TRUST tutorial v1.7.3

CEA Saclay

triou@cea.fr

June 2, 2016

Overview

1. Initialization
2. Obstacle VDF
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. Tank filling (3D-two phases flow)
11. Meshing tools
12. Validation form
13. Index

1. Initialization
2. Obstacle VDF
   - Obstacle: Sequentiel calculation
   - Obstacle: Parallel calculation
   - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. Tank filling (3D-two phases flow)
11. Meshing tools
    - Salomé to create a 3D VEF mesh
      - Cylinder
      - Revolution
      - T-shape
    - Gmsh
      - 2D VEF mesh
      - 3D VEF mesh
12. Validation form
13. Index

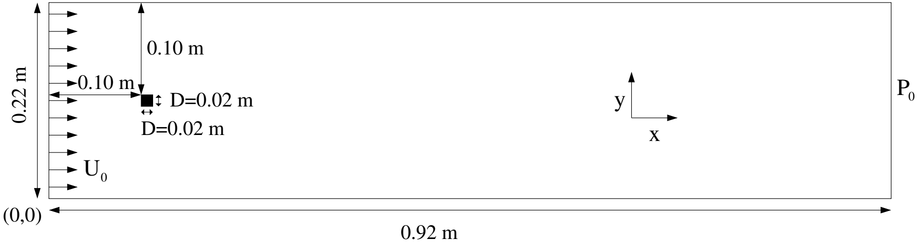
Initialization

- First, initialize the TRUST environment. On CEA Saclay PCs and callisto cluster, TRUST versions are available with (e.g. X.Y.Z=1.7.3):

  source /home/triou/env_TRUST_X.Y.Z.sh

- Second, several editors (vim, emacs, nedit) can be configured to highlight TRUST keywords in the data files. If you prefer using **nedit**, please do the following:
  - Run **nedit**, and select Preferences → Save Defaults.
  - Then run **trust -config nedit**, the message "nedit.rc updated" should appear.

Geometry



- Fluid: $\mu = 3.7\,10^{-5} kg.m^{-1}.s^{-1}$, $\rho = 2kg.m^{-3}$ and $R_e = \frac{U_0 H_{inlet} \rho}{\mu} = \frac{1\times0.22\times2}{3.7\ 10^{-5}} = 11891$
- Boundary conditions:
  - Inlet with fixed velocity: $U_0 = 1m.s^{-1}$
  - Outlet with a fixed pressure: $P_0 = 0$
  - Square cylinder: Wall
  - Upper and Lower walls: Symmetry

Create a study

- First, you must have already done the commands of the first section.
- Open a terminal and run the commands to create a directory for your studies:

```
mkdir -p  ~/test/yourname
cd  ~/test/yourname
```

- Copy a test case from the TRUST database to your study with the command:

```
trust -copy Obstacle
cd Obstacle
```

- Run the test case with the command:

```
trust Obstacle
```

- Edit there the data file Obstacle.data and change some lines in order to modify the time step to 0.004s:

```
nedit Obstacle.data &
```

Probes and parameters

- Add to the post processing block of Obstacle.data the following elements :
  - A segment of pressure probes between the points (0.01, 0.12) and (0.91, 0.12).
  - A segment of velocity probes between the points (0.92, 0.00) and (0.92, 0.22) to see the velocity profile behind the square cylinder.
  - Add the vorticity to the fields being post processed and change the writing period to 0.5s. To find the keyword for this field, you can open the User's manual with:
    trust -doc &
  - Add the keyword "**format lata**" inside the block, just before the keyword **fields** in order to use the post processing tool VisIt during and/or after the calculation.

→ You have access to useful resources in the $TRUST_ROOT/index.html file with your favourite browser (eg: firefox). Take few minutes to find test case examples containing a particular keyword thanks to the Keywords link:
firefox $TRUST_ROOT/index.html &
or **trust -index &**

Visualization during the calculation

- Run the calculation with "-monitor" option to access to a visualization menu: **trust -monitor Obstacle**
- Visualize the evolution of the pressure with the plot number 3. Check also the velocity profile behind the square cylinder. Close the plots (for example, -3 to close the plot 3).
- Check the convergence monitoring: plot 0 in the first menu to open the convergence menu, then open the 4 following plots:
  - Pressure linear system convergence at each time step
  - Residual error $\|Ax - B\|$ of the pressure linear system at the last time step
  - Time step evolution
  - Flow rate error evolution

  Type "5" to quit this menu and return to the first one.
- Visualize the equation residuals (plot 2).

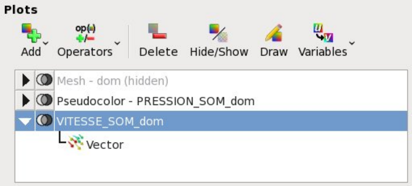
VisIt

- Open the menu "Surface fluxes characteristics" (plot1), visualize:
    - the drag exerted by the fluid on the square Cylinder (plot 1, then 4 for the square), close this menu with 7.
    - the "Pressure drag + Friction drag" with 5 and select the Square boundary (4). Each drag is plotted together with the total drag also called viscous drag.

  To quit this tool, close the plots and type "#".
- Once the calculation is finished, visualize the results with the graphical tool VisIt: **visit &**
    - First, we are going to configure VisIt: In the menu File → Open file, select Off instead of Smart for File grouping option. For the Filter, specify *.lata to list only the LATA results file. Then save your choices, in the menu Options → Save Settings.
    - In the menu File → Open file, select the Obstacle.lata file.
    - Visualize the mesh in the "Plots" area with "Add → Mesh → dom" then click on the button "Draw". Zoom and move the mesh in the right window. You will de-zoom with right button (View → Reset view) or with a combination of CTRL keypad and left button.
    - Visualize the pressure field (Plots area: "Add → Pseudocolor → PRESSION_SOM_dom" + Draw then select the last time on the Time slider)

VisIt

- Suppress or hide the mesh (Select Mesh then clcik on Delete or Hide/Show).
- Visualize the velocity field (Plots area: "Add → Vector → VITESSE_SOM_dom" + Draw). You can change each plot attributes:
  - click once onto the small arrow "▶" then
  - double click on the item Vector (cf the figure below). For example, change the amount of vectors plotted (by default 400, increase this number to 40000 then click the button "Make default" and save definitively this change with the menu Options → Save Settings). You need to click Apply to update. Then click "Dismiss" to close the window.



- Print your visualization (File → Save window): a PNG file is created into your study directory.

VisIt

- Add a second screen with "Windows → Layouts → 1x2",
- Plot a pressure horizontal profile:
  - select the pressure field and thanks to the right button,
  - select "Mode → Line out", and define your profile with left button,
  - click on the origin point, let the left button pushed, and release at the end point.
  - The profile is shown on the second window.
- You notice that it is necessary to update (button Draw) the right window after adding a new plot or changing an option. It is possible to automatically update by activating "Auto apply" on the top right of the VisIt's GUI.
- You can create create new fields (expression) with "Controls → Expressions → New" by using existing variables and complex functions and visualize it.
- You can animate your visualization and/or create a movie (File → Save movie)
- You can operate calculations on variables with complex queries (Controls → Query),
- You can save a complex session (File → Save session) and reopen it during a next analyze with VisIt (File → Restore session),

Outputs and "reprise"

  - ○ During a 3D visualization, you will use one of the available Operators (In Plots, "Operators → Slicing → Slice") to create a 2D slice either in a 3D space, or projected to a 2D space.

- Edit the different output (*.out) files to read the complete balances (mass, stress, energy, ...) on the whole domain or at the boundaries.
- Change the data file to restart the calculation. You will modify **tinit** (pick in the .err file the last backup time of the previous calculation) and **tmax** values, and add in the problem definition just before the last "}":
  reprise binaire Obstacle_pb.sauv
  (The file "Obstacle_pb.sauv" must have been created during the first run.)
- Restart the calculation again with:
  trust -monitor Obstacle
  to see that values are added to the first probes during the new calculation.

⇒ *Remark :* to restart your calculation, you can also use the keyword **resume_last_time** instead of **reprise** and only change the **tmax** value (cf User's manual).

1. Initialization
2. Obstacle VDF
    - Obstacle: Sequentiel calculation
    - Obstacle: Parallel calculation
    - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. Tank filling (3D-two phases flow)
11. Meshing tools
    - Salomé to create a 3D VEF mesh
        - Cylinder
        - Revolution
        - T-shape
    - Gmsh
        - 2D VEF mesh
        - 3D VEF mesh
12. Validation form
13. Index

Parallel calculation: method 1

The goal of this exercise is to introduce parallelism in the data file of the previous exercise.

- Go to the previous study (should be done) and after you had suppressed the **reprise** keyword and set **tinit** to 0 again in the *Obstacle.data* file, create the two new files:

  ```
  cd ~/test/yourname/Obstacle
  mkdir PARA1
  cd PARA1
  cp ../Obstacle.data DEC_Obstacle.data
  cp ../Obstacle.data PAR_Obstacle.data
  cp ../Obstacle.geo .
  ```

- Edit the first file (*DEC_Obstacle.data*) to create the partition of the mesh.

Parallel calculation: method 1

- In this file, uncomment the block around the **Partition** keyword.
  - Here, the partitioning tool **Metis** is used. We cut in **nb_parts** blocks, here in 2.
  - The overlapping width **Larg_joint** between two parts of the partition should be defined according to the numerical scheme higher order, generally the convective scheme. Its value is generally 1 for the two order scheme, and 2 for the three or four-order scheme like Quick scheme.
  - In VEF, you should use **2** for **Larg_joint** except when partitioning a domain where only the conduction equation will be solved.
  - At least, the keyword **zones_name** is useful to define the name of the files containing the partitioned mesh and to write these files.
  - Notice the presence of the keyword **End** in the "Partition" block: the code will stop to read the dta file at this point!
- Run the data file: **trust DEC_Obstacle**
- Check the partitioned mesh files DOM_0000.Zones and DOM_0001.Zones are generated in your directory: **ls *.Zones**

Parallel calculation: method 1

- Now, edit the file PAR_Obstacle.data and comment the read of the mesh (**Read_file**) and uncomment the **Scatter** keyword which will read the partitioned mesh. Visualize it with VisIt:
  trust -mesh PAR_Obstacle
- Now, run a parallel calculation with TRUST:
  trust PAR_Obstacle 2
- The post-processing task is identical in sequential or parallel mode. You have the probes into the .son files and the whole fields in the .lata files. To run VisIt with the command line:
  visit -o PAR_Obstacle.lata &
- Select the last time step and visualize the blocks (with Plots: Add → Subset → blocks) which represent the parts of the domain partition, then the velocity fields. You can also visualize a field only on a selected part (block) with the menu Control → Subset.
- To visualize probes after the end of the calculation, you can run the command line:
  trust -probes PAR_Obstacle

Parallel calculation: method 2

- The existing tool **make_PAR.data** is useful on the data files which have the marks MAILLAGE/MESH, DECOUPAGE/PARTITION and LECTURE/SCATTER. If you run the following commands:
- First we will create a new working directory with our data files:

  ```
  cd ~/test/yourname/Obstacle
  mkdir PARA2
  cd PARA2
  cp ../Obstacle.data exemple.data
  cp ../Obstacle.geo .
  ```

