Trio_U tutorial v1.6.8

Table of Contents

Initialization

a) First, initialize the Trio_U environment

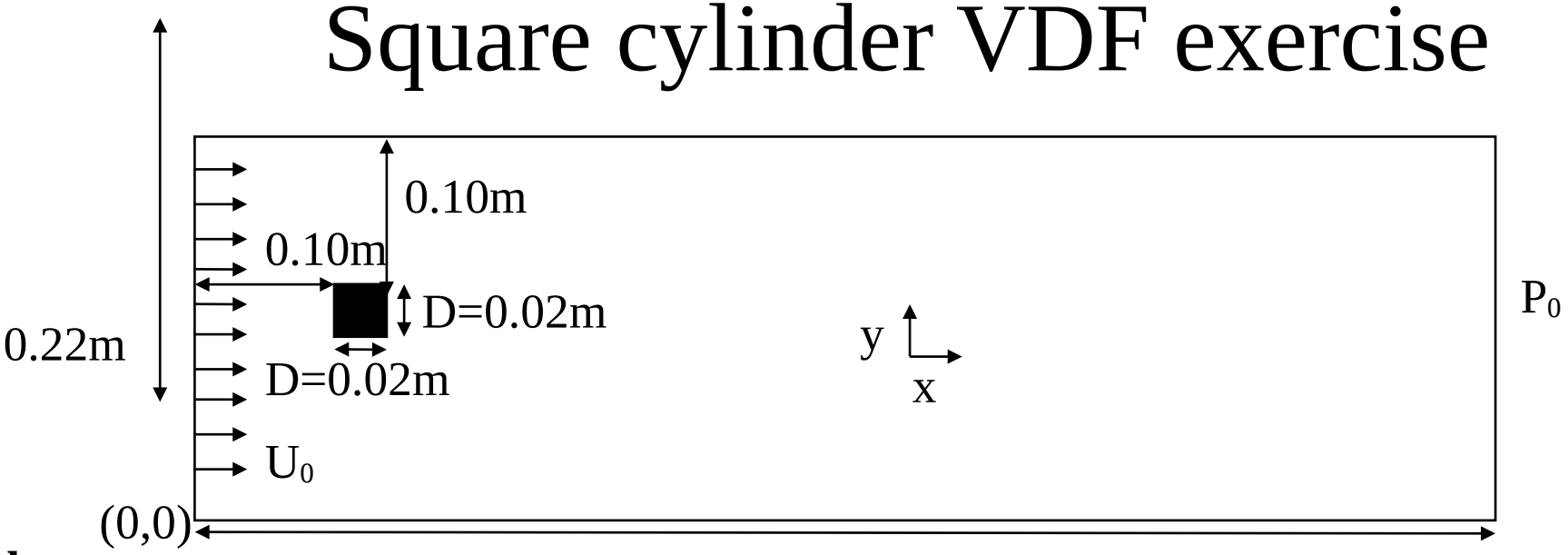
On CEA Saclay PCs (with login fsalomeN (N=1 to 6), password fsal_04), a Trio_U version is installed sometimes under /export/home/triou of the PC, so run:

source /export/home/triou/Version_test_..../Trio_U/bin/Init_Trio_U

b) Second, several editors (vi, emacs, nedit) are configured since the 1.6.8 version to highlight Trio_U keywords in the data files. If you prefer using nedit, please do the following:

Run **nedit**, and select Preferences->Save Defaults, then re-initialize the Trio_U environment (source $TRIO_U_ROOT/bin/Init_Trio_U), the message “nedit.rc updated” should appear).

Square cylinder VDF exercise



Fluid :

$\mu=1.10^{-5}$ kg.m^{-1}.s^{-1} $\rho=1$ kg.m^{-3} Re=1000

Boundary conditions:

Inlet with fixed velocity: $U_0=0.5$ m.s^{-1}

Outlet with a fixed pressure: $P_0=0$

Square cylinder: Wall

Upper and Lower walls: Symmetry

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 Trio_U database to your study with the command:

triou -copy Obstacle
cd Obstacle

First, we are going to configure the **Nedit** editor to recognize the Trio_U keywords in a data file. So run nedit, and select Preferences->Save Defaults, then re-initialize the Trio_U environment (source $TRIO_U_ROOT/bin/Init_Trio_U), the message “nedit.rc updated” should appear).

