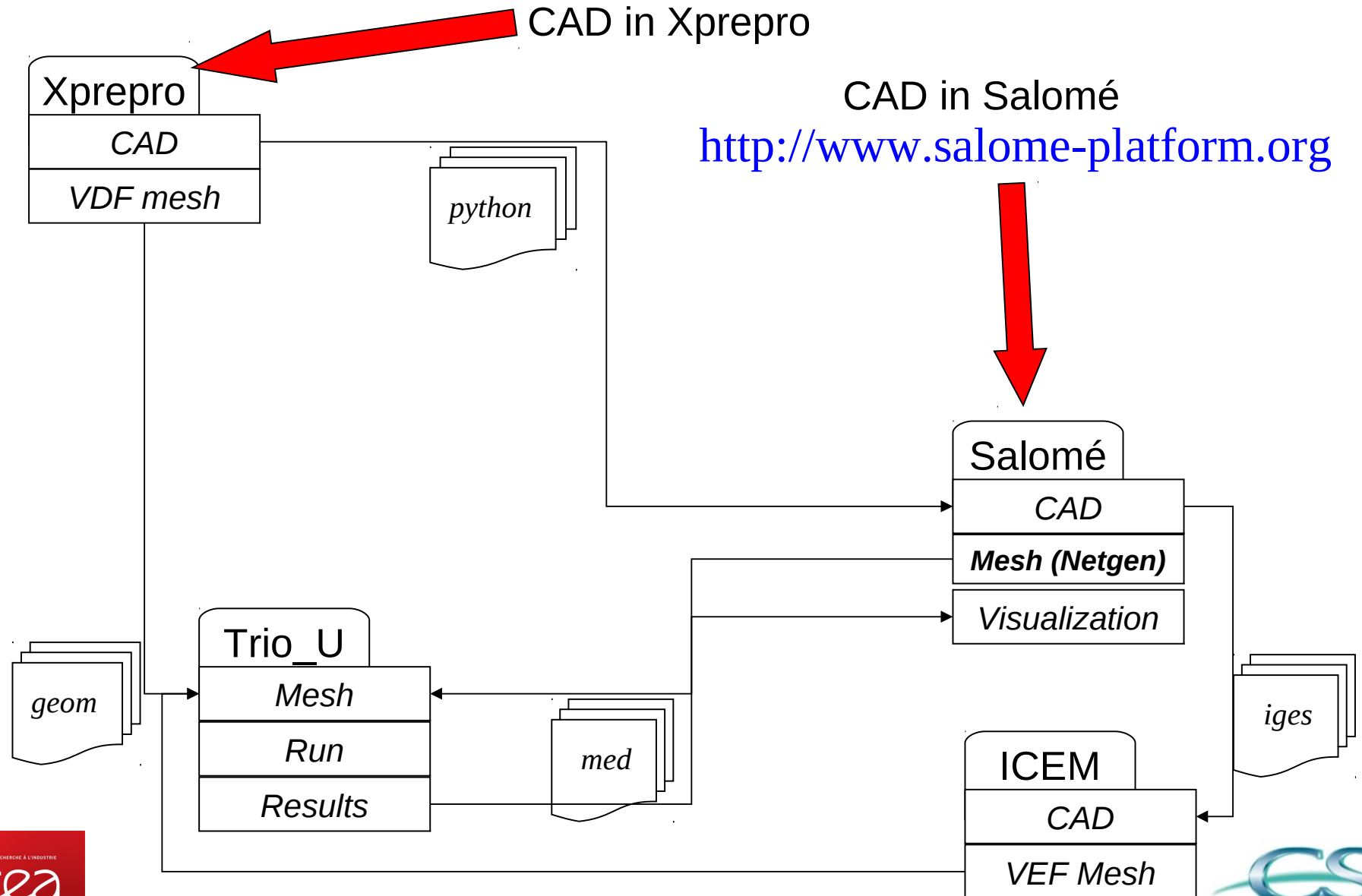
*:tetrahedral meshing only

Meshing tools presented

- Presentation of 3 mesh generators: **Xprepro**, **Salomé**, **Gmsh**
- Exercise with 1 mesh generator according to your needs

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

Integration with Trio_U & Salomé

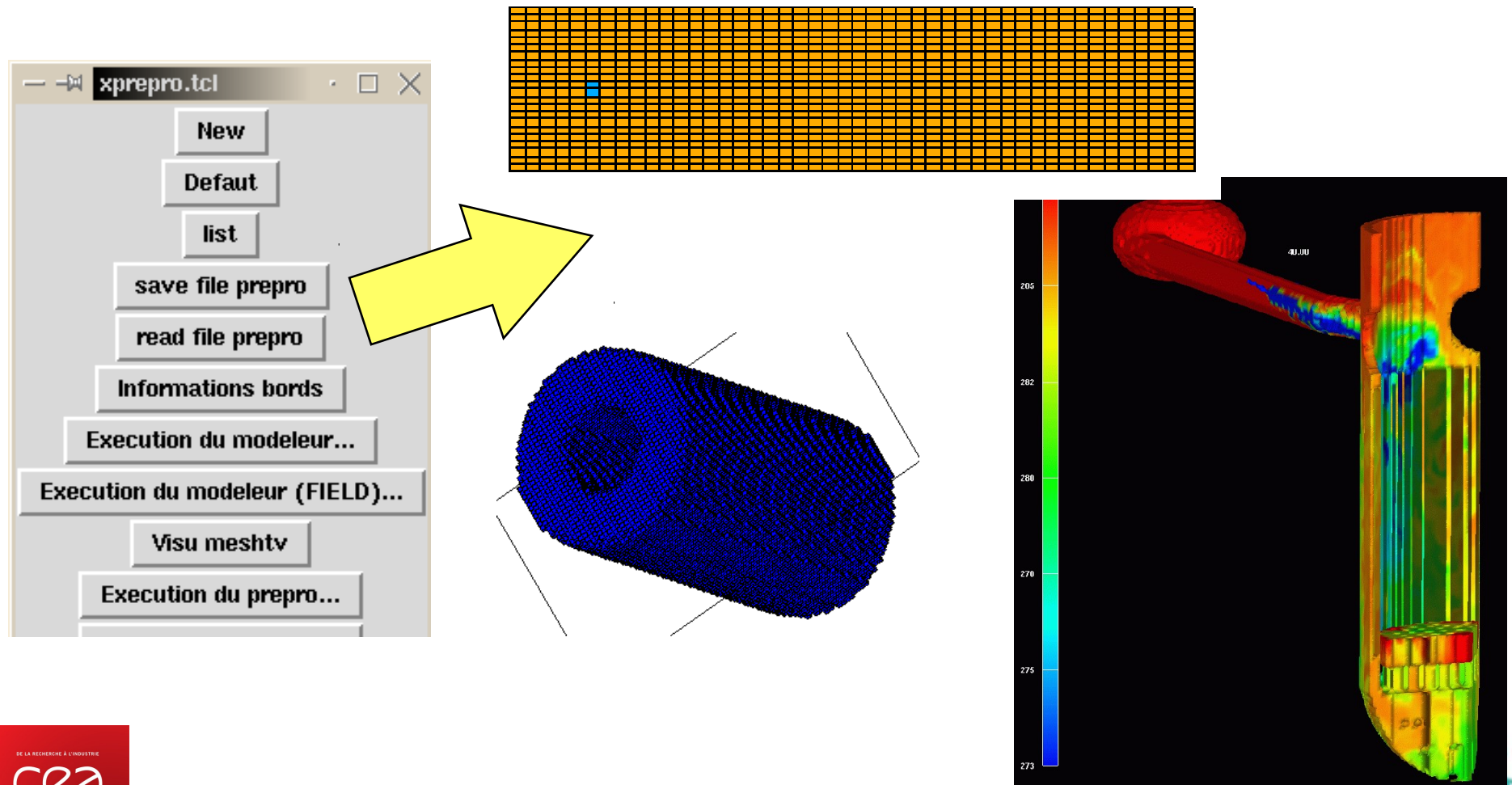


Xprepro -> Salomé

- Geometric model
 - Import the Xprepro CAD (python file) or build the CAD in Salomé
- Boundary definition
 - Create a geometric group
 - Work with the OCC viewer
- Mesh phase
 - Define the surface mesh then the volume mesh (ex: Netgen)
 - Detect/suppress the duplicated nodes
- Mesh files
 - MED export from Salomé
 - Read the mesh in Trio_U data file (**Lire_MED** keyword)