- Then we can run the command:

  ```
  make_PAR.data exemple 3
  ls
  ```

Parallel calculation: method 2

- It creates:
  - a SEQ_exemple.data file which is a copy of the sequential data file exemple.data,
  - a DEC_exemple.data file which is the first data file to be run. It is immediately run by the command line **make_PAR.data** to create a partition (with 3 sub zones here), located in the ***.Zones files.

    ls *.Zones

    Note that the code stops reading this file at the keyword "End" just before the "# END PARTITION #" block.
  - a PAR_exemple.data file which is the data file for the parallel calculation. It uses the ***.Zones files to read the mesh through the line "Scatter DOM.Zones dom". Note that the meshing and cut of the mesh are commented here.
- Then you have to run the calculation by the usual command completed by the number of processors needed:

  trust PAR_exemple 3

Parallel calculation: method 2

Useful informations:

- Be careful when you want to modify you data file! You have two possibilities:
  - you want to modify your mesh,
  - you want to modify the calculation parameters.
- For the first one, you can modify:
  - the file exemple.data and run **make_PAR.data**. But it will erase the DEC_exemple.data, SEQ_exemple.data and PAR_exemple.data files and create new zones. Then it will run the new DEC_exemple.data file which gives your new ***.Zones files or,
  - the meshing part of file DEC_exemple.data and run it with:
    trust DEC_exemple.data

  Then run the parallel calculation normally, on the new ***.Zones files.
  trust PAR_exemple 3

Parallel calculation: method 2

- For the second possibility, you can modify:
  - the file exemple.data and run **make_PAR.data**. But it will erase the DEC_exemple.data, SEQ_exemple.data and PAR_exemple.data files and create new ones. Then it will run the new DEC_exemple.data file. Note that in that case, you don't need to re-create the mesh so you can use the second point below:
  - modify the PAR_exemple.data file without running **make_PAR.data**.

  Then run the PAR_exemple.data file with:
  trust PAR_exemple 3
- Notice that if after a certain time, you want to reopen an old case and understand want you did in it without any doubts, you may create two files by your hands:
  - one "BuildMeshes.data" file only for the mesh and the cut of the mesh, and
  - one "calculation.data" file for the parallel calculation.

  You will run it like:
  trust BuildMeshes
  trust calculation nb_procs

Parallel calculation on a cluster

- Log on to the cluster callisto and initialize the TRUT version:

```
ssh -X login@callisto-login1(.intra.cea.fr)
source /panfs/ixion/home/triou/TRUST/TRUST_X.Y.Z/env_TRUST.sh
or source /home/triou/env_TRUST_X.Y.Z.sh
```

- Copy the study Obstacle:

```
cd /panfs/ixion/home/login
mkdir -p test/yourname
cd test/yourname
trust -copy Obstacle
cd Obstacle
```

- Open Obstacle.data and add **"format lata"** in the post-traitement block.
- Create automatically a 2 partitioned mesh and a parallel datafile with: **make_PAR.data Obstacle**

Parallel calculation on a cluster

- Submit the parallel calculation on 2 cores:

  trust PAR_Obstacle 2

- For clusters, you have to create a submission file:

  trust -create_sub_file PAR_Obstacle 2

- Open the file sub_file and change the name of the job. Note that we will see only the first eight characters of the name of the job in the list of submitted jobs.

- Submit the job with: **sbatch sub_file**

- Check the state of the job with: **"squeue"** or **"squeue -u login"**

- You could run VisIt from the cluster but it is not recommanded. But you can run VisIt on your PC and access/visualize the result file on callisto cluster, with File → Open File and select callisto in the localhost menu.

1 Initialization

2 Obstacle VDF
- Obstacle: Sequentiel calculation
- Obstacle: Parallel calculation
- Obstacle: Parallel calculation on a cluster

3 **Heat exchange VDF/VEF**

4 Low mach number flow

5 Periodic 3D channel flow

6 Constituents and turbulent flow

7 3D turbulent flow in a curved pipe

8 Turbulent flow on a 3D step

9 Tank filling (2D-single phase flow)

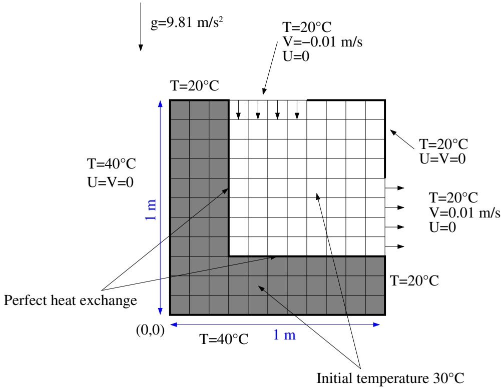
10 Tank filling (3D-two phases flow)

11 Meshing tools
- Salomé to create a 3D VEF mesh
  - Cylinder
  - Revolution
  - T-shape
- Gmsh
  - 2D VEF mesh
  - 3D VEF mesh

12 Validation form

13 Index

Heat exchange VDF/VEF exercise



Fluid:

$$Pr = \frac{\mu C_p}{\lambda} = 1,$$
$$T_{ref} = 30^o C,$$
$$\mu = 2\ 10^{-3} kg.m^{-1}.s^{-1},$$
$$\rho = 2\ kg.m^{-3},$$
$$\lambda = 1.W.m^{-1}.K^{-1}$$
$$Cp = 500 J.kg^{-1}.K^{-1},$$
$$\beta = 1.10^{-4} K^{-1}$$

Solid:

$$\rho = 1000 kg.m^{-3}$$
$$\lambda = 250 W.m^{-1}.K^{-1}$$
$$Cp = 100 J.kg^{-1}.K^{-1}$$

Heat exchange VDF/VEF exercise

- Create a new study Coupling_VDF by copying the docond study:
  cd ∼/test/yourname
  trust -copy docond
  mv docond Coupling_VDF
  cd Coupling_VDF
- Check the fluid and solid characteristics inside the docond.data file.
- This coupled problem is constitued by 2 domains of calculation with a mesh of 10x10 cells ($\Delta x = \Delta y = 0.1m$) created with 3 blocks.
- Modify the data file to have the 2 domains on a mesh of 40x40 cells ($\Delta x = \Delta y = 0.025m$). For this, you will change the number of nodes for each block like this:
  First block (Cavite1): 4 11 → 13 41
  Second block (Cavite2): 8 4 → 29 13
  Third block (Cavite3): 8 8 → 29 29
- Change **"format lml"** to **"format lata"** into the two problems definition and run the calculation and check the evolution.

Heat exchange VDF/VEF exercise

- Then post process the temperature field with the tool VisIt. A natural convection cell appears. Change the colour tables for the temperature to have the same one on the 2 domains.
- We are going to change the discretization of the test case. Triangulate the domains with the keyword **Trianguler_H** (See the syntax in the User's manual). Give an unstructured aspect to the 2 meshes thanks to the following keyword :
  Transformer name_of_domain x*(1-0.5*y*y) y*(1+0.1*x*y)
- Substitute the discretization VDF (pressure nodes at the element center) to VEFPreP1B (pressure nodes at the element center and nodes).
- Check the meshes with:
  trust -mesh docond
- Run the calculation with:
  trust -monitor docond
- Follow the residuals (plot number 2) and a probe evolution (for example temperature), to see when convergence is reached.

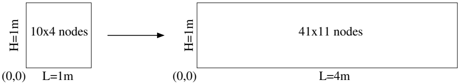
Heat exchange VDF/VEF exercise

- Post-process the results and compare with VDF discretization about the CPU performance: the VEF calculation is running 10 times slower (cause higher number of pressure unknowns and smaller time steps). Study the docond.out file to see the time steps for each equation.
- Accelerate the calculation by impliciting the diffusion operators of each equation with **diffusion_implicite** option in the explicit Euler scheme (check again the User's manual: **trust -doc &**). Run the calculation.
- Use a full implicit scheme now (suppress **diffusion_implicite**), by substituting **Scheme_Euler_Explicit** by **Schema_Euler_implicite** and adding the **Implicite** solver. Have a look at the User's manual for the **gmres** options and define accordingly to the advices given into the User's manual, a value for **facsec, facsec_max**. You block will looks like:
  Solveur Implicite { solveur gmres { diag seuil 1e-30 nb_it_max 5 impr } seuil_convergence_implicite 0.01 }
- Run the calculation.

1. Initialization
2. Obstacle VDF
   - Obstacle: Sequentiel calculation
   - Obstacle: Parallel calculation
   - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. **Low mach number flow**
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. Tank filling (3D-two phases flow)
11. Meshing tools
    - Salomé to create a 3D VEF mesh
      - Cylinder
      - Revolution
      - T-shape
    - Gmsh
      - 2D VEF mesh
      - 3D VEF mesh
12. Validation form
13. Index

Low mach number flow

- Open a terminal and create a directory using Unix commands, and copy the study **TP_Temp_QC_VEF** (it is a 2D simulation of helium gas flow from left to right between two heated walls):
  mkdir -p ∼/test/yourname
  trust -copy TP_Temp_QC_VEF
  cd TP_Temp_QC_VEF
- Open the TRUST User's manual with (it will be useful to search for keywords in this exercice): **trust -doc &**
- Edit the data file with your favourite editor (**nedit** is recommended because it is configured to recognize TRUST syntax):
  nedit TP_Temp_QC_VEF.data &
- Modify the data file in order to:
  - Change the geometry and the mesh:



Low mach number flow

- Add several probes (velocity, volume mass, temperature) near the upper right corner of the geometry at location (x,y)=(4,1).
- Add a probe **"segment"** (with 9 points) between the locations (x,y)=(4,0.05) and (x,y)=(4,0.95) for the temperature field.
- Write the results with the **LATA** format and change the **dt_post** period to 1s.
- We are looking for the stationary state, so suppress tmax keyword and change the **seuil_statio** ε value to 10 ($|dT/dt| < \varepsilon$ and $dt \sim 0.001s$ so $|dT| < 0.01$).
- Add the keyword **impr** into the pressure solver to print its convergence.

- Run the simulation with the TRUST command:
  trust -monitor TP_Temp_QC_VEF
  - Check the convergence information by entering 0 (convergence monitoring in the menu). Visualize the 3 plots:
    - Conjugate Gradient iterations number at each time step,
    - Error evolution of Conjugate Gradient resolution at each time step,
    - Time step evolution.

Low mach number flow

- And check mass flow rate (absolute and relative values) in the TP_Temp_QC_VEF.out file:
  nedit TP_Temp_QC_VEF.out &
- Once the calculation finishes, visualize the results by running **VisIt**:
  visit -o TP_Temp_QC_VEF.lata &
- See the mesh (Plots: "Add → Mesh → dom → Draw").
- Visualize the temperature field (Select the last Time with the slicer, then Plots: "Add → Pseudo Color → TEMPERATURE_SOM_dom → Draw")
- Suppress or hide the mesh (Select "Mesh-dom" in the list of plots then "Delete" or "Hide/Show")
- Visualize the velocity field (Add → Vector → VITESSE_SOM_dom → Draw).
- Select the Zoom mode with the right button of the mouse (Mode → Zoom) then zoom by selecting an area in the plot. To de-zoom push "Ctrl" button and select an area with the left button or with the right button select "View → Reset view".
- Print your visualization (File → Set Save options → File type → Select a type → Save): a file named visit*** is created into your study directory.

Low mach number flow

  - ○ Add a second screen with "Window → Layout → 1x2".
  - ○ Plot a temperature horizontal profile (Select the temperature field and thanks to the right button, select "Mode → Lineout", and define your profile with left button): the profile is shown on the second window.

- Substitute the time scheme by an implicit time scheme (like **schema_euler_implicite**).
- Use the **implicite** solver and specify **facsec** and **facsec_max** parameters according to the advices given into the User's manual (search for the **schema_euler_implicite** keyword for the according syntax). You can also see the informations at the end of the Heat exchange VDF/VEF exercise at p.29.
- Run the calculation with this time scheme:

  trust TP_Temp_QC_VEF.data 1>TP_Temp_QC_VEF.out 2>TP_Temp_QC_VEF.err
- Edit the following file where is located information about time-step used dt, stability time-step dt_stab, **facsec** evolution (dt=dt_stab*facsec), residuals for each equation:

  nedit TP_Temp_QC_VEF.dt_ev &

Low mach number flow

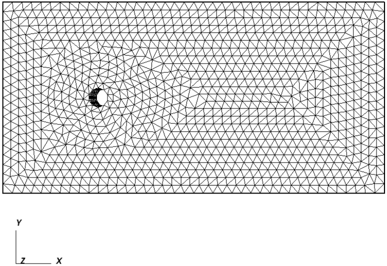
- If everything is OK, try to fine tune the speed convergence of the implicit solver with the value of **seuil_convergence_implicite** keyword (look inside the TP_Temp_QC_VEF.out file and check that the number of iterations for GMRES is between 3 and 5, it is enough to quickly converge).
- To restart a calculation, you will need to change the **tinit** value into the data file (pick up the last save time in the **.err** file) and insert into the data file in the problem definition the following keywords:
  reprise binaire TP_Temp_QC_VEF_pb.sauv
- Then run the calculation with:
  trust TP_Temp_QC_VEF.data 1>TP_Temp_QC_VEF.out 2>TP_Temp_QC_VEF.err
  or
  trust -monitor TP_Temp_QC_VEF.data which creates automaticaly the .out and .err files.

1 Initialization
2 Obstacle VDF
- Obstacle: Sequentiel calculation
- Obstacle: Parallel calculation
- Obstacle: Parallel calculation on a cluster

3 Heat exchange VDF/VEF
4 Low mach number flow
5 **Periodic 3D channel flow**
6 Constituents and turbulent flow
7 3D turbulent flow in a curved pipe
8 Turbulent flow on a 3D step
9 Tank filling (2D-single phase flow)
10 Tank filling (3D-two phases flow)
11 Meshing tools
- Salomé to create a 3D VEF mesh
  - Cylinder
  - Revolution
  - T-shape
- Gmsh
  - 2D VEF mesh
  - 3D VEF mesh

12 Validation form
13 Index

Periodic 3D channel flow



Fluid: $Re = 2000$, $\rho = 2 kg.m^{-3}$, $\mu = 0.01 kg.m^{-1}.s^{-1}$, initial velocity $V0 = 1 m/s$, periodic boundary condition on Z direction.

- Copy the study named **P1toP1Bulle** like explained page 7. It simulates a 3D incompressible laminar flow ($Re = 2000$) with periodic boundary according Z direction only.
- Open the P1toP1Bulle.data file and use **RegroupeBord** keyword to merge Entree and Sortie boundaries into a single one named periox.

Periodic 3D channel flow

- Change the boundary conditions to apply a periodic boundary on the new boundary.
- Change the velocity initial condition to $U_0 = (1, 0, 0)$.
- Set the option **diffusion_implicite** to 1 into the Euler scheme to implicit the diffusive operator of the Navier Stokes equation.
- You have now a 3D calculation with periodic condition on X and Z directions. Run the calculation on 30 time-steps (keyword **nb_pas_dt_max**).
- Have a look at the P1toP1Bulle_pb_Debit.out file, check the flow rate on the periox boundary. Why does it decrease?
- Add the **Canal_perio** source term in the Navier Stokes equation of the data file and run again the calculation to check the flow rate evolution on 30 time steps.
- Look for the pressure and viscous forces applied to the cylinder inside the .out files.

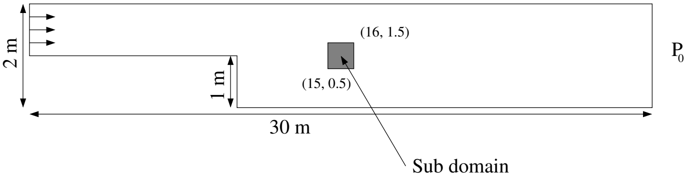
Periodic 3D channel flow

- Now, the calculation domain is a rotating channel according to Z direction with a constant velocity $\Omega = 1rad/s$.
- Add the **Acceleration** source term in the Navier Stokes equation. Suppress the **nb_pas_dt_max** keyword and set **tmax** to 100s.
- Add, if you wish, velocity or statistic calculation in the post processing instructions.
- Run the calculation.
- You can create a uniformly refined mesh using for instance the keyword **Raffiner_Anisotrope**.
- Then improve the calculation speed on this mesh, you could try to use a coarse discretization **P1** (**Read dis** { **P1** } keywords) with less pressure unknowns. On this mesh, it runs 3 times faster than P1Bulle but with less accuracy: 8452 unknowns compare to 49221 unknowns.

Periodic 3D channel flow

- Then restart the calculation with **VEFPreP1B** discretization by reading the velocity field with **Champ_fonc_reprise** keyword in the initial conditions for the velocity:
  vitesse champ_fonc_reprise P1toP1Bulle_pb.xyz pb vitesse last_time
  This will be useful to reach the quasi-stationary state faster.
- You can also use implicit scheme (change the scheme to **Schema_Euler_implicite** scheme and use an **Implicite** solver) only if you are looking for the stationary state.
  You can also see the informations at the end of the Heat exchange VDF/VEF exercise page 29.

Constituents and turbulent flow



Fluid: $\mu = 3.7\,10^{-5} kg.m^{-1}.s^{-1}, \rho = 2\;kg.m^{-3}, R_e = \frac{U_0 H_{inlet} \rho}{\mu} = 54054$
Boundary conditions:
Inlet with fixed velocity: $U_0 = 1 m.s^{-1}$, C1_eps=10^{-2}, C2_eps=10^{-3} (dimensionless values)
Outlet with a fixed pressure: $P_0 = 0$, C1_eps=0, C2_eps=0
Upper and lower walls: No slip wall ($U = 0$) and non-porous wall ($\partial C_i / \partial n = 0$)

- Copy the study named **Marche** like explained page 7. It simulates a 2D incompressible turbulent flow with a K-Eps model in the geometry described above.

Constituents and turbulent flow

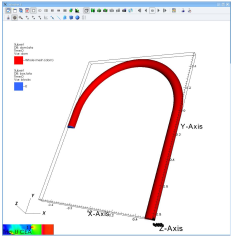
- Copy also the **Constituants** study which will give you examples for constituents keyword used for a 2D incompressible laminar flow.
- Edit your data file in the Marche directory. First, change the name of the problem in order to add concentration equations (look for the good keyword in the User manual).
- Add 3 constituents with the same diffusivities ($\alpha = 1m/s$) and associate the constituents to the problem.
- Define the concentration equation into the problem (remember that concentrations will be a vector of 3 components) with correct initials ($C_1 = 0, C_2 = 0, C_3 = 0$) and boundaries conditions.
- Use the Schmidt model to close the turbulence modelization in the concentration equation.
- Change the sources of the Navier-Stokes turbulence model to a **Source_Transport_K_Eps_aniso_concen** { **C1_eps 1.44 C2_eps 1.92 C3_eps 1.** } to fit with the new concentration equation.

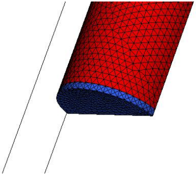
Constituents and turbulent flow

- Add into the fluid definition, the volume expansion coefficient for the concentration: **beta_co** as an uniform field fixed to 0.
- You have also to add a gravity field which can be initialized to 0.
- Run the calculation to see if it is ok.
- Define a sub domain (in grey on the previous picture) with the keyword **Sous_Zone** (like in **PCR** data file).
- Add a source term for the second constituent only ($S_2 = 1m^{-3}$) applied on the sub domain thanks to the keyword **Champ_Uniforme_Morceaux**.
- Add **format lata** in te post-processing block.
- Add the keyword **concentration0, concentration1, concentration2** in the **fields** of the post-processing block to write the 3 concentrations into the .lata results file.
- Run the calculation and check the results.

3D turbulent flow in a curved pipe

TrioCFD





Goals:

- Use of a RANS or LES model.
- Use of a periodic box to initialize a fully developed turbulent flow.
- Use of the TrioCFD parallel capabilities.

3D turbulent flow in a curved pipe

TrioCFD

- First initialize TrioCFD environnement :

```
source /home/triou/env_TrioCFD_X.Y.Z.sh
echo $exec
export TrioCFD_ROOT=` dirname $exec`
echo $TrioCFD_ROOT
```

- Create a new directory and copy some data:

```
mkdir -p ~/test/yourname/PeriodicBox
cd ~/test/yourname/PeriodicBox
cp $TrioCFD_ROOT/validation/share/Validation/Rapports_automatiques/
                              Validant/pas_fini/PeriodicBox/src/*   .
```

Notice that this directory correspond to an automated validation test case. If you want to run it and generated the pdf report, see the last exercise of this tutorial named "Validation form", but be carefull this case is very slow !

3D turbulent flow in a curved pipe

TrioCFD

- There are several files :
  - *BuildMeshes.data*: To build the meshes
  - *PeriodicBoxRANS.data*: To run the flow in the box with RANS model
  - *DomainFlowRANS.data*: To run the flow in the domain with inlet steady conditions from the box domain
  - *PeriodicBoxLES.data*: To run the flow in the box with LES model
  - *DomainFlowLES.data*: To run the flow in the domain with inlet unsteady conditions from the box domain
- First, edit and read the BuildMeshes.data file.
- If you wish to run a RANS simulation, open the PeriodicBoxRANS.data and DomainFlowRANS.data files.
- Or if you wish to run a LES simulation, open the PeriodicBoxLES.data and DomainFlowLES.data files.

3D turbulent flow in a curved pipe

TrioCFD

- Then build the meshes:
  ./prepare
  trust BuildMeshes
  Notice that we use **"trust"** command line because this script will use the variable $exec which is the path to the TrioCFD executable.
- You can visualize the partitioned meshes with (MODEL=RANS or LES):
  trust -mesh PeriodicBoxMODEL
- Then, fix the maximal number of time steps in the time scheme using **nb_pas_dt_max** to 100 in the files PeriodicBoxRANS and PeriodicBoxLES and run the 2 cores parallel calculations, to initialize the turbulent flow in the box:
  trust PeriodicBoxRANS 2
  trust PeriodicBoxLES 2
  (The full calculation in RANS takes approximatively 1h and 10h in LES.)