Xprepro

Modeling/mesh tool created by the Trio_U team



Xprepro

Main features:

- Tool to create the geometric model of the domain to mesh
- This model does not depend of the mesh
- Model can be parameterized (variables can be used to define the dimensions of the model)
- A language can be used to modify/duplicate several parts of the model
- Tool to create a 2D or 3D structured mesh of the domain for a VDF calculation

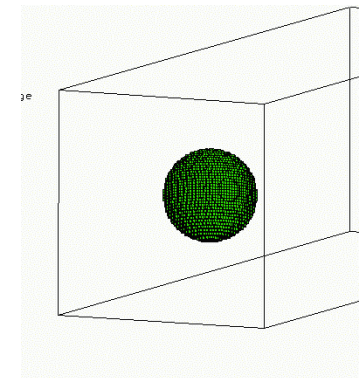
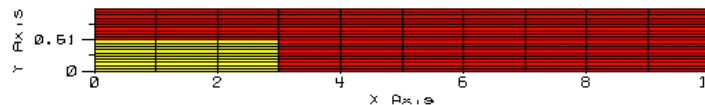
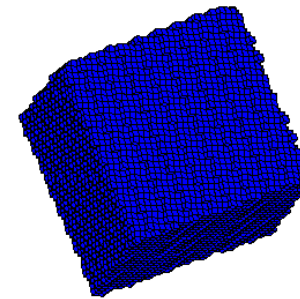
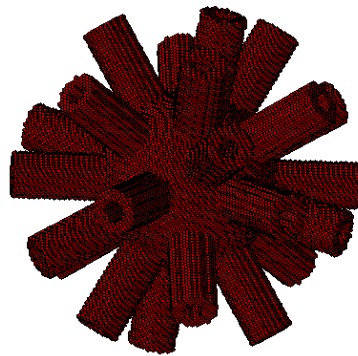
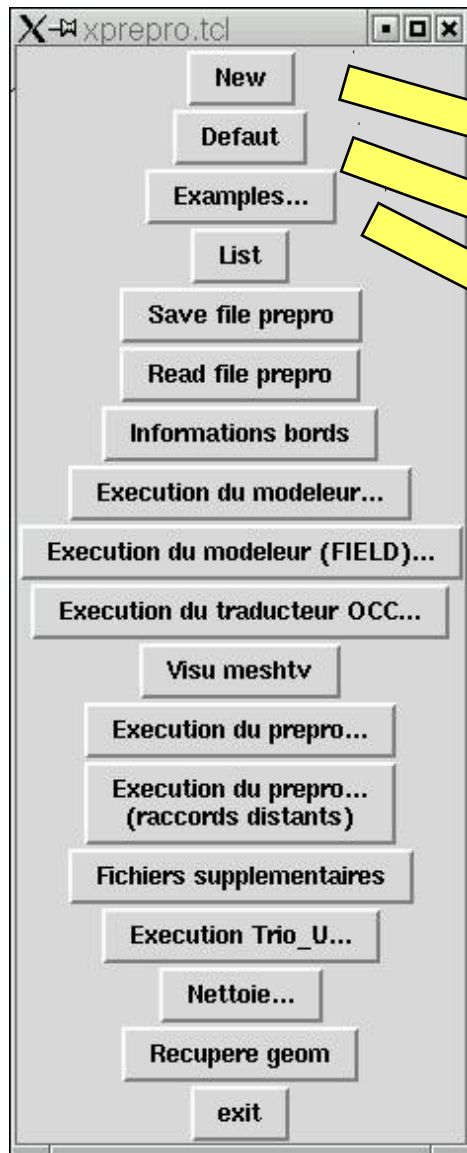
Xprepro

- Other features:
 - Once the model is finished, visualization of a pre-mesh
 - We can create sub-zones (group of several cells)
 - We can partition the domain with Xprepro for // calculation (obsolete feature)
- Xprepro is made of:
 - A GUI written in Tcl/Tk
 - A fortran program
 - Built and compiled during model and mesh phases

Xprepro

- 3 ways to start with the GUI :

- New : Begin from scratch
- Default : Start from a parallelepiped
- Examples :



```
      real xomin,xomax,yomin,yomax,zomin,zomax,eps
C Cotes extremes de la geometrie maillee
      xomin=?
      xomax=?
      yomin=?
      yomax=?
      zomin=?
      zomax=?

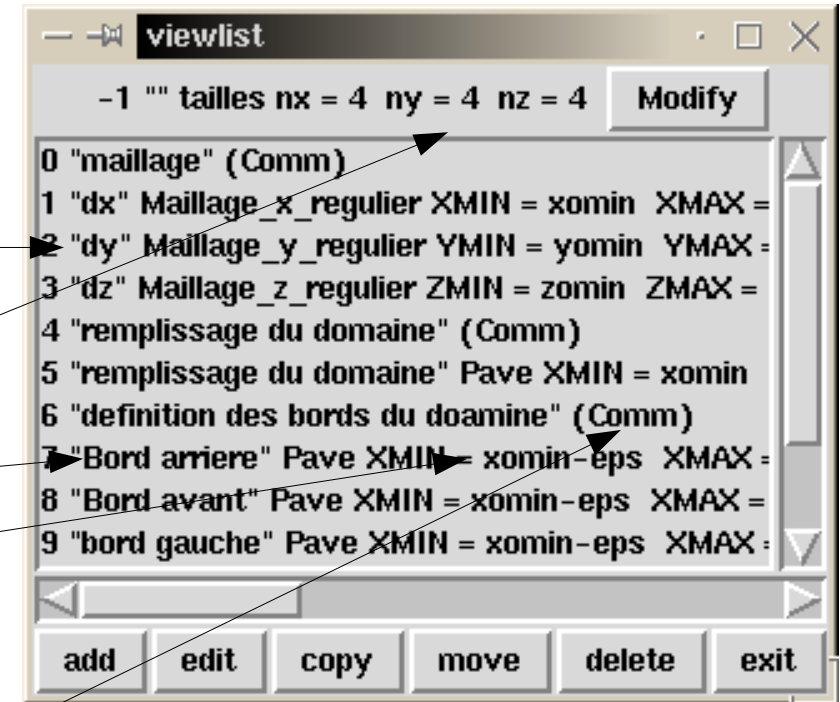
C eps sert a l'epaisseur des paves bords
      eps=0.01

      XM(1)=xomin-eps
      XM(nx)=xomax+eps
      YM(1)=yomin-eps
      YM(ny)=yomax+eps
      ZM(1)=zomin-eps
      ZM(nz)=zomax+eps
```

- Button Default
 - A parallelepiped domain
 - 6 boundaries already defined
- Edit a dimensions file
 - Define the parallelepiped coordinates
 - Define a tolerance eps
 - It is possible to add new variables
 - Take care: it is fortran language !