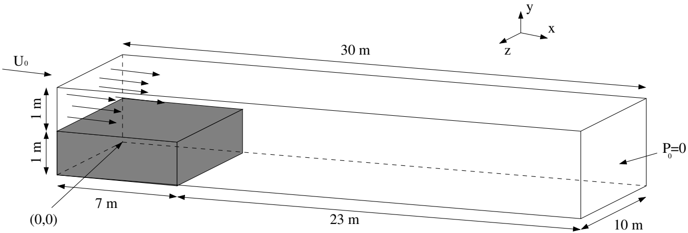
3D turbulent flow in a curved pipe

TrioCFD

- Open the file PeriodicBoxRANS.dt_ev and PeriodicBoxLES.dt_ev to read the last time of the two calculation after 100 time steps.
- Once finished, open the file DomainFlowRANS.data and DomainFlowLES.data and change the maximal number of time steps to 10.
- You can see that theses data files are constitued by 2 problems, one for the box and one for the domain.
  - We use the velocity and temperature fields of the last time step of the PeriodicBoxRANS (or LES) calculation into the initiales conditions of the pb_box problem with the keyword "Champ_fonc_reprise".
  - In addition in the pb_dom, the velocity and temperature fields of the pb_box are used in the initiales conditions of the pb_dom throught the keyword "champ_front_recyclage".
- Run a 6 cores parallel calculation of the domain (it will stop by default after 10 times steps):
  trust DomainFlowRANS 6
  trust DomainFlowLES 6

Turbulent flow on a 3D step

TrioCFD



Meshing: 30x10x10 ($\Delta x = 1m, \Delta y = 0.2m, \Delta z = 1m$)
Fluid: $\mu = 5.10^{-5} kg.m^{-1}.s^{-1}, \rho = 2kg.m^{-3}$
Boundaries conditions: with in entry $R_e = \frac{U_0 H_{inlet} \rho}{\mu} = \frac{1\times1\times2}{5.10^{-5}} = 40000$

Inlet: $U_0 = 1m.s^{-1}$

Outlet: $P_0 = 0$

Turbulent flow on a 3D step

TrioCFD

- First initialize TrioCFD environnement :
  source /home/triou/env_TrioCFD_X.Y.Z.sh
  echo $exec
  export TrioCFD_ROOT=` dirname $exec`
  echo $TrioCFD_ROOT
- Copy the study named Marche3D: **trust -copy Marche3D**
- Edit the data file and:
  - Note that we use a "Pb_Hydraulique_Turbulent" problem with "Navier_Stokes_Turbulent" equations and a "modele_turbulence" model.
  - Modify the fluid characteristics to perform a calculation at $R_e = 50000$. For example, impose $\rho = 1kg.m^{-3}$ and $\mu = 2.10^{-5}kg.m^{-1}.s^{-1}$.
  - Select the sub-grid Smagorinsky turbulence model with standard wall law instead of the "sous_maille" model (LES).
  - Select the Quick convection scheme.
  - Post-process of velocity, pressure, vorticity, turbulent viscosity at the nodes and elements.

Turbulent flow on a 3D step

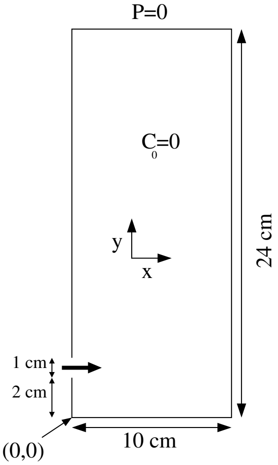
TrioCFD

- Run the calculation and post-process the main calculated fields.
  trust Marche3D
- Notice that we use **"trust"** command line because this script will use the variable $exec which is the path to the TrioCFD executable.
- Replace the sub-grid model by the standard k_eps model (RANS).
- Run the calculation and post-process of the velocity field to see the differences between the different turbulence models used.

1. Initialization
2. Obstacle VDF
   - Obstacle: Sequentiel calculation
   - Obstacle: Parallel calculation
   - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. **Tank filling (2D-single phase flow)**
10. Tank filling (3D-two phases flow)
11. Meshing tools
    - Salomé to create a 3D VEF mesh
      - Cylinder
      - Revolution
      - T-shape
    - Gmsh
      - 2D VEF mesh
      - 3D VEF mesh
12. Validation form
13. Index

Tank filling (2D-single phase flow)

We want to simulate the following flow:



Fluid: Coloured water
diffusion $D = 10^{-9} m^2.s^{-1}$,
$\rho = 1000 kg.m^{-3}$, $\mu = 10^{-3} kg.m^{-1}.s^{-1}$

Boundary condition:
Inlet: Velocity: $(V_x, V_y) = (V(t), 0)$

with $V(t) = \begin{cases} 1 - (y - 0.025/0.005)^2 & , \ t \leq 0.5s \\ 0 & , \ t > 0.5s \end{cases}$

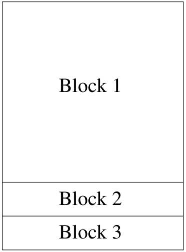
Concentration: $C = \begin{cases} 1 & , \ t \leq 0.5s \\ 0 & , \ t > 0.5s \end{cases}$

Outlet: Pressure $P = 0$
Wall: Velocity $V_x = 0$, $V_y = 0$

Initial condition: Concentration $C_0 = 0$,
Velocity $V = 0$

Tank filling (2D-single phase flow)

- Source the TRUST environnement:
  source /home/triou/env_TRUST_X.Y.Z.sh
- Copy the study named **diagonale**. This test case deals with a 2D flow with Navier Stokes and the equation for one constituent.
- Edit the data file and modify the fluid characteristics to the previous ones (μ, ρ, D).
- We want to modify the geometry of this problem to the previous picture. So we want to create 3 blocks like:



Block 1

Block 2

Block 3

Tank filling (2D-single phase flow)

- Create the corresponding mesh with 3 blocks (start with $dx = dy = 0.2cm$ which gives a total node number $Nx = 51$ and $Ny = 121$).
  - Create a first block "Block1" which origin is (0, 0.03), $Nx = 51$, $Ny = 121$ (for $dx = dy = 0.2cm$), $L = 0.1m$, $H = 0.21m$. Name the wall boundaries Left1, Outlet(=Top1) and Right1. (Don't forget the comma between two block definition.)
  - Create the second block "Block2" which origin is (0, 0.02), $Nx = 51$, $Ny = 6$ (for $dx = dy = 0.2cm$), $L = 0.1m$, $H = 0.01m$. Name the wall boundaries Inlet(=Left2) and Right2.
  - Create the third block "Block3" which origin is (0, 0), $Nx = 51$, $Ny = 11$ (for $dx = dy = 0.2cm$), $L = 0.1m$, $H = 0.02m$. Name the wall boundaries Left3, Bottom3 and Right3.
- Define the boundary wall, using the keyword **"RegroupeBord"**.
- You could also use facteurs and symx, symy keywords to define a refined mesh near the walls.
- Check the mesh with: **"trust -mesh diagonale"** and correct the mesh errors if needed.

Tank filling (2D-single phase flow)

- In the data file, change the values in the time scheme to finish the calculation at 1 second, and modify **dt_min** and **dt_max** values to let TRUST calculates itself its time step.
- Change values for the gravity to $-9.81 m.s^{-2}$ following y axe.
- Note that the **beta_co** keyword may be useful in order to have a Boussinesq coupling between momentum and concentration equations ($\beta C_0 g(C - C_0)$ source term added in Navier Stokes equation).
- Change the initial and boundary conditions for Navier Stokes equation:
  - for the Outlet boundary, you have to impose $P = 0$,
  - for the Wall boundary, you have to impose $V_x = V_y = 0$ with **"paroi_fixe"** keyword.,
  - for the Intlet boundary, you have to impose $(V_x, V_y) = (V(t), 0)$ with $V(t) = \begin{cases} 1 - (y - 0.025/0.005)^2 & , t \leq 0.5s \\ 0 & , t > 0.5s \end{cases}$. You will use the **Champ_Front_Fonc_txyz** keyword for the velocity, to write something like: **Champ_Front_Fonc_txyz** 2 $(1 - ((y - 0.025)/0.005)^2) * (t < 0.5)$ 0.
    Note: Use $(t[0.5)$ syntax if you prefer $(t <= 0.5)$

Tank filling (2D-single phase flow)

- Change the initial and boundary conditions for the constituent equation.
  - You will also use **Champ_Front_Fonc_txyz** field for the Inlet boundary condition for concentration.
  - For the Outlet, use the following keywords to insure the external concentration is 0: **Frontiere_ouverte C_ext Champ_front_uniforme 1 0.**
  - For the Wall, the keyword for impermeable boundary condition for concentration is **paroi**.
- Check you have high order schemes (i.e. **"Quick"** scheme) used in both equations to reduce numerical diffusion.
- Notice you could have suppressed diffusion term in concentration equation rather than using a small diffusion coefficient with:
  Diffusion { negligeable }
- Add a concentration probe near the inlet (e.g.: at (0,0.025)).
- Add a velocity segment probe (with 5 points between (0,0.021) and (0,0.029)) at the inlet boundary to see the time evolution of these two quantities (period 0.01s).

Tank filling (2D-single phase flow)

- Run the study and follow the time evolution with the probes:
  "trust -monitor diagonale"
- Check the flow rate in inlet boundary in the diagonale_pb_Debit.out file. You should find a value near 6.8 $10^{-3} m^2.s^{-1}$.
- Use VisIt to post process the results at t=0.2, t=0.4s and t=0.7s. VisIt has some interesting feature for this study. It can give concentration histogram to check the numerical diffusion in the concentration equation: Add → Histogram → CONCENTRATION_ELEM_dom. The volume of colored water (in m^3) is given by $Vol(t) = 6.66.10^{-3} t$ before $t = 0.5s$ and $Vol(t) = 3.33.10^{-3}$ after.

Tank filling (2D-single phase flow)

→ VEF

- Copy diagonale.data to diagonale_VEF.data.
- Triangulate your mesh (**trianguler** keyword).
- In this new file, change the discretization (**VEFPreP1B** instead of **VDF**).
- Use **muscl** instead of **quick** scheme.
- And you can switch **GCP** solver by **Cholesky** solver of the Petsc library (direct method which may need large amount of RAM memory) to increase the speed resolution of the pressure linear system:
  GCP { **precond ssor** { **omega 1.5** } **seuil 1.e-6** } → **Petsc Cholesky** { }
- Run the calculation. You must have an error, and TRUST stop the calculation.

Tank filling (2D-single phase flow)

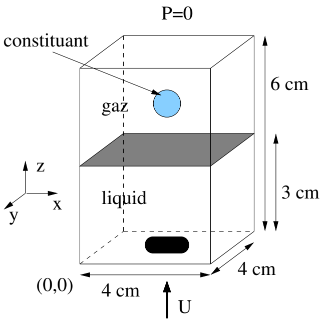
→ VEF

- As TRUST explain, to avoid this problem, you can:
  - change the **trianguler** keyword to **trianguler_h**,
  - or use the **VerifierCoin** keyword. For this, after this first error you must find a "diagonale_VEF.decoupage_som" file in your directory, so you can use it by adding:
    VerifierCoin dom { **read_file diagonale_VEF.decoupage_som** }
    just after "trianguler dom". This will subdivides inconsistent 2D/3D cells used with VEFPreP1B discretization (cf User's Manual).
- Run the calculation and compare the results between **VDF/quick** and **VEFPreP1B/muscl** which must take much more time!

1. Initialization
2. Obstacle VDF
   - Obstacle: Sequentiel calculation
   - Obstacle: Parallel calculation
   - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. **Tank filling (3D-two phases flow)**
11. Meshing tools
    - Salomé to create a 3D VEF mesh
      - Cylinder
      - Revolution
      - T-shape
    - Gmsh
      - 2D VEF mesh
      - 3D VEF mesh
12. Validation form
13. Index

Tank filling (3D-two phases flow)

TrioCFD



Liquid: $\rho = 1000 kg.m^{-3}$,
$\mu = 2,82.10^{-4} kg.m^{-1}.s^{-1}$,
$\sigma = 0.05 N.m^{-1}$, $D = 10^{-6} m^2.s^{-1}$

Gas: $\rho = 100 kg.m^{-3}$,
$\mu = 2,82.10^{-4} kg.m^{-1}.s^{-1}$

Boundary conditions:
Up : Free outlet, Wall : $V = 0$
Down: $V(x,y,z) = (0,0,10^{-3} m.s^{-1})$

Initial conditions: $V = 0$,
$C = e^{(-((x-0.02)^2+(y-0.02)^2+(z-0.03)^2)/0.03^2)}$

N.B.: The interface between the air and the gaz is a parabolic function.

Tank filling (3D-two phases flow)

TrioCFD

- First initialize TrioCFD environnement :
  source /home/triou/env_TrioCFD_X.Y.Z.sh
  echo $exec
- Copy the study named **FTD_all_VDF**: **trust -copy FTD_all_VDF**
- This test case deals with a 3D two phase flow in a tank with one initial interface between liquid and gas, a droplet, and a rotating solid in the liquid. The model used is the Discontinuous Front Tracking method and the mesh is 3D structured mesh.
- Notice that:
  - 2D Discontinuous Front Tracking method has not been intensively tested yet.
  - the type of the problem: **Probleme_FT_disc_gen** in the data file. This refers to the Discontinuous Front Tracking method.
  - the keyword **modele_turbulence**. Navier-Stokes equation of the Discontinuous Front Tracking problem needs the read of this keyword even if the flow is laminar. In this case, use the **nul** keyword just after **modele_turbulence**. Else, specify the turbulence model chosen.

Tank filling (3D-two phases flow)

TrioCFD

- Increase the height of the tank (0.06 to 0.12).
- Add a second drop above the first one, at $z = 0.08$ (keywords **ajout_phase0** could be useful to add other interfaces, cf User's manual for **ajout_phase0/ajout_phase1** keywords). Don't forget the comma between the two definition of the drops.
- Change the **dt_post** period of the 3 post processing blocks (0.05 to 0.01). The first one (add **format lata**) is the classical block for post processing probes and fields. Here, we want to see the concentration field and the **"indicatrice_interf"** field. Value of this field is 0 for liquid and 1 for gas, so the interface is located at "indicatrice" value 0.5.
- Change the interpolation location of **indicatrice_interf** and the **concentration** fields in the first post processing block, by adding the keyword **elem** just after the fields: the values in the post processing tool will be plotted at the centre of each element of the mesh.

Tank filling (3D-two phases flow)

TrioCFD

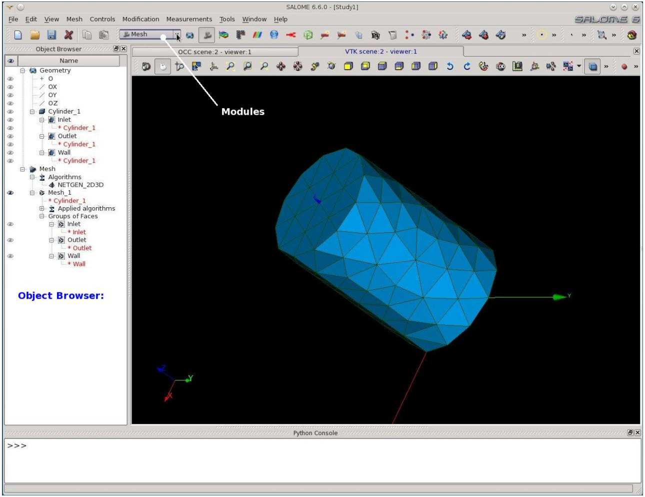
- The second post processing block is the new syntax to post process interfaces moving meshes. You can visualize it with **VisIt**.
- On each interface, you can plot several fields ie: curvature with **courbure** keyword and velocity interface with **vitesse** keyword, **pe** field is for debugging purpose, it is useless here, you can suppress it) on several locations (on nodes with **sommets** keyword, on cells with **elements** keyword).
- Run the calculation. Follow the time step evolution by having a look at the **dt_ev** file. It contains on each line the physical time, the time step, security factor and residuals.
- Post process to visualize the interface and the concentration field.
- You can increase the number of cells to have a finest simulation or also change to VEF discretization.

1 Initialization
2 Obstacle VDF
- Obstacle: Sequentiel calculation
- Obstacle: Parallel calculation
- Obstacle: Parallel calculation on a cluster

3 Heat exchange VDF/VEF
4 Low mach number flow
5 Periodic 3D channel flow
6 Constituents and turbulent flow
7 3D turbulent flow in a curved pipe
8 Turbulent flow on a 3D step
9 Tank filling (2D-single phase flow)
10 Tank filling (3D-two phases flow)
11 Meshing tools
- Salomé to create a 3D VEF mesh
  - Cylinder
  - Revolution
  - T-shape
- Gmsh
  - 2D VEF mesh
  - 3D VEF mesh

12 Validation form
13 Index

Salomé to create a 3D VEF mesh: Cylinder


Modules

Object Browser:

Cylinder

Create a geometry

- Create a directory and run Salomé (we suppose it is installed):

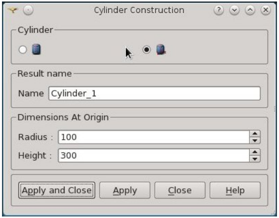
  ```
  mkdir -p ~/test/yourname/salome/exo1
  cd ~/test/yourname/salome/exo1
  /export/home/salome/SALOME-7.7.1-FD18/salome
  # or if no local version: /home/salome/salome
  # take the network version but to be used for testing NOT for work.
  ```

- Create a new study: File → New
- Select the Geometry module into the SALOME drop-down menu (contains all the modules).
- Create a first geometry with: New Entity → Primitives → Cylinder
- Save your study in hdf format (Salome format) frequently.

Cylinder

Create a geometry

- Specify Radius $R = 100$ and Height $H = 300$ for the cylinder (the default values). Then Apply and Close.



Cylinder Construction

Cylinder: ○ ●

Result name

Name: Cylinder_1

Dimensions At Origin

Radius : 100

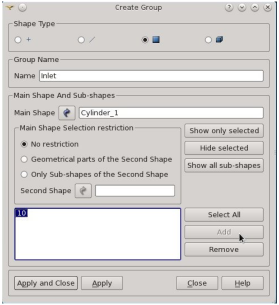
Height : 300

- Rotate, zoom, move the geometry by switching to "Interaction style switch": Mouse icon.
- Create groups for the geometry to define the top, the bottom and the lateral parts of the cylinder: New Entity → Group → Create Group
- Select the good Shape Type (→ □ surface).

Cylinder

Create a geometry

- Give a Group Name for the top: "Inlet".
- Click on the arrow button of the Main Shape field and select the "Cylinder_1" in the "Object browser" or in the visualisation window.
- Select the shape defining the top of the cylinder on the visualisation window then Add → Apply.
- Select the shape defining the part on the window then Add → Apply.



Create Group

Shape Type: ○ + ○ ⁄ ● ■ ○ ▣

Group Name
Name: Inlet

Main Shape And Sub-shapes
Main Shape: Cylinder_1

Main Shape Selection restriction
● No restriction
○ Geometrical parts of the Second Shape
○ Only Sub-shapes of the Second Shape
Second Shape:

10

Cylinder

Create a geometry

- Do the same for the two other parts:
  - For the lateral: "Wall"
  - For the bottom: "Outlet" (you can rotate the cylinder to click on the bottom).
- Close the window once the 3 groups has been created. Check that they appear in the Object Browser (by clicking on the "▷" in front of the "Cylinder_1" object).

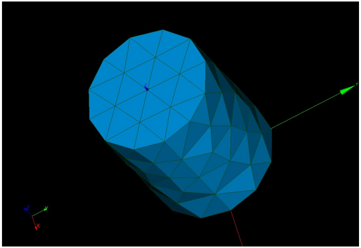
Create a mesh

- Now, switch to the Mesh module in the SALOME drop-down menu.
- Select the Cylinder_1 in the Object Browser and Right Click → 'Show' to visualize the geometry or click on the 'eye' next to the Cylindre_1 object.
- Create a mesh with: Mesh → Create Mesh
- Select the Geometry used for the mesh if not selected by clicking on the Cylinder_1 object in the Object Browser.

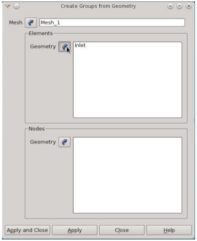
Cylinder

Create a mesh

- Choose Netgen 1D-2D-3D algorithm and click on "Apply and Close".
- Select the object Mesh_1 in the Object Browser and Right Click → Compute (or Mesh → Compute).
- A tabular must appear with the number of triangles, quadrangles... Click on "Close".
- Hide the geometry by selecting the Cylinder_1 in the Object Browser and Right Click → Hide (or click on the eye).



Mesh



Create Groups from Geometry

Mesh: Mesh_1

Elements

Geometry: Inlet

Nodes

Geometry:

Create Groups from Geometry

Cylinder

Export your mesh in MED format

- Now, we are going to create groups for the mesh, by using the geometry groups: Mesh → Create Groups from Geometry.
- Click on "Mesh_1" object in the Object Browser to select it.
- Then push the arrow button of Elements/Geometry if it's not already done.
- Select together the 3 groups (Inlet, Wall and Outlet) with CTRL+left click in the Cylinder_1 object of the Object Browser, then press Apply and Close.
- Check that the 3 boundaries are in the "Group of Faces" of the Mesh_1 object in the Object Browser.
- Export your mesh with the MED format :
  Select the Mesh_1 object then Right Click → Export → MED file (or File → Export → MED file).

Cylinder

Read your mesh with TRUST

- Now build a data file named dom.data for TRUST:

```
dimension 3
domaine dom
Lire_med family_names_from_group_names dom Mesh_1 Mesh_1.med
Discretiser_domaine dom
Postraiter_domaine { domaine dom fichier mesh format lata }
```

- Run the data file and post process the mesh with VisIt:

```
source /home/triou/env_TRUST_X.Y.Z.sh
trust dom
visit -o mesh.lata
```

Warning: The more common error is to forget to define the boundaries with the groups for the mesh. The error in TRUST is printed and detected during the discretization where all the faces of the mesh (in particular the boundary faces) are built.

Cylinder

Refine your mesh and use viscous layers

Goal: Improve the mesh for TRUST near the wall by using viscous layers.

- Create a new mesh named "Refined_mesh" with: Mesh → Create Mesh
- Select the Cylinder_1 geometry in the Object Browser.
- Select the "Tetrahedron (Netgen)" or "MG-Tetra" 3D algorithm.
- Click on the wheel of "Add. Hypothesis" → "Viscous Layers" with:
    - Total thickness: 30
    - Number of layers: 3
    - Stretch factor: 1.1
    - Add to "Faces without layers" the 2 geometry groups "Inlet" and "Outlet" of Cylinder_1 object (select or unselect the mouse icon).
    - Click OK
- Add a 2D algorithm: "Netgen 1D-2D" or "MG-CADSurf".