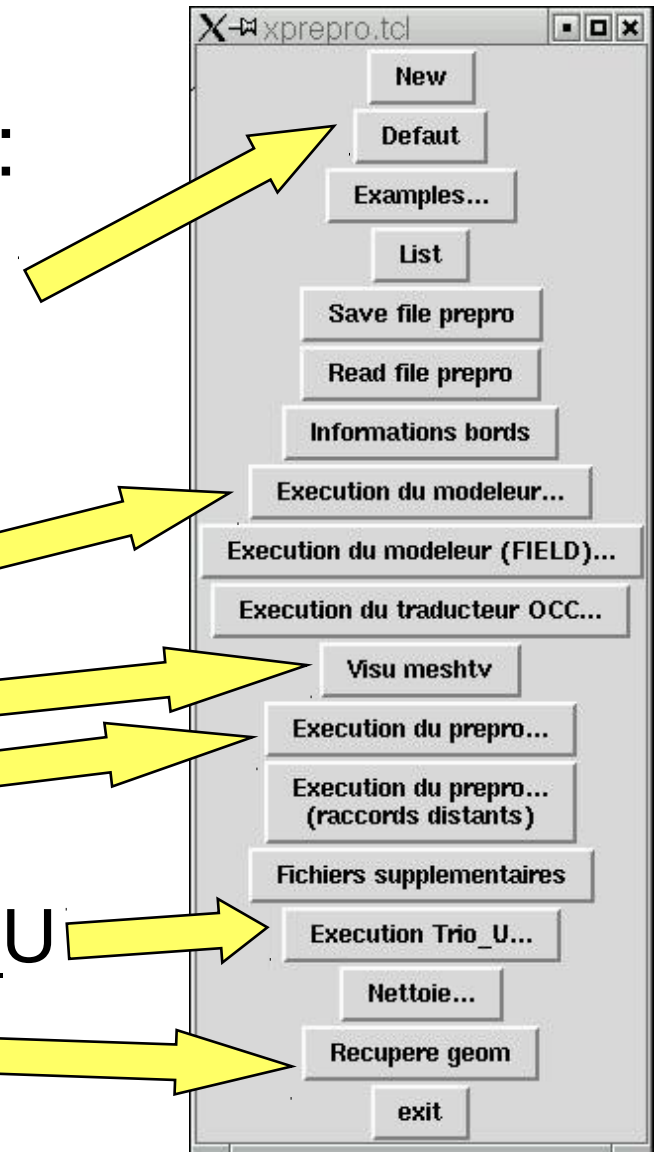
Xprepro

- The GUI Viewlist contains:
 - The pre-mesh
 - The grids according X,Y,Z
 - The number of pts Nx, Ny, Nz
 - Shapes definition
 - The name
 - The coordinates
 - Take care of the definition order (read from top to bottom)
 - Some comments (COMM)

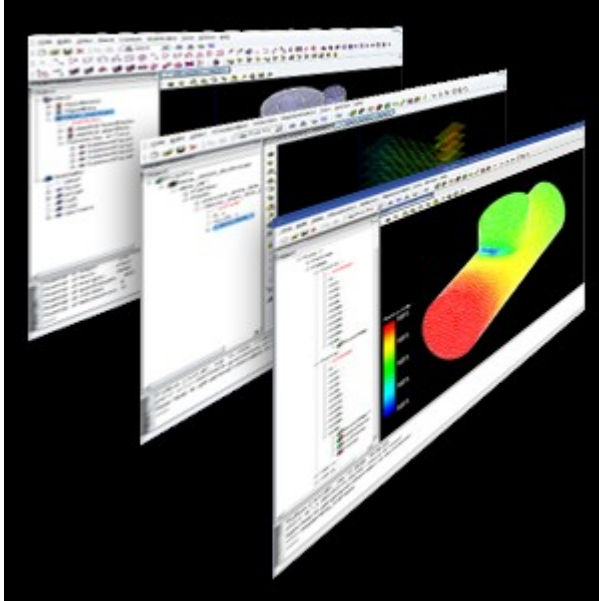


Xprepro

- GUI typical working sequence:
 - Choice of the initial geometry
 - Edit the dimensions file
 - Work into in the view-list
 - Build the model
 - Visualize the pre-mesh
 - Build the final mesh
 - Test the .geom mesh with Trio_U
 - Move files into your study



Salomé



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 Trio_U package

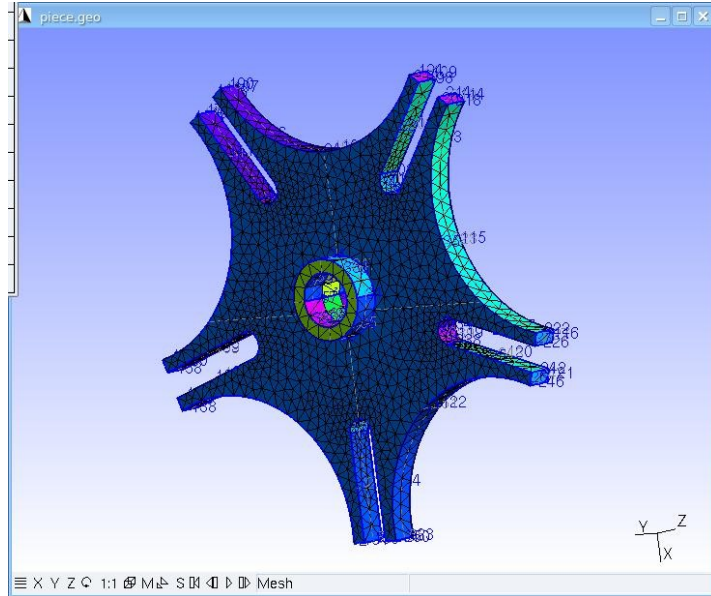
-> To have a full training session, or receive support for install or use, see:

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

support-salome@cea.fr

Short description soon to use the Geometry and Mesh modules....

Gmsh



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 Trio_U install. There are tutorials and examples under:

`$TRIO_U_ROOT/exec/gmsh/share/doc/gmsh/tutorial`

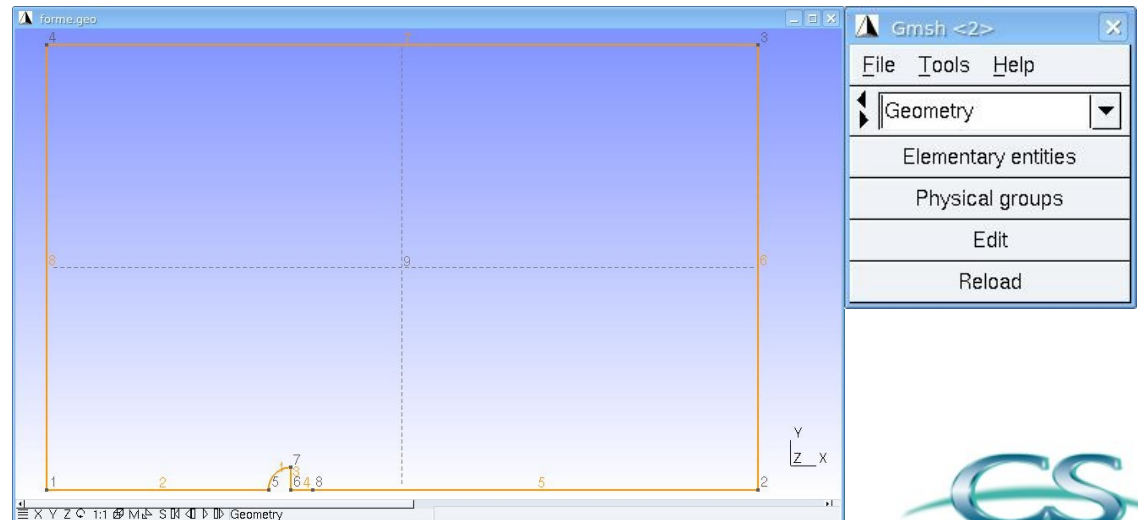
`$TRIO_U_ROOT/exec/gmsh/share/doc/gmsh/demos`