Cylinder

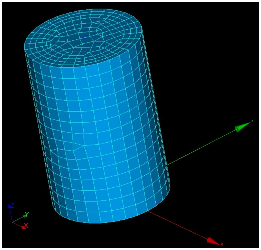
Refine your mesh and use viscous layers

- Click on the wheel of "Hypothesis" → "Netgen 2D parameters" or "MG-CADSurf parameters":
  - For "Netgen 2D parameters":
    - Change "Fineness" from "Moderate" to "Very Fine".
    - Select "Allow Quadrangles".
    - Click OK.
  - For "MG-CADSurf parameters", change "User size" to 20. Click OK.
- "Apply and Close" the close mesh window.
- Select the Refined_Mesh object in the Object Browser and Right click → Compute

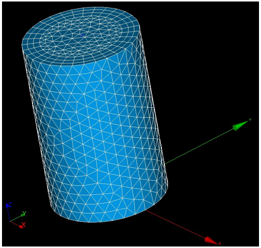
Cylinder

Refine your mesh and use viscous layers

- You should have a refined mesh with a mix of tetra, hexa, pyramid, prism elements for Netgen algorithms, and a mix of tetra and prisms for MG algorithms:



Netgen



MG

- As TRUST accepted only tetras elements, you can quickly tetraedrize:
  - Select the Refined_Mesh in the Object Browser.
  - "Modification" → "Split Volums" and select "Tetrahedron".
  - Don't change the parameters, and click "Apply and Close".

Cylinder

Refine your mesh and use viscous layers

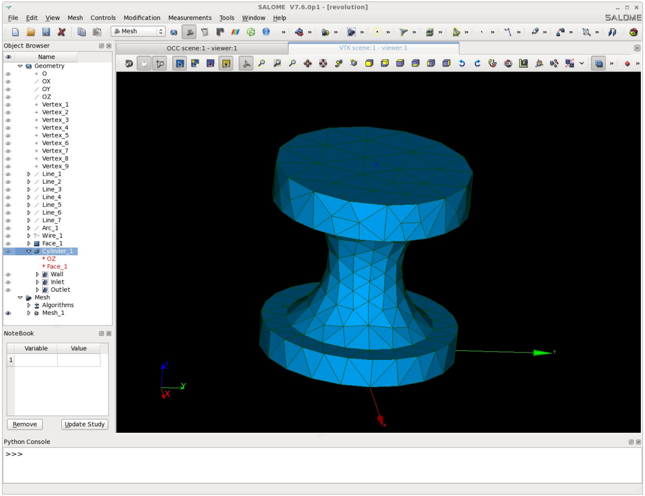
- As usual, define the mesh boundaries with Groups:
  - Select the Refined_Mesh, Right click → Create Groups from Geometry.
  - Select "Inlet", "Outlet", "Wall" with Ctrl+click in the Cylinder_1 object, then Apply and Close.
- Export the mesh:
  - Select the Refined_mesh, Right click → Export → MED file.
  - Save into a Refined_Mesh.med file.
- Save your work in hdf format ("File" → "Save/Save As..."), and in python format with "File" → "Dump Study..."

Run with TRUST

- Change or create a new data file for TRUST to read and visualize your refined mesh.
- **N.B.**: The solutions of the exercise (mesh.py file for the first mesh and prism.py file for the second mesh) are located here: $TRUST_ROOT/doc/TRUST/exercices/salome.

1. Initialization
2. Obstacle VDF
   - Obstacle: Sequentiel calculation
   - Obstacle: Parallel calculation
   - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. Tank filling (3D-two phases flow)
11. **Meshing tools**
    - **Salomé to create a 3D VEF mesh**
      - **Cylinder**
      - **Revolution**
      - **T-shape**
    - Gmsh
      - 2D VEF mesh
      - 3D VEF mesh
12. Validation form
13. Index

Salomé to create a 3D VEF mesh: Revolution



Salomé to create a 3D VEF mesh: Revolution

Create a geometry

- Create a directory and run Salomé (we suppose it is installed):

  ```
  mkdir -p ∼/test/yourname/salome/exo2
  cd ∼/test/yourname/salome/exo2
  /export/home/salome/SALOME-7.7.1-FD18/salome
  # or if no local version: /home/salome/salome
  # take the network version but to be used for testing NOT for work.
  ```

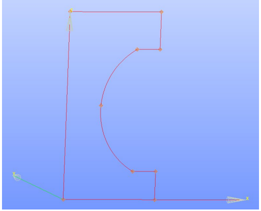
- Create a new study: File → New.
- Select the Geometry module into the SALOME drop-down menu.
- Create points : New Entity → Basic → Point

Vertex_1 (0,0,0)	Vertex_2 (1,0,0)	Vertex_3 (1,0,0.3)
Vertex_4 (0.75,0,0.3)	Vertex_5 (0.375,0,1)	Vertex_6 (0.75,0,1.6)
Vertex_7 (1,0,1.6)	Vertex_8 (1,0,2)	Vertex_9 (0,0,2)

  Then "Apply and Close"

Revolution

Create a geometry



- Create edges:
  - New Entity → Basic → Line
    - Line_1 with Vertex_1 and Vertex_2
    - Line_2 with Vertex_2 and Vertex_3
    - Line_3 with Vertex_3 and Vertex_4
    - Line_4 with **Vertex_6** and **Vertex_7**
    - Line_5 with Vertex_7 and Vertex_8
    - Line_6 with Vertex_8 and Vertex_9
    - Line_7 with Vertex_9 and Vertex_1
    - Then Apply and Close.

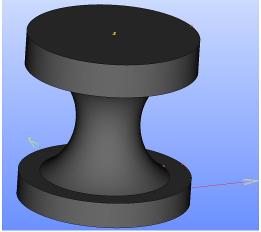
Revolution

Create a geometry

- Create edges:
  - New Entity → Basic → Arc
    - Arc_1 with Vertex_4, Vertex_5 and Vertex_6.
    - Then Apply and Close.
- Create a wire: New Entity → Build → Wire
  - Wire_1 on edges with Line_1,... , Line_7 and Arc_1 (with "Ctrl" button).
  - Then Apply and Close.
- Create a face: New Entity → Build → Face.
  - Face_1 with Wire_1 and "Apply and Close".
- Create a revolution cylinder: New Entity → Generation → Revolution.
  - named Cylinder_1,
  - with Face_1 in Objects,
  - click on the arrow button next "Axis" and select OZ in the Object Browser,
  - set the angle to 360^o and "Apply and Close".

Revolution

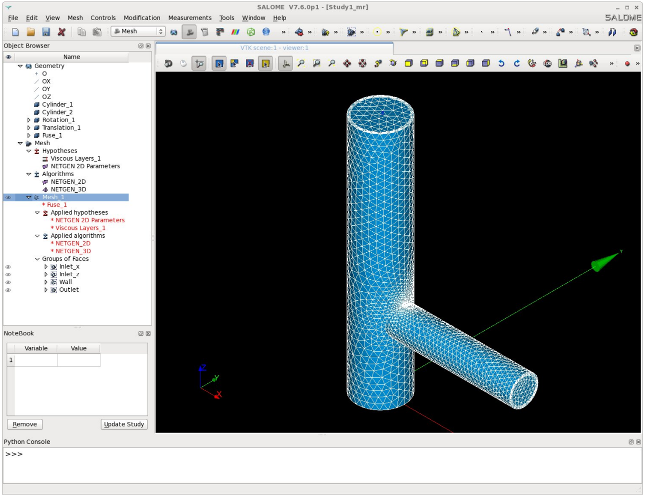
Create a geometry



- Create groups for the geometry to define the top, the bottom and the lateral parts of the cylinder: New Entity → Group → Create Group.
- Save your study in hdf format ("File" → "Save/Save As..."), and in python format with "File" → "Dump Study..."
- Now you can create the mesh in the same way than page 95.
- **N.B.**: You can find the solutions of this exercise (revolution.py) in $TRUST_ROOT/doc/TRUST/exercices/salome.

1. Initialization
2. Obstacle VDF
   - Obstacle: Sequentiel calculation
   - Obstacle: Parallel calculation
   - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. Tank filling (3D-two phases flow)
11. Meshing tools
    - Salomé to create a 3D VEF mesh
      - Cylinder
      - Revolution
      - T-shape
    - Gmsh
      - 2D VEF mesh
      - 3D VEF mesh
12. Validation form
13. Index

Salomé to create a 3D VEF mesh: T-shape



Name
- Geometry
  - O
  - OX
  - OY
  - OZ
  - Cylinder_1
  - Cylinder_2
  - Rotation_1
  - Translation_1
  - Fuse_1
- Mesh
  - Hypotheses
    - Viscous Layers_1
    - NETGEN 2D Parameters
  - Algorithms
    - NETGEN_2D
    - NETGEN_3D
  - Mesh_1
    - * Fuse_1
    - Applied hypotheses
      - * NETGEN 2D Parameters
      - * Viscous Layers_1
    - Applied algorithms
      - * NETGEN_2D
      - * NETGEN_3D
    - Groups of Faces
      - Inlet_x
      - Inlet_z
      - Wall
      - Outlet

Salomé to create a 3D VEF mesh: T-shape

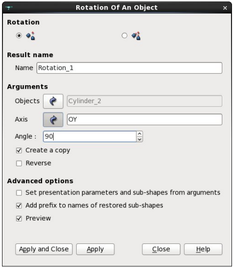
Create a geometry

- Create a directory and run Salomé (we suppose it is installed):
  mkdir -p ~/test/yourname/salome/exo3
  cd ~/test/yourname/salome/exo3
  /export/home/salome/SALOME-7.7.1-FD18/salome
  # or if no local version: **/home/salome/salome**
  # take the network version but to be used for testing NOT for work.
- Create a new study: File → New.
- Select the Geometry module into the SALOME drop-down menu.
- Create two cylinders: New Entity → Primitives → Cylinders:
  Cylinder_1: radius 0.5, height 5. Then "Apply".
  Cylinder_2: radius 0.3, height 3. Then "Apply and Close".
- Save your study in hdf format (Salome format) frequently.

T-shape

Create a geometry

- Rotate Cylinder_2: Operations → Transformation → Rotation
  Name:Rotation_1, Object: Cylinder_2, Axis: 'OY', Angle: 90
  Then "Apply and Close".
- Translate Rotation_1: Operations → Transformation → Translation
  Name: Translation_1, Object: Rotation_1, Dx=Dy=0, Dz=1.5
  Then "Apply and Close".



Rotation Of An Object

Rotation

● ○

Result name

Name: Rotation_1

Arguments

Objects: Cylinder_2

Axis: OY

Angle : 90

☑ Create a copy

□ Reverse

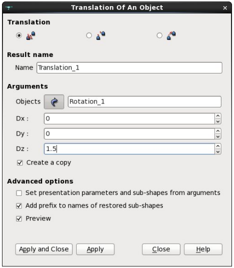
Advanced options

□ Set presentation parameters and sub-shapes from arguments

☑ Add prefix to names of restored sub-shapes

☑ Preview

Rotation



Translation Of An Object

Translation

● ○ ○

Result name

Name: Translation_1

Arguments

Objects: Rotation_1

Dx : 0

Dy : 0

Dz : 1.5

☑ Create a copy

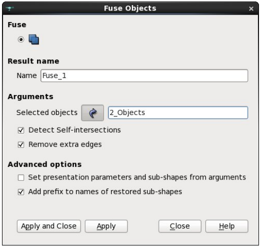
Advanced options

□ Set presentation parameters and sub-shapes from arguments

☑ Add prefix to names of restored sub-shapes

☑ Preview

Translation



Fuse Objects

Fuse

●

Result name

Name: Fuse_1

Arguments

Selected objects: 2_Objects

☑ Detect Self-intersections

☑ Remove extra edges

Advanced options

□ Set presentation parameters and sub-shapes from arguments

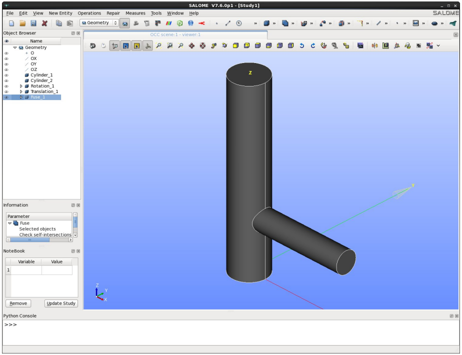
☑ Add prefix to names of restored sub-shapes

Fuse

T-shape

Create a geometry

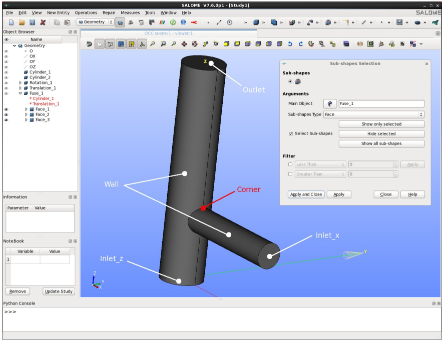
- Fuse Cylinder_1 and Translation_1: Operations → Boolean → Fuse
  Name: Fuse_1, Selected Objects: 2_Objects (use Ctrl key to select Cylinder_1 and Translation_1 in the Object Browser).
  Then "Apply and Close".



T-shape

Create a geometry

- We are now going to create the boundaries: New Entity → Explode
  - Main Object: Fuse_1, Sub-shape type: Face, select "Select sub-shape" and click on the surface Outlet and "Apply".
  - This will create a face named "Face_1" in the Fuse_1 object (click on the "▶"), rename it "Outlet" (by right-clicking and "rename").



SALOME V7.6.0p1 - [Study1]

Object Browser

- Geometry
  - O
  - OX
  - OY
  - OZ
  - Cylinder_1
  - Cylinder_2
  - Rotation_1
  - Translation_1
  - Fuse_1
    - * Cylinder_1
    - * Translation_1
    - Face_1
    - Face_2
    - Face_3

Sub-shapes Selection

Sub-shapes: ●

Arguments

Main Object: Fuse_1

Sub-shapes Type: Face

☑ Select Sub-shapes

Filter

□ Less Than 0

□ Greater Than 0

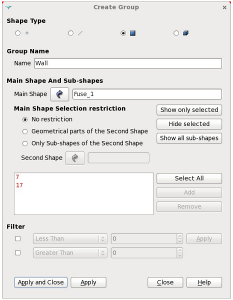
Python Console

```
>>>
```

T-shape

Create a geometry

- Do the same for "Inlet_x" and "Inlet_z".
- We are now going to create the boundary Wall: New Entity → Group → Create group:
  Shape Type: surface, Name: Wall, Main Shape: Fuse_1.
  Click on the surface of the Cylinder_1 then "Add", click on the surface of Translation_1 then "Add" and "Apply and Close".



Create Group

Shape Type

○ ○ ● ○

Group Name

Name: Wall

Main Shape And Sub-shapes

Main Shape: Fuse_1

Main Shape Selection restriction

● No restriction

○ Geometrical parts of the Second Shape

○ Only Sub-shapes of the Second Shape

Second Shape:

7
17

Filter

☐ Less Than 0

☐ Greater Than 0

T-shape

Create a geometry

- We are now going to create the point "Corner": New Entity → Explode
  - Main Object: Fuse_1, Sub-shape type: Vertex, select "Select sub-shape" and click on the choosen point and "Apply and Close".
  - This will create a vertex name "Vertex_1", rename it "Corner" (by right-clicking and "rename").

Create a mesh

- Now, switch to the Mesh module in the SALOME drop-down menu.
- Select the Fuse_1 in the Object Browser and Right Click → 'Show' to visualize the geometry or click on the 'eye' next to the Fuse_1 object.
- Create a mesh with: Mesh → Create Mesh.
- Select the Geometry used for the mesh if not selected by clicking on the Fuse_1 object in the Object Browser.

T-shape

Create a mesh

- Choose "Tetrahedron (Netgen)" for 3D algorithm.
- Click on the wheel of "Add. Hypothesis" → "Viscous Layers" and set:
  - Total thickness: 0.05
  - Number of layers: 3
  - Stretch factor: 1.1
  - Extrusion method: Node Offset
  - Add to "Faces with layers (Wall)" the geometry group "Wall" of Fuse_1 object in the Object Browser (select or unselect the mouse icon). Click on "Add".
  - Click "OK".
- Choose "Netgen 1D-2D" for 2D algorithm.
- Click on the wheel of "Hypothesis" → "Netgen 2D parameters" and set for "Arguments" menu:
  - Max. Size: 0.6
  - Min. Size: 0
  - without "Second Order"
  - Finess: Custom

T-shape

Create a mesh

  - Growth rate: 0.1
  - Nb. segs per Edge: 2
  - Nb. segs per Radius: 4
  - Select "Limit size by Surface Curvature", "Optimize", "Fuse Coincident Nodes on Edges and Vertices".
  - Unselect "Allow Quadrangles".
- For "Local Size" menu:
  - Select "Corner" object in the Object Browser and click on "On Vertex" in the "Hypothesis Construction" window.
  - Double-click on the value in the table and set it to "0.01".
  - Click "OK".
- Click on "Apply and Close".
- Select the Mesh_1 object in the Object Browser and Right click → Compute.
- You should have a mesh with a mix of tetra and prism elements.

T-shape

Create a mesh

- As TRUST accepted only tetras elements, you can quickly tetraedrize:
  - Select Mesh_1 in the Object Browser.
  - "Modification" → "Split Volums" and select "Tetrahedron".
  - Don't change the parameters, and click "Apply and Close".
- Define the mesh boundaries with Groups:
  - Select the Mesh_1, Right click → Create Groups from Geometry.
  - Select "Inlet_x", "Inlet_z", "Outlet" and "Wall" with Ctrl+click in the Fuse_1 object.
  - Then Apply and Close.
- Export the mesh:
  - Select the Mesh_1, Right click → Export → MED file.
  - Save into a Mesh_1.med file.
- Save your study in hdf format ("File" → "Save/Save As..."), and in python format with "File" → "Dump Study..."
- **N.B.**: You can find the solutions of this exercise (T_shape.py) in $TRUST_ROOT/doc/TRUST/exercices/salome.

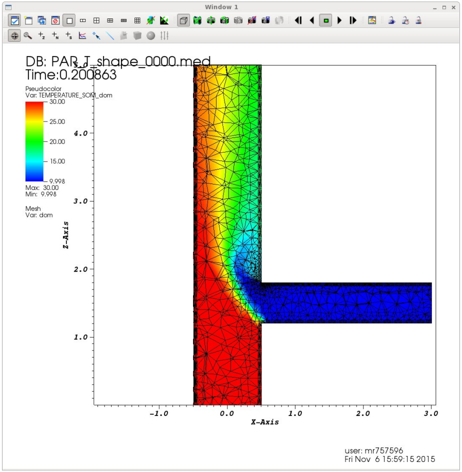
T-shape

Run with TRUST

- Copy the T_shape.data file in your directory:
  cp $TRUST_ROOT/doc/TRUST/exercices/salome/T_shape.data .
- Run it with TRUST:
  trust T_shape
  or in parallel with:
  make_PAR.data T_shape
  trust PAR_T_shape 4
- You can visualize the results with Visit or Salomé by opening the T_shape_0000.med file for sequential calculation or PAR_T_shape_0000.med for parallel calculation.

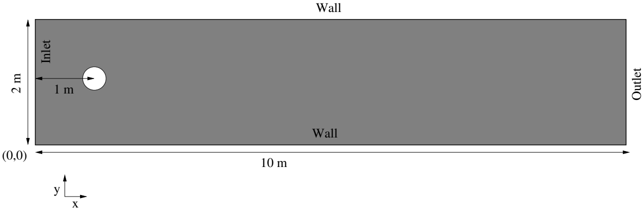
T-shape

Visu with Visit



Gmsh to create a 2D VEF mesh

Geometry which will be created, based on a TrioCFD validation test case geometry



- Create a directory and copy an example:

```
mkdir -p ~/test/yourname/gmsh
cd ~/test/yourname/gmsh
source /home/triou/env_TrioCFD_X.Y.Z.sh
export TrioCFD_ROOT=` dirname $exec`
echo $TrioCFD_ROOT
dir=$TrioCFD_ROOT/validation/share/Validation/Rapports_automatiques/Validant
cp $dir/pas_fini/Drag/src/shape.geo file.geo
nedit file.geo &
gmsh file.geo &
```

Gmsh to create a 2D VEF mesh

- First configure gmsh to show points, lines, and surface numbers of the geometry. In menu Tools → Options → Geometry → Visibility, select Lines, Surfaces, Point labels, Line labels, Surface labels, Volume labels close this window.
- Save definitively your choices with File → Save Options As default.
- Now, look at the file.geo file, you can see the definition of parameters, points, lines... You can see the position of the points and lines with theirs numbers in gmsh.
- Modify the file to suppress the obstacle. We want to keep only 4 points and 4 lines, so you have to suppress the points 5,6,7,8 and the lines 1,3,4,5. For this, you have to:
  - comment (with "//") the definition of the points 5,6,7 and 8.
  - comment the definition of the lines 1,3,4,5. Note that the "Circle" is a line so "Circle(1)=Line(1)".
  - modify the line(2), it will now links points 1 and 2.
  - comment the "Physical Line" named "Shape" which use the line(1) (= Circle(1)) and the line(3).

Gmsh to create a 2D VEF mesh

- ○ suppress the numbers 4 and 5 in the "Physical Line" "Axis". It refers to the lines 4 and 5 which don't exist anymore.
- ○ suppress the numbers 1,3,4 and 5 in the "Line Loop(1)".
- ○ set H to 2 and L to 10. (You can comment D, E, param and X definitions.)

- Press "Reload" in the gmsh GUI → Geometry to update the geometry visualization.
- Now we will had the circle. Note that you can only create circle arcs with angle strictly smaller than π!
  - ○ create the middle of the circle like "Point(10)={1,1,0,lc};" where the triplet "1,1,0" are the coordinates of the point and "lc" the thickness of the cells next this point.
  - ○ create 4 points around this point, which will correspond to the 4 arc of the circle:

    Point(11)={1.25,1,0,lc2};
    Point(12)={1,1.25,0,lc2};
    Point(13)={0.75,1,0,lc2};
    Point(14)={1,0.75,0,lc2};

Gmsh to create a 2D VEF mesh

- ○ define the 4 arcs of the circle with this points:
  Circle(10)={11,10,12};
  Circle(11)={12,10,13};
  Circle(12)={13,10,14};
  Circle(13)={14,10,11}; "
- ○ create a "Physical Line" for the circle:
  "Physical Line("Circle") = {10,11,12,13};"
  This name will be used in your TRUST data file as the name of your boundaries.
- ○ create a line loop for the circle just after the first line loop:
  Line Loop(2) = {10,11,12,13};
- ○ Add the number of this line loop in the "Plane Surface(1)":
  Plane Surface(1) = {1,2};

- Suppress the Physical lines "Axis" and "Top" and create a physical line named "Wall" which regroups the top and the bottom of this geometry (lines 2 and 7).