-> 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 (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};
```

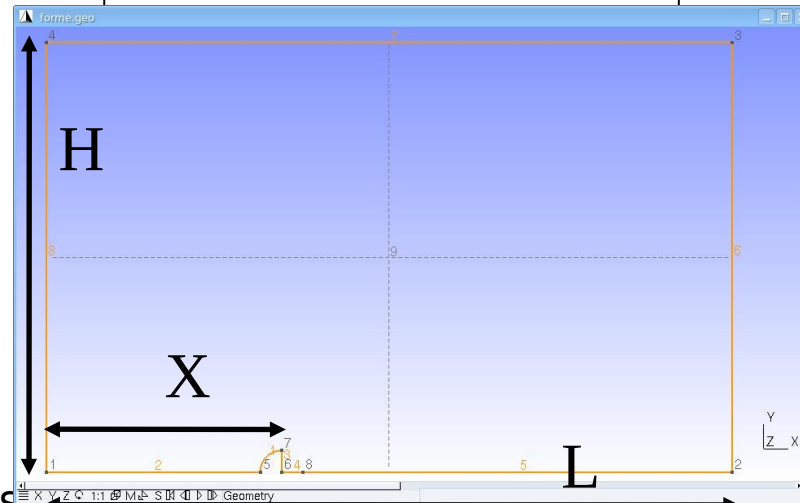
```
// Lines definition
Line(2) = {1,5}; // 2 points
Line(5) = {8,2};
Line(6) = {3,2};
Line(7) = {3,4};
Line(8) = {4,1};

// 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};

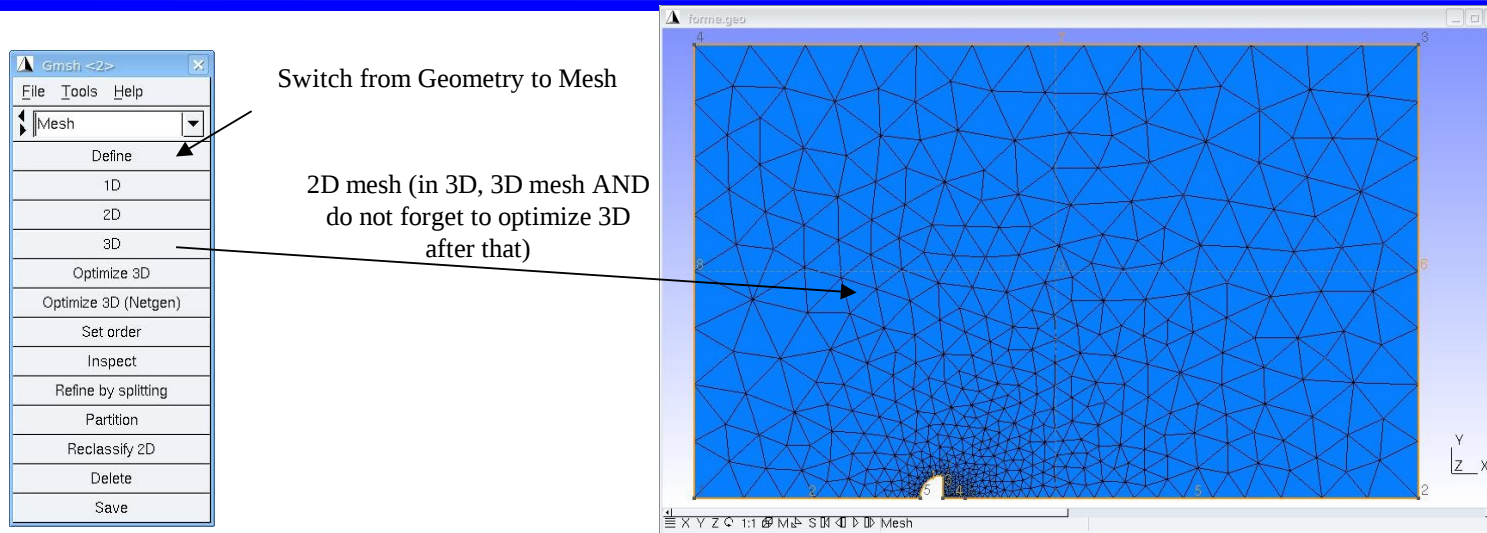
// A circle arc is STRICTLY smaller than Pi
```

```
// 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 line loop 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 line loop
Plane Surface(1) = {1};
// Naming the domain is MANDATORY
Physical Surface("domain") = {1};
```



Gmsh (Mesh and import)



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

To import the mesh, add in the Trio_U datafile:

Dimension 2
Domaine dom
Lire_MED family_names_from_group_names dom mesh_name file.med # By default, the mesh name is the name of the file, so there mesh_name=file #

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 }

Trio_U 1.7.0 user's training session

Practice a mesh tool (2nd day morning)

VEF calculation with Trio_U




- 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 Trio_U

Table of contents

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

Examples of Data files

- If one is interested by Quasi compressible model:
 - The Low mach flow data file should be presented 
 - Exercise 5 (Low mach number flow)
- If one is interested by Front Tracking model:
 - The Front Tracking data file should be presented 
 - Exercise 11 (3D Tank filling exercise)
- If one is interested by Genepi+ data file (old) :
 - See p146 

Data file example (Front Tracking)

```
# Front Tracking calculation #
dimension 3

# Generic problem used for Front Tracking calculation #
Probleme_FT_Disc_gen pb
Domaine DOM
# BEGIN MESH #
Mesh DOM
{
    Pave pave1
    {
        origine 0. 0. 0.
        longueurs 0.04 0.04 0.06
        nombre_de_noeuds 11 11 16

        bord paroi X = 0. 0. <= Y <= 0.04 0. <= Z <= 0.06
        bord paroi X = 0.04 0. <= Y <= 0.04 0. <= Z <= 0.06
        bord paroi Y = 0. 0. <= X <= 0.04 0. <= Z <= 0.06
        bord paroi Y = 0.04 0. <= X <= 0.04 0. <= Z <= 0.06
        bord bas Z = 0. 0. <= X <= 0.04 0. <= Y <= 0.04
        bord haut Z = 0.06 0. <= X <= 0.04 0. <= Y <= 0.04

    }
}
# END MESH #
# BEGIN PARTITION
Partition DOM
{
    Partitionneur tranche { tranches 2 1 1 }
    Larg_joint 2
    Nom_Zones DOM
}
End
END PARTITION #
# BEGIN READ
Scatter DOM.Zones dom
END SCATTER #
```

```
VDF dis
Schema_Euler_explicite sch
Read sch
{
```

```
    tinit 0.
    tmax 0.1
    dt_min 1.e-7
    dt_max 0.5e-2
    dt_impr 10.
    dt_sauv 100
    seuil_statio -1
```

```
}
```

```
# First phase: liquid #
```

```
Fluide_Incompressible liquide
```

```
Read liquide
```

```
{
```

```
    mu Champ_Uniforme 1 0.282e-3
    rho Champ_Uniforme 1 1000.
```

```
}
```

```
# Second phase: gas #
```

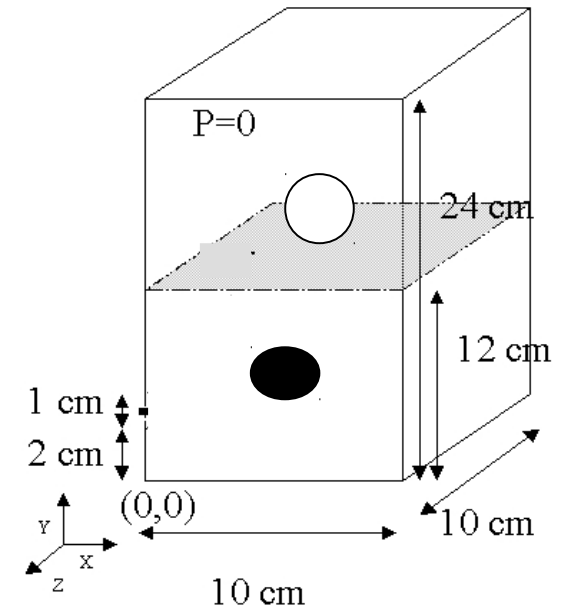
```
Fluide_Incompressible gaz
```

```
Read gaz
```

```
{
```

```
    mu Champ_Uniforme 1 0.282e-3
    rho Champ_Uniforme 1 100.
```

```
}
```



Data file example (Front Tracking)

```
# Definition of the two phase media #
Fluide_Diphasique fluide
Read fluide
{
    # Give a number for each phase #
    fluide0 liquide
    fluide1 gaz
    # Surface tension #
    sigma champ_uniforme 1 0.05
}
# Add a constituent #
Constituant constituant
Read constituant
{
    diffusivite Champ_Uniforme 1 1e-6
}
# Gravity field #
Champ_Uniforme gravite
Read gravite 3 0. 0. -9.81
Associate fluide gravite

# Navier Stokes equation #
Navier_Stokes_FT_Disc hydraulique
# One equation for the two phase flow interface #
Transport_Interfaces_FT_Disc interf
# One equation for a moving body #
Transport_Interfaces_FT_Disc body
# One equation for the constituent #
Convection_Diffusion_Concentration concentration

Associate pb hydraulique
Associate pb interf
Associate pb body
Associate pb concentration
Associate pb DOM
Associate pb sch
```

```
Associate pb fluide
Associate pb constituant
Discretize pb dis
# Define the front tracking problem #
Read pb
{
    hydraulique
    {
        # Turbulence model needed and zeroed for laminar flow #
        modele_turbulence sous_maille_wale {
            Cw 0 turbulence_paro negligible
        }
        # Iterative method to solve the pressure linear system with a non-constant matrix #
        solveur_pression GCP { precondition ssor { omega 1.5 } seuil 1e-12 impr }
        convection { quick }
        diffusion { }
        conditions_initiales { vitesse champ_uniforme 3 0. 0. 0. }
        # Relation between Navier Stokes equation and interface equations #
        equation_interfaces_proprietes_fluide interf # The velocity field moves the gas-liquid interface #
        equation_interfaces_vitesse_imposee body # The body has an imposed velocity field, so moves the fluid #
        boundary_conditions
        {
            haut Sortie_libre_rho_variable champ_front_uniforme 1 0. # Outlet boundary condition for FT model #
            paroi paroi_fixe
            bas Frontiere_ouverte_vitesse_imposee champ_front_uniforme 3 0.0 0.0 0.001
        }
    }
}
```

Data file example (Front Tracking)

```

interf
{
  # Definition of the transport method of the interface: velocity from the
  # Navier Stokes equation #
  methode_transport vitesse_interpolee hydraulique
  # Initial position of the water-gas interface and a drop of water #
  conditions_initiales {
    fonction z-0.03-((x-0.02)^2+(y-0.02)^2)*10 ,
    fonction ajout_phase0 (x-0.02)^2+(y-0.02)^2+(z-0.045)^2-(0.01)^2
  }
  # Options for the meshing algorithm #
  iterations_correction_volume 1
  n_iterations_distance 2
  remaillage {
    pas 0.000001 nb_iter_remaillage 1
    critere_arete 0.35 critere_remaillage 0.2
    pas_lissage 0.000001 lissage_courbure_iterations 3
    lissage_courbure_coeff -0.1 nb_iter_barycentrage 3
    relax_barycentrage 1 facteur_longueur_ideale 0.85
    nb_iter_correction_volume 3
    seuil_dvolume_residuel 1e-12
  }
  # Algorithm for the collision algorithm between interfaces #
  collisions
  {
    active juric_pour_tout
    type_remaillage Juric { source_isevaleur indicatrice }
  }
  # Boundary condition, variable contact angle is possible #
  boundary_conditions
  {
    paroi Paroi_FT_disc symetrie
    haut Paroi_FT_disc symetrie
    bas Paroi_FT_disc symetrie
  }
}

```

```

body
{
  # Initial position of the moving body #
  conditions_initiales { fonction -(((x-0.02))^2+((y-0.02)/0.6)^2+((z-
  0.02)/0.6)^2-(0.015^2)) }
  remaillage { pas 1e8 }
  boundary_conditions
  {
    haut Paroi_FT_disc symetrie
    paroi Paroi_FT_disc symetrie
    bas Paroi_FT_disc symetrie
  }
  # 2 methods to move the body: velocity(x,y,z)=f(x,y,z) or
  x(t),y(t),z(t)=f(t),g(t),h(t) #
  methode_transport vitesse_imposee
    -(y-0.02)*10 (x-0.02)*10 0.
  }
  # Constituent equation #
  concentration
  {
    diffusion { negligeable }
    convection { quick }
    conditions_initiales { concentration champ_fonc_xyz DOM 1 EXP(-((x-
    0.02)^2+(y-0.02)^2+(z-0.03)^2)/0.03^2) }
    boundary_conditions {
      haut frontiere_ouverte C_ext Champ_Front_Uniforme 1 0.
      paroi paroi # Concentration flux = 0 #
      bas paroi
    }
  }
}

```



Data file example (Front Tracking)

```

Postraitement
{
    Sondes {
        vitesse vitesse periode 1.e-7 point 1 0.02 0.02 0.03
        pression pression periode 1.e-7 point 1 0.02 0.02 0.03
        indicatrice_interf indicatrice_interf periode 1.e-7 point 1 0.02 0.02 0.03
    }
    Champs dt_post 0.05
    {
        indicatrice_interf
        concentration
    }
}
liste_postraitements
{
    # Another keywords to post process FT results #
    Postraitement_ft_lata liquid_gas
    {
        dt_post 0.05 nom_fichier liquid_gas
        format binaire print
        champs sommets { vitesse }
        champs elements
        {
            distance_interface_elem_interf
            distance_interface_elem_body
            indicatrice_interf
            pression
            concentration
            vitesse
        }
        # Post process the moving grid of the interface #
        interfaces interf {
            champs sommets { courbure vitesse }
        }
    }
}

```

```

}
Postraitement_ft_lata body
{
    dt_post 0.05
    nom_fichier body
    format binaire
    print
    interfaces body {
        champs sommets { courbure }
    }
}
}
}
Solve pb
End

```

Practice

Exercise 11 on Front tracking :

Tank Filling 3D

Data file example (Low mach flow in Trio_U coupled with Cathare)

```

dimension 2
Domaine dom_fluid
Mesh dom_fluid {
  Pave Entree1      {
    Origine 0. 0.
    Nombre_de_Noeuds 4 3
    Longueurs 0.6 0.4
  }
  {
    Bord inlet1 Y = 0.    0. <= X <= 0.6
    Bord wall  X = 0.    0. <= Y <= 0.4
    Bord wall  X = 0.6   0. <= Y <= 0.4
  },
  Pave Sortie1      {
    Origine 0. 5.4
    Nombre_de_Noeuds 4 3
    Longueurs 0.6 0.4
  }
  {
    Bord outlet1 Y = 5.8   0. <= X <= 0.6
    Bord wall  X = 0.    5.4 <= Y <= 5.6
    Bord wall  X = 0.6   5.4 <= Y <= 5.8
    Bord P_imp X = 0.    5.6 <= Y <= 5.8
  },
  Pave Entree2      {
    Origine 1. 0.
    Nombre_de_Noeuds 4 3
    Longueurs 0.6 0.4
  }
}

```