Gmsh to create a 3D VEF mesh

- Select "Mesh" in the drop-down menu of gmsh and mesh in 2D.
- Export it to a MED file: "File" → "Save As..." and name the file file.med. (Keep the default options.) You can verify your mesh by opening it with gmsh: **gmsh file.med &**.
- Build a TRUST data file with the **Postraiter_domaine** keyword, to read the mesh (like in the Salome exercise page 77). Visualize the mesh with VisIt.
- Then we will try to create a 3D mesh, by using the Extrusions feature of Gmsh. See more about extrusions in http://geuz.org/gmsh/doc/texinfo/gmsh.html.
- Save your initial file in a new one named file3D.geo.
- Comment your "Physical lines", they will not be used here.
- Add the line "Extrude {0,0,1} { Surface{1} ; }" just before the definition of the physical surface.

Gmsh to create a 3D VEF mesh

- Comment the line "Physical Surface("domain") = {1};" in 3D we will have a "Physical Volume" which will be define at the end of the .geo file.
- Define the "Physical Surface" which will be the boundaries of your geometry with the number of the surfaces which can be read on the geometry plotted by gmsh:
  Physical Surface("Inlet") = {38};
  Physical Surface("Outlet") = {30};
  Physical Surface("Wall") = {1,26,34,55};
  Physical Surface("Obstacle") = {42,46,50,54};
- Define you physical volume, you can see its number in yellow in the window:
  Physical Volume("dom") = {1};
- Select the "Mesh" tool in the drop-down menu pf gmsh and mesh in 3D your geometry. It takes a few minutes, to reduce this time, increase the size of your cells by changing the values of lc.

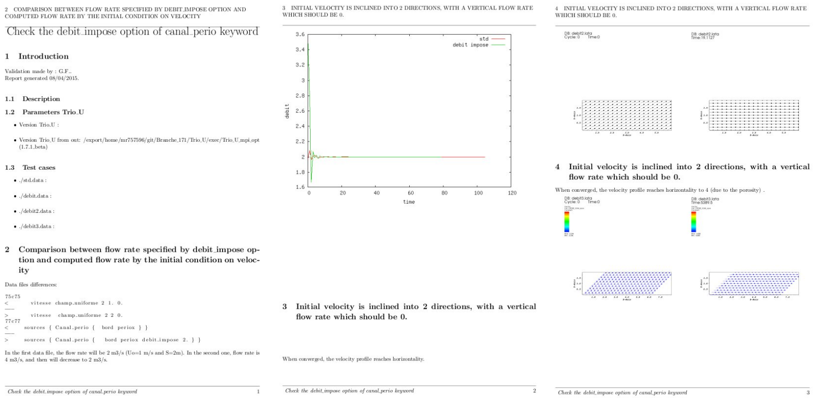
Gmsh to create a 3D VEF mesh

- You can use the "Optimize 3D (Netgen)" algorithm to optimize your mesh.
- Export your mesh to a MED file. Run gmsh again on this exported MED file to check everything is defined:
  gmsh file.med &
- Now, read the mesh by TRUST. You can, for example, take the first exercise and create a Obstacle_VEF.data file (from a copy of Obstacle_VDF.data) to read the MED file and run the simulation on the unstructured mesh.

1. Initialization
2. Obstacle VDF
   - Obstacle: Sequentiel calculation
   - Obstacle: Parallel calculation
   - Obstacle: Parallel calculation on a cluster
3. Heat exchange VDF/VEF
4. Low mach number flow
5. Periodic 3D channel flow
6. Constituents and turbulent flow
7. 3D turbulent flow in a curved pipe
8. Turbulent flow on a 3D step
9. Tank filling (2D-single phase flow)
10. Tank filling (3D-two phases flow)
11. Meshing tools
    - Salomé to create a 3D VEF mesh
      - Cylinder
      - Revolution
      - T-shape
    - Gmsh
      - 2D VEF mesh
      - 3D VEF mesh
12. **Validation form**
13. Index

Validation form

Example of Validation form:



2 COMPARISON BETWEEN FLOW RATE SPECIFIED BY DEBIT_IMPOSE OPTION AND COMPUTED FLOW RATE BY THE INITIAL CONDITION ON VELOCITY

Check the debit_impose option of canal_perio keyword

1 Introduction

Validation made by : G.F..
Report generated 08/04/2015.

1.1 Description

1.2 Parameters Trio_U

- Version Trio_U :
- Version Trio_U from out: /export/home/mr757596/git/Branche_171/Trio_U/exec/Trio_U_mpi_opt (1.7.1_beta)

1.3 Test cases

- ./std.data :
- ./debit.data :
- ./debit2.data :
- ./debit3.data :

2 Comparison between flow rate specified by debit_impose option and computed flow rate by the initial condition on velocity

Data files differences:

```
75c75
<        vitesse champ_uniforme 2 1. 0.
---
>        vitesse  champ_uniforme 2 2 0.
77c77
<     sources { Canal_perio {  bord  periox } }
---
>     sources { Canal_perio {   bord  periox  debit_impose  2. } }
```

In the first data file, the flow rate will be 2 m3/s (Uo=1 m/s and S=2m). In the second one, flow rate is 4 m3/s, and then will decrease to 2 m3/s.

Check the debit_impose option of canal_perio keyword 1

3 INITIAL VELOCITY IS INCLINED INTO 2 DIRECTIONS, WITH A VERTICAL FLOW RATE WHICH SHOULD BE 0.

3 Initial velocity is inclined into 2 directions, with a vertical flow rate which should be 0.

When converged, the velocity profile reaches horizontality.

Check the debit_impose option of canal_perio keyword 2

4 INITIAL VELOCITY IS INCLINED INTO 2 DIRECTIONS, WITH A VERTICAL FLOW RATE WHICH SHOULD BE 0.

4 Initial velocity is inclined into 2 directions, with a vertical flow rate which should be 0.

When converged, the velocity profile reaches horizontality to 4 (due to the porosity) .

Check the debit_impose option of canal_perio keyword 3

Validation form

- First copy the validation form named Source_canal_perio:

```
mkdir -p ~/test/yourname/validation
cd ~/test/yourname/validation
VERIF=$TRUST_ROOT/Validation/Rapports_automatiques/Verification
cp -r $VERIF/Verification_codage/Source_canal_perio  .
cd Source_canal_perio
```

- Build the report:

```
Run_fiche -xpdf
```

or

```
Run_fiche
evince build/rapport.pdf &
```

Validation form

- Now, we are going to change the validation form (Examples are given in pages 10 & 11 of the HowTo_Validation.pdf note):
  nedit src/Canal.prm &
  - Add the mesh plot in the report. For this, at end of .prm file, introduce a new block with ”visu” keyword:

    ```
    Chapter {
       Title ”Additional information”
       Visu {
          Title ”Mesh visualization”
          Mesh lata_file_name domain_name [color]
       }
    }
    ```
  - You can help with:
    - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml#paramVisu
    - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/exemples/visu.prm
  - Save the .prm file and re-build the report without running the calculations:
    Run_fiche -not_run

Validation form

- Add the evolution of residus in the report in log scale (Cf. .dt_ev file). For this, introduce a new block with "Figure" and "Curve" keywords:
  - Complete the chapter "Additionnal information" with a new block "Figure":

```
Figure {
   Title "Residus evolution"
   LabelX "Time (s)"
   LabelY "Residu"
   LogX
   LogY
   Include_description_Curves 0
   Curve {
     legend data_file_name.data
     file data_file_name.dt_ev
     columns  $column_number_for_x   $column_number_for_y
     style lines
   }
   ...
}
```

  - You can help with:
    - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml#paramFigure
    - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml#paramCourbe
    - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/exemples/impl.prm

Validation form

- Visualize the pressure field at last time: complete the chapter "Additionnal information" with a new block "Visu" using "PseudoColor" keywords. The name of field and localisation must be uppercase letter:

```
Visu {
   Title "Visualization of pressure field at last time"
   Description "Pressure field..."
   Pseudocolor data_file_name.lata domain_name FIELD POSITION
   Cycles cycle_numbers
   Width size_in_cm
}
```

  NB: Cycle '-1' correspond to the last time.

  NB: You can use the "magnitude" keyword to visualize the velocity field without using arrows.

- Save and close the .prm file and build the report with the options: (Options are given in page 7 of the HowTo_Validation.pdf note)

  Run_fiche -xpdf -not_run

Validation form

- Now, we are going to extract the number of cells and the last time in three of the .err files and write the results in .dat files via a post_run script.
    - Create a ”post_run” file in the src directory containing:

      ```
      nb1=` grep ”Total number of elements” std.err | awk '{print $NF}' `
      nb2=` grep ”Total number of elements” debit.err | awk '{print $NF}' `
      nb3=` grep ”Total number of elements” debit2.err | awk '{print $NF}' `
      echo $nb1 $nb2 $nb3 > nbcells.dat
      tp1=` grep ”Backup of the field” std.err | awk '{print $NF}' | head -n 1 `
      tp2=` grep ”Backup of the field” debit.err | awk '{print $NF}' | head -n 1 `
      tp3=` grep ”Backup of the field” debit2.err | awk '{print $NF}' | head -n 1 `
      echo $tp1 $tp2 $tp3 > lastime.dat
      ```
    - Add a table to display the results of .dat files: complete the chapter ”Additionnal information” by introducing a new block with ”table” and ”line” keywords.

      You can help with:
        - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml#paramTableau
        - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml#paramLigne
        - $TRUST_ROOT/Validation/Outils/Genere_courbe/doc/exemples/tableau.prm

Validation form

- The table block will looks like:

```
Table {
   Title ”Number of cells and last time results”
   Nb_columns number_of_colomns_without_the_first_one
   Label label_of_the_first_column | label_of_the_second_column ...
   Line {
      legend title_of_the_first_line
      file name_of_the_file.dat
   }
   ...
}
```

- Save the .prm file in src directory and build the report with a new option: **Run_fiche -xpdf -post_run**

Validation form

- Now we are going to modify the "prepare" file in "src" directory in order to add a fourth test case: "debit4"
  - "debit4" correspond to "std" test case with zero initial velocity and imposed flow rate to $2m^3/s$ on "periox" boundary.
  - Add a line in the "prepare" file using the "sed" command to create the new data file like the other ones:
    nedit src/prepare &
- We are going to update the validation form.
  - Modify the "Canal.prm" file in "src" directory to take into account the new test case debit4. Add in the **Parameters** block :
    TestCase . Debit4.data
  - Build the report by running the 4 test case concurrently and not sequentially:
    Run_fiche -parallel_run -xpdf
  - You can add the results of this test case to your "visu" and "table".

Index I

Index II

Index III

Index IV

End