```

{
  Bord inlet2 Y = 0.    1. <= X <= 1.6
  Bord wall  X = 1.    0. <= Y <= 0.4
  Bord wall  X = 1.6   0. <= Y <= 0.4
},
Pave Sortie2      {
  Origine 1. 5.4
  Nombre_de_Noeuds 4 3
  Longueurs 0.6 0.4
}
{
  Bord outlet2 Y = 5.8   1. <= X <= 1.6
  Bord wall  X = 1.    5.4 <= Y <= 5.8
  Bord wall  X = 1.6   5.4 <= Y <= 5.6
  Bord P_imp X = 1.6   5.6 <= Y <= 5.8
},
Pave Enceinte      {
  Origine 0. 0.4
  Nombre_de_Noeuds 9 26
  Longueurs 1.6 5.
}
{
  Bord wall  Y = 0.4      0.6 <= X <= 1.
  Bord wall  Y = 5.4      0.6 <= X <= 1.
  Bord wall  X = 0.    0.4 <= Y <= 5.4
  Bord wall  X = 1.6   0.4 <= Y <= 5.4
}
}
Trianguler_fin dom_fluid # Triangulate the mesh #

```

Data file example (Low mach flow in Trio_U coupled with Cathare)

```
Schema_Euler_implicite sch
Read sch
{
    tinit 0.
    tmax 10.
    dt_min 1.e-6
    dt_max 0.01
    dt_impr 0.01
    dt_sauv 1000.0
    seuil_statio 1.e-6
    # Options related to implicit scheme #
    facsec 20
    facsec_max 500
    Solveur Piso
    {
        # Convergence threshold for the iterative method #
        #
        # (by default GMRES) to solve the unsymmetric #
        # linear system #
        seuil_convergence_solveur 1.e-6
    }
}

# Definition of the a thermohydraulic problem #
# using the Quasi-compressible model #
Pb_thermohydraulique_QC pb1

# VEF discretization selected #
VEFPreP1B dis

# Gravity defined as a uniform field #
Champ_Uniforme gravite
Read gravite 2 0 -9.81
```

```
# Define the media, helium gas #
Fluide_Quasi_Compressible helium
Read helium
{
    # Pressure in Pa #
    pression 7000000.

    # Sutherland law for viscosity and conductivity #
    Sutherland mu0 1.6E-5 T0 273.15 Slambda 235. C 235.

    # Ideal gas law #
    loi_etat_gaz_parfait {
        Prandtl 1.
        # Specific heat at constant pressure #
        Cp 5193.
        # Cv, specific heat at constant volume given by gamma=Cp/Cv #
        gamma 1.666
    }

    # Keyword for open flow (Neuman condition ), total pressure is constant #
    traitement_ptp constant
}

# Create link between objects #
Associate helium gravite
Associate pb1 dom_fluid
Associate pb1 sch
Associate pb1 helium
Discretize pb1 dis
```

Data file example (Low mach flow in Trio_U coupled with Cathare)

```
# Definition of four surface domains #
# each extracted from a boundary of the problem #
domaine entree1
extraire_surface
{
    domaine entree1
    probleme pb1
    avec_certains_bords 1 inlet1
}

domaine entree2
extraire_surface
{
    domaine entree2
    probleme pb1
    avec_certains_bords 1 inlet2
}

domaine sortie1
extraire_surface
{
    domaine sortie1
    probleme pb1
    avec_certains_bords 1 outlet1
}

domaine sortie2
extraire_surface
{
    domaine sortie2
    probleme pb1
    avec_certains_bords 1 outlet2
}
```

```
Read pb1
{
    Navier_Stokes_QC
    {
        solveur_pression petsc cholesky { } # Direct solver, pressure matrix is constant #
        convection { muscl }
        diffusion {}
        conditions_initiales { vitesse Champ_Uniforme 2 47. 0. }
        boundary_conditions {
            # Dirichlet boundary condition mass flow rate #
            # Ch_front_input : ICOCO coupling field #
            inlet1 frontiere_ouverte_rho_u_impose ch_front_input {
                nb_comp 2 nom rho_u_entree1 probleme pb1
            }
            inlet2 frontiere_ouverte_rho_u_impose ch_front_input {
                nb_comp 2 nom rho_u_entree2 probleme pb1
            }
            outlet1 frontiere_ouverte_rho_u_impose ch_front_input {
                nb_comp 2 nom rho_u_sortie1 probleme pb1
            }
            outlet2 frontiere_ouverte_rho_u_impose ch_front_input {
                nb_comp 2 nom rho_u_sortie2 probleme pb1
            }
            P_imp Frontiere_ouverte_pression_imposee
                    Champ_front_Uniforme 1 70.e5
            wall paroi_fixe
        }
    }
    # Definition of a source term (pressure loss) #
    sources {
        Perte_Charge_isotrope {
            diam_hydr_champ_fonc_txyz dom_fluid 1 1.4-1.3908*(y)1.4)*(y[4.4)
            lambda 4*((16/Re)$(0.079/(Re^0.25))$0.003)
        }
    }
}
```

Data file example (Low mach flow in Trio_U coupled with Cathare)

Convection_Diffusion_chaleur_QC

```
{
    diffusion {}
    convection { muscl }
    conditions_initiales { Temperature Champ_Uniforme 1 673. }
    boundary_conditions {

        # Boundary conditions for temperature #
        # ch_front_input coupling keyword #
        inlet1 frontiere_ouverte t_ext ch_front_input {
            nb_comp 1 nom temperature_entree1 probleme pb1
        }
        inlet2 frontiere_ouverte t_ext ch_front_input {
            nb_comp 1 nom temperature_entree2 probleme pb1
        }
        outlet1 frontiere_ouverte t_ext ch_front_input {
            nb_comp 1 nom temperature_sortie1 probleme pb1
        }
        outlet2 frontiere_ouverte t_ext ch_front_input {
            nb_comp 1 nom temperature_sortie2 probleme pb
        }
        # Outlet boundary condition for temperature #
        P_imp Frontiere_ouverte T_ext champ_front_uniforme 1 700.
        # Zero heat flux #
        wall paroi_adiabatique
    }
    # Heat source term #
    sources
    {
        Puissance_thermique champ_fonc_xyz dom_fluid 1
        2400e6*2/3./10./(3*3.1416*0.7*0.7)*0.6/1.8*1.53*(y)1.4)*(y<4.4)
    }
}
```

Postraitement

```
{
    # As usual, just notice the masse_volumique keyword
    # for the volume mass field #
    Sondes
    {
        sonde_pression1 pression periode 0.05 segment 31
        0.1 0 0.1 5.8
        sonde_pression2 pression periode 0.05 segment 31
        1.5 0 0.1 5.8
        sonde_vitesse1 vitesse periode 0.05 segment 31 0.1
        0 0.1 5.8
        sonde_vitesse2 vitesse periode 0.05 segment 31 1.5 0
        0.1 5.8
        sonde_temperature1 temperature periode 0.05 segment
        31 0.1 0 0.1 5.8
        sonde_temperature2 temperature periode 0.05 segment
        31 1.5 0 0.1 5.8
        sonde_rho1 masse_volumique periode 0.05 segment 31
        0.1 0 0.1 5.8
        sonde_rho2 masse_volumique periode 0.05 segment 31
        1.5 0 0.1 5.8
    }

    format lata
    Champs dt_post 0.05
    {
        pression elem
        pression som
        vitesse elem
        temperature elem
        temperature faces
        masse_volumique elem
    }
}
```


Data file example (Low mach flow in Trio_U coupled with Cathare)

```
Poitraitements      # List of postprocessing blocks #
{
entree1 {
    # Restrict postprocessing on surface meshes previousluy defined #
    domaine entree1
    # Keyword to create new postprocessing fields #
    Definition_champs
    {
        # Define a pressure surface field by Interpolating pressure #
        # volume field on the elements of the surface mesh #
        pressure_entree1 champ_post_interpolation
        {
            localisation elem
            domaine entree1
            source champ_post_refchamp {
                pb_champ pb1 pression
            }
        }
        # Define a temperature surface field #
        temperature_entree1 champ_post_interpolation
        {
            localisation elem
            domaine entree1
            source champ_post_refchamp {
                pb_champ pb1 temperature
            }
        }
    }
}
# Same tasks for each surface meshes #
entree2 { ... }
sortie1 { ... }
sortie2 { ... }
}
Solve pb1      # Solve the problem #
End
```

Data file example (Genepi+)

```
dimension 3
ecriturelecturespecial 1 # Keyword to not save .xyz file #

# Create domains #
domaine dom
domaine MAILLAGE_GENEPI
# Read a MED file (may be composed of several meshes, a
given mesh name is necessary) #
lire_med dom MAILLAGE_GENEPI tface.med
lire_med MAILLAGE_GENEPI MAILLAGE_GENEPI maigro.med

# The keyword is used to rename boundary #
regroupebord dom Wall { WALL_01 }
regroupebord dom Entree_branche_chaude { INLET_03 }
regroupebord dom Entree_branche_froide { INLET_02 }
regroupebord dom Sortie { OUTLET_04 }

ef dis # Define the specific discretization for Genepi+ #

schema_euler_implicite sch2 # Define an implicit scheme #
Lire sch2 {
  tinit 0.0 tmax 55.0
  dt_min 0.02439 dt_max 0.02439
  dt_start dt_fixe 0.02439 # To insure the first time step is imposed #
  facsec 50000
  nb_pas_dt_max 40000
  dt_sauv 1.0 dt_impr 0.5
  seuil_statio 1e-05
  solveur implicite {
    solveur gmres # Iterative solver for the implicit linear system #
    {
      impr seuil 1e-08 # Threshold convergence for the solver #
      diag # Diagonal pre-conditioning #
    }
  }
}
```

```
# Define the gravity field #
champ_uniforme gravite
Lire gravite 3 0.0 0.0 -9.80665

# Specific Genepi+ keywords: #
fluide_melange freon
Lire freon {
  type_fluide fr_114_9_12b
  init_file dump.all
}
pb_melange pb

# Link objects #
Associer freon gravite
Associer pb dom
Associer pb sch2
Associer pb freon
discretiser pb dis

# Keyword to write the domain in a LATA file to visualization purpose #
postraiter_domaine { format lata fichier geom domaine dom }

# Keyword to define a volume porosity field and a surface porosity field #
# Here the fields are read from a MED file #
porosites_champ pb champ_fonc_med last_time betan_betae.med MAILLAGE_GENEPI
POROSITE_2 elem 0.0
```

Data file example (Genepi+)

```
Lire pb {
  # Genepi+ keywords #
  navier_stokes_melange {
    # Usual keywords for Navier Stokes #
    solveur_pression gcp
    {
      precondition { omega 1.5 } seuil 1e-08
    }
    convection { generic amount }
    diffusion { option { grad_u_transpose_partout 1 } }
    conditions_initiales { vitesse champ_fonc_xyz dom 3 0 0 0 }
    conditions_limite
    {
      Wall symetrie
      Sortie frontiere_ouverte_pression_imposee champ_front_uniforme 1 880000.0
      # Boundary condition can also be read in MED files, here velocity profiles: #
      Entree_branche_chaude frontiere_ouverte_vitesse_imposee champ_front_med_test champ_fonc_med last_time maigro.med MAILLAGE_GENEPI VITESSE_3 som 0.0
      Entree_branche_froide frontiere_ouverte_vitesse_imposee champ_front_med_test champ_fonc_med last_time maigro.med MAILLAGE_GENEPI VITESSE_3 som 0.0
    }
    modelisation {
      diffusion_turbulente 1
      prandtl 0.5
      faisceau
      {
        champ_rotation champ_fonc_med last_time rotation_faisceau.med MAILLAGE_GENEPI champ_vectoriel_1 elem 0.0
        champ_presence champ_fonc_med last_time rotation_faisceau.med MAILLAGE_GENEPI champ_scalaire_2 elem 0.0
        transpose_rotation
      }
      plaques_in_file plaques.data # Definition in an external data file #
      init_file dump.all
    }
    pression_initiale champ_uniforme 1 880000
    # List of source terms : separated by comma #
    sources { source_derive , source_frottement , source_pesanteur }
  }
}
```

Data file example (Genepi+)

```
enthalpie_melange # Energy equation definition #
{
  convection { generic amount }
  diffusion { option { grad_u_transpose_partout 1 } }
  conditions_initiales { enthalpie champ_uniforme 1 140000.0 }
  conditions_limite {
    Wall symetrie
    Sortie frontiere_ouverte T_ext champ_front_fonc_txyz 1 1.193e5
    # An example of space dependant boundary condition, hear an Heaviside function for enthalpy: #
    Entree_branche_chaude frontiere_ouverte_enthalpie_imposee champ_front_fonc_txyz 1 1.193e5-400*(x_gt_0.31)*(x_lt_0.32)
    Entree_branche_froide frontiere_ouverte_enthalpie_imposee champ_front_fonc_txyz 1 1.185e5+400*(x_gt_0.31)*(x_lt_0.32)
  }
  # Source terms and an example of a source term read in a file #
  sources { source_derive , source_scalaire champ_fonc_med last_time sourceHgros.med MAILLAGE_GENEPI scalar_1 elem 0.0 }
  # It is possible to change the default parameters of the implicit scheme in case of slow convergence of one equation #
  parametre_equation parametre_implicit {
    solveur gmres { impr seuil 0.0001 diag controle_residu 1 } # controle_residu is a parameter to check the residual do not increase suddenly #
  }
}

postraitement
{
  # Probes to monitor some fields, here on the cell 0 of the master process #
  sondes {
    sonde_hsat enthalpie_saturation_liquide periode 1e-06 numero_elem_sur_maitre 0
    sonde_L chaleur_latente_melange periode 1e-06 numero_elem_sur_maitre 0
  }
  format_lata_champs binaire dt_post 1e-0 # binaire is useless cause it is now the default for LATA output format #
  {
    pression elem
    vitesse som
    enthalpie som
    taux_de_vide_melange elem
  }
}

Resoudre pb
```

Data file example (Genepi+)

```
# File plaques.data #
{
  {
    description OBSTACLE1
    champ_aire champ_fonc_med last_time frt-singulier.med MAILLAGE_GENEPI AIRE_OBSTACLE_1 elem 0.
    transpose_rotation champ_rotation champ_uniforme 9
    -1. 0. 0.
    0. 1. 0.
    0. 0. -1.
  }
  ,
  {
    description OBSTACLE2
    champ_aire champ_fonc_med last_time frt-singulier.med MAILLAGE_GENEPI AIRE_OBSTACLE_2 elem 0.
    transpose_rotation champ_rotation champ_uniforme 9
    1. -0. 0.
    0. 1. -0.
    0. 0. 1.
  }
  ,
  {
    description OBSTACLE3
    champ_aire champ_fonc_med last_time frt-singulier.med MAILLAGE_GENEPI AIRE_OBSTACLE_3 elem 0.
    transpose_rotation champ_rotation champ_uniforme 9
    0. -1. 0.
    1. 0. -0.
    0. 0. 1.
  }
}
```

Table of contents

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

Trio_U automated validation test case

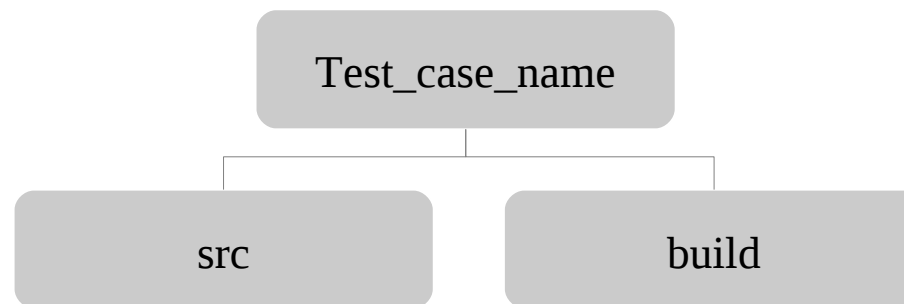
- What is an automated test case ?
- How to generate an automated test case?
- How to run an automated test case?

Trio_U automated validation test case

- **What is an automated test case ?**
 - Tool to compare Trio_U results and experimental data and/or analytical solutions
 - > As final result a report PDF file containing
 - » figures (images or gnuplot plots)
 - » tables with results
 - » visualizations (built by VisIt tool)
 - » etc...
 - Useful to quickly validate a new Trio_U version or to compare different versions of the code

Automated validation test case

- **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
 - build directory will contain the results of the building run



Automated validation test case

- **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,...

Automated validation test case

Figure {

Titre " geometry"

Image cas_test_3.jpg

}

Visu {

titre "Coarse mesh"

Width 6 cm

mesh ./Canal_2D_grossier.lata dom_pb2

pseudocolor ./Canal_2D_grossier.lata dom_pb2 TEMPERATURE ELEM

blackvector ./Canal_2D_grossier.lata dom_pb2 VITESSE ELEM

}

Figure {

Titre "Heat flux as a function of time for various meshings"

Dimension 2

Description "The graph below shows the heat flux"

Description "The calculation stops when the calculation has reached a stationary state."

LabelX "time (in s)"

LabelY "Heat flux at the wall (in W)"

InclureDescCourbes 0

Legende below title "légende"

Courbe {

Legende "-Flux bord coarse mesh"

Origine "Trio_U"

Version "1.5.4"

fichier ./Canal_2D_grossier_pb2_Diffusion_chaleur.out

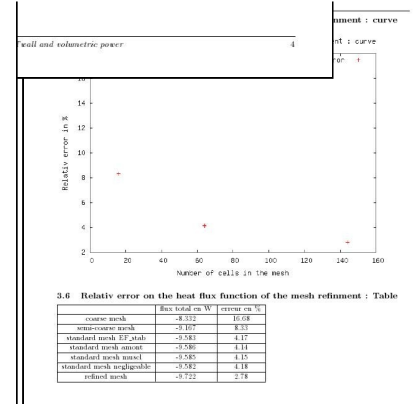
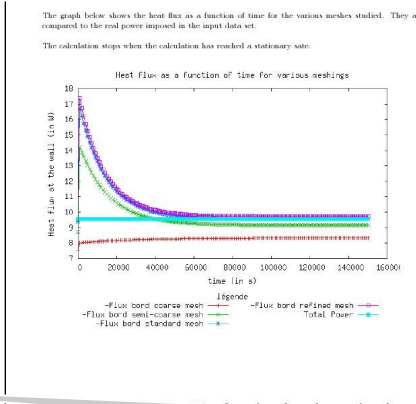
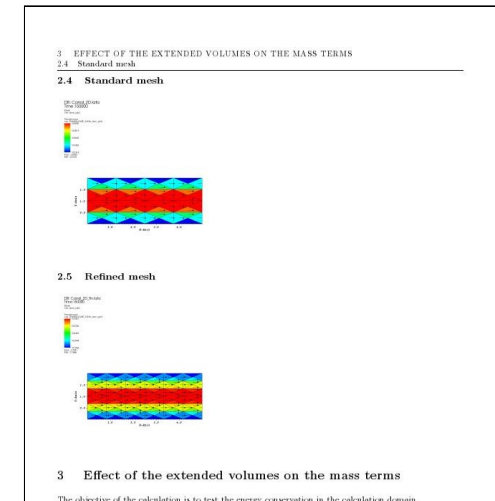
Style linespoints

colonnes (\$1) (-\$2)

}

An image (jpg format for example) will be included to describe the geometry

An image with temperature and vector velocity from VisIt will be included



Example of figures plotted by gnuplot

Automated validation test case

- 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 ..
```

Automated validation test case

- Example of a **pre_run** script:

```
#!/bin/bash  
# Uncompress the mesh file  
gunzip -c ../Channel.msh.gz > Channel.msh  
# Partition the mesh with the make_PAR.data tool:  
make_PAR.data flow.data
```

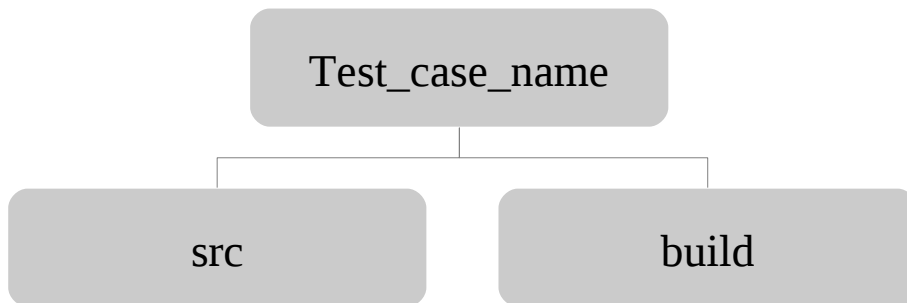
Automated validation test case

- 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
```

Automated validation test case

- **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 PDF report file

Automated validation test case

- User guide:
–\$TRIO_U_ROOT/doc/Trio_U/HowTo_Validation.pdf
- .prm syntax documented in:
–\$TRIO_U_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml
- Examples of automated validation test case:
–\$TRIO_U_ROOT/Validation/Rapports_automatiques/Validant/Fini
- Examples of automated verification test case:
–\$TRIO_U_ROOT/Validation/Rapports_automatiques/Verification/Fini
- Demo:
cd \$TRIO_U_ROOT/Validation/Rapports_automatiques/Validant/pas_fini/Drag
Run_fiche -xpdf
Run_fiche -help # Give all the options #

Table of contents

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

Trio_U Support (1/2)

Subscribe to the Trio_U newsletters (diffusion list):

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

Ftp site: ftp://ftp.cea.fr/pub/Trio_U/a87pour/index.html

- Download new versions
- Release Notes

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

Hot line

- **triou@cea.fr**
- Céline CAPITAINE 01 69 08 42 68 (Saclay bât 451 pièce 43)
- Marthe ROUX 01 69 08 00 02 (Saclay bât 451 pièce 43)

Trio_U Support (2/2)

- A release every 3 months:
 - Linux version only
 - Installed on several CEA clusters and TGCC/CCRT (HPC CEA institute)
 - Installation by users or Trio_U support
- Documentation available here ftp://ftp.cea.fr/incoming/y2k01/Trio_U/doc/ or under \$TRIO_U_ROOT/doc/Trio_U directory:
 - **Trio_U_and_Xprepro_presentation.pdf** (these slides)
 - **Models_Equations_Trio_U.pdf** “Methodology for incompressible single phase flow” (~Trio_U equations & models)
 - **Best_Practice_Trio_U.pdf** “Validation of Trio_U code” (~Trio_U User Guide)
 - **User_Manual_Trio_U.pdf** “User Manual Trio_U”
 - **Manuel_Xprepro.pdf** “User Manual Xprepro”
 - **Developer_Trio_U_presentation.pdf** Trio_U development Presentation
 - **Tutorial_Trio_U.pdf** Trio_U & Xprepro tutorial