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

nedit *Obstacle.data*

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 (find the keyword for this field in the user's manual with: **triou –doc**) to the fields being post processed and change the writing period to 0.5s
- Add the keyword « **Format lata** » inside the block, just before the keyword **Champs** in order to use the post processing tool VisIt during and/or after the calculation
- You have access to useful resources in the *index.html* file under $TRIO_U_ROOT directory with your favourite browser (eg: firefox). Take few minutes to find test case examples containing a keyword thanks to the Keywords link:

firefox $TRIO_U_ROOT/index.html

Run and monitor the calculation :

triou –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, then 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

Visualize the equation residuals (plot 2)

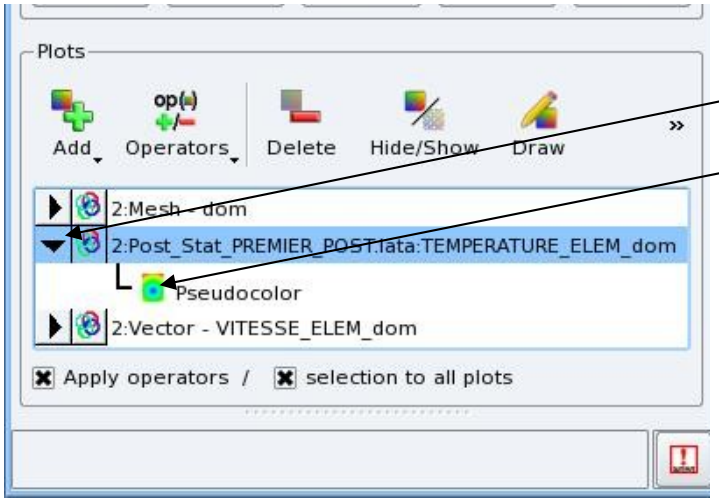
Visualize the drag exerted by the fluid on the square Cylinder. Select the plot 1 (Surface fluxes), then the plot 5 (Pressure drag + Friction drag) and select the Square boundary. Each drag is plotted together with the total drag also called viscous drag.

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 and visualize the mesh with “Plots->Add->Mesh->dom->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 (Double click on the last Time, then “Plots->Pseudo Color->PRESSION_SOM_dom->Draw”)
-Suppress or hide the mesh (Select Mesh then Delete or Hide/Show)
-Visualize the velocity field (Plots->Vector->VITESSE_SOM_dom->Draw)



You can change each plot attributes:
- click once onto the small arrow then
- double click on the item (here, Pseudocolor)

For example for the velocity, 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.

-Print your visualization (File->Save window) : a PNG file is created into your study directory.
-Add a second screen with “Windows->Layouts->1*2”

-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) : 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)
-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** and **tmax** values (pick in the .err file the last backup time of the previous calculation), and add in the problem definition something like:

reprise binaire *Obstacle_pb.sauv*

Restart the calculation again with:

triou –monitor Obstacle

Parallel calculation exercise

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
cp  Obstacle.data  DEC_Obstacle.data
cp  Obstacle.data  PAR_Obstacle.data
```

Edit the first file (*DEC_Obstacle.data*) to create the partition of the mesh. In this file, uncomment the block around the **Decouper** keyword. Here, the partitioning tool **Tranche** is used. We cut in 2 bands according to X axis and 1 band according to Y axis. The overlapping width between two parts of the partition (**Larg_joint**) 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 **Nom_Zones** is useful to define the name of the files containing the partitioned mesh and to write these files.

Run the data file:

triou DEC_Obstacle

Check the partitioned mesh files *DOM_0000.Zones* and *DOM_0001.Zones* are generated in your directory:

ls *.Zones

Now, edit the file *PAR_Obstacle.data* and comment the read of the mesh (**lire_fichier**) and uncomment the **Scatter** keyword which will read the partitioned mesh. Visualize it with VisIt:

triou –mesh PAR_Obstacle

Now, run a parallel calculation with Trio_U:

triou 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->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, you can run the command line:

triou –probes

Tip: The existing tool **make_PAR.data** is useful on the data files which have the marks MAILLAGE, DECOUPAGE and LECTURE. If you run the following command:

cp Obstacle.data exemple.data
make_PAR.data exemple 3

ls *.data *.Zones

It will create a *DEC_exemple.data* file and will immediately run this file to create a partition with 3 sub zones. The command **make_PAR.data** will also create *PAR_exemple.data* file and then you will run the calculation by the usual command:

triou PAR_exemple 3

Parallel calculation on a cluster

1) Log onto the cluster eris and initialize the Trio_U version

ssh –X eris.intra.cea.fr
source /workdir/triou/Version_test_eris/Trio_U/bin/Init_Trio_U

2) Copy the study Obstacle:

triou -copy Obstacle
cd Obstacle

3) Create automatically a 2 partitioned mesh and a parallel datafile with:

make_PAR.data Obstacle

4) Submit the parallel calculation on 2 cores:

triou PAR_Obstacle 2

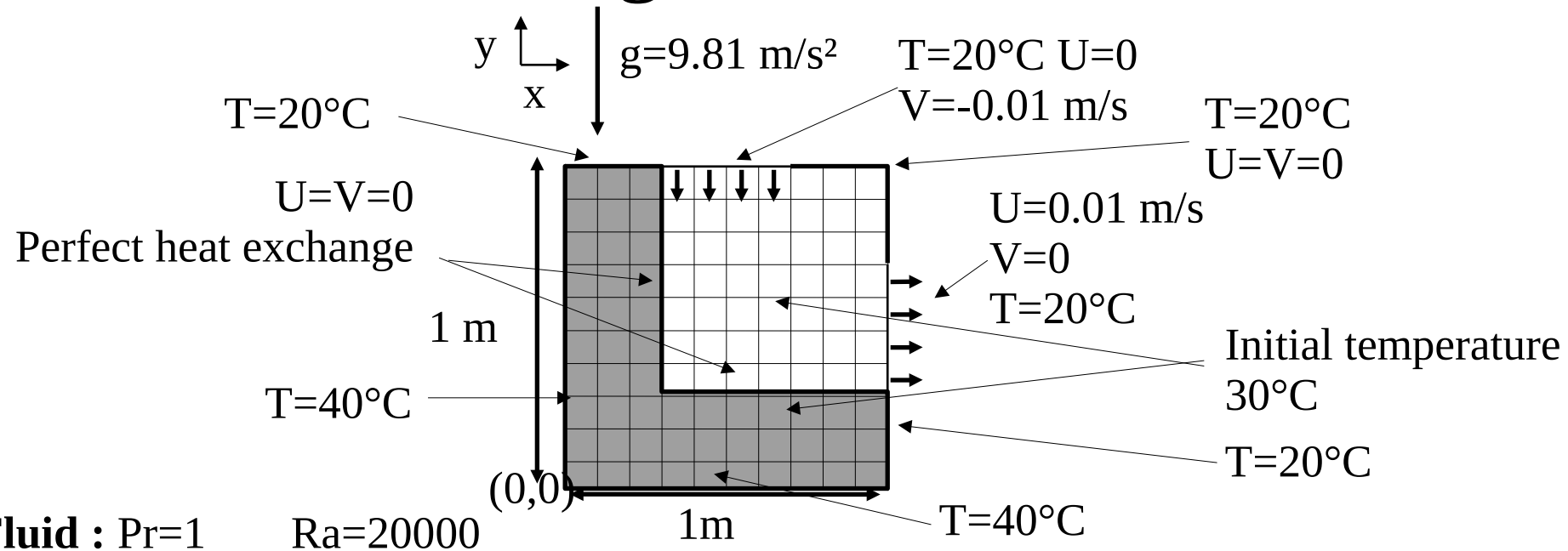
5) Create a submission file:

triou –create_sub_file PAR_Obstacle 2

6) Submit the job with **qsub** and check the state of the job with **qstat -u ***

7) You could run VisIt from your PC, and access/visualize the result file on eris cluster

Heat exchange VDF/VEF exercise



Fluid : Pr=1 Ra=20000

μ=1,0.10^{-3} kg.m^{-1}.s^{-1} ρ=1,0 kg.m^{-3}

λ=1,0 W.m^{-1}.K^{-1} C_p=1000 J.kg^{-1}.K^{-1} β=1,0.10^{-4} K^{-1} T_{ref}=30°C

Solid : ρ=1000 kg.m^{-3} λ=250 W.m^{-1}.K^{-1} C_p=100 J.kg^{-1}.K^{-1}

-Create a new study **Coupling_VDF** by copying the **docond** study:

```
cd ~/test/your_name
triou –copy docond
mv  docond  Coupling_VDF
cd Coupling_VDF
```

-Check the fluid and solid characteristics inside the *docond.data* file.

-It is a 2 domains calculation with a 10*10 mesh (dx=dy=0.1m) created with 3 blocks. Modify the data file to have the 2 domains on a mesh of 40*40 cells (dx=dy=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

-Add ‘format lata” into the two problems definition and run the calculation and check the evolution. Then post process (temperature) 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:

triou –mesh docond

-Run the calculation with:

triou –monitor docond

-Follow the residuals (plot number 2) and a probe evolution (for example temperature), to see when convergence is reached.

-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). Run the calculation.

-Use a full implicit scheme now, by substituting **Schema_Euler_Explicite** by **Schema_Euler_implicite** and adding the **Implicite** solver (keywords **Solveur Implicite** { **solveur gmres** { ... } }). Have a look at the User's manual for the **gmres** options. Define accordingly to the advices given into the User's manual, a value for **facsec**, **facsec_max**. Run the calculation.

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/your_name
triou –copy TP_Temp_QC_VEF
cd TP_Temp_QC_VEF

Open the Trio_U user's manual with (it will be useful to search for keywords in this exercice):

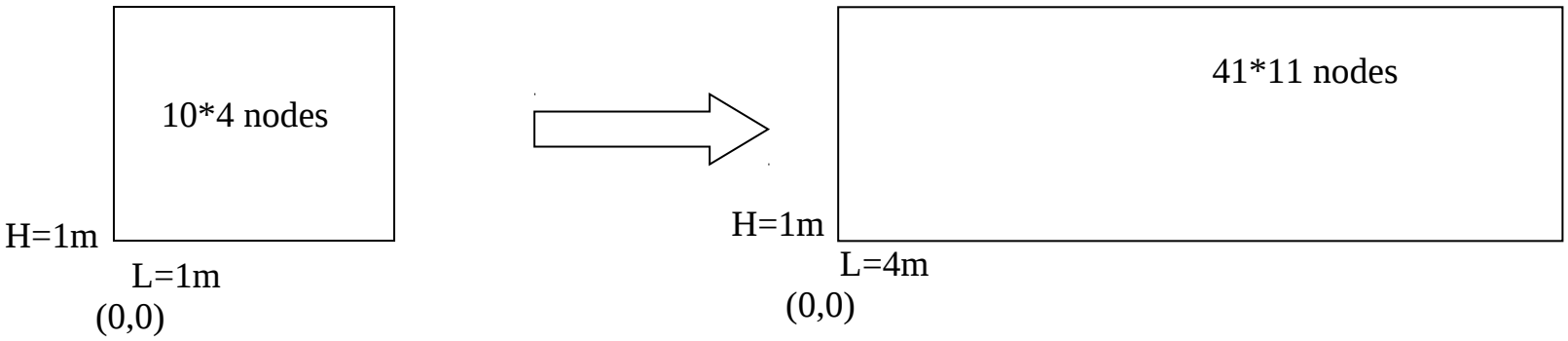
triou -doc &

Edit the data file with your favourite editor (**nedit** is recommended because it is configured to recognize Trio_U syntax):

nedit TP_Temp_QC_VEF.data &

Modify the data file in order to:

-Change the geometry and the mesh:



-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 1 second
-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~0.001s so $|dT|<0.01$)

-Add the keyword **impr** into the pressure solver to print its convergence
-Run the simulation with the Trio_U command **triou** :

triou -monitor TP_Temp_QC_VEF 2>TP_Temp_QC_VEF.out

-Check the convergence information by entering 0 (convergence monitoring in the menu). Visualize the 3 plots :

-Iterations number for the Conjugate Gradient solver at each time step
-Error evolution for the Conjugate Gradient resolution at each time step
-Time step evolution

-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 the graphical tool **VisIt**:

visit –o TP_Temp_QC_VEF.lata

-See the mesh (“Plots->Mesh->dom->Draw”)

-Visualize the temperature field (Open the last Time, then “Plots->Pseudo Color->TEMPERATURE_SOM_dom->Draw”)

-Suppress or hide the mesh (Select Mesh then Delete or Hide/Show)

-Visualize the velocity field (Plots->Vector->VITESSE_SOM_dom->Draw)

-Zoom with the right button of the mouse (Mode->Zoom). Then to de-zoom with right button (Reset view).

-Print your visualization (File->Set Save options->File type->Select a type->Save) : a file is created into your study directory.

-Add a second screen with “Window->Layout->1*2”

-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. 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). Run the calculation with this time scheme. 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 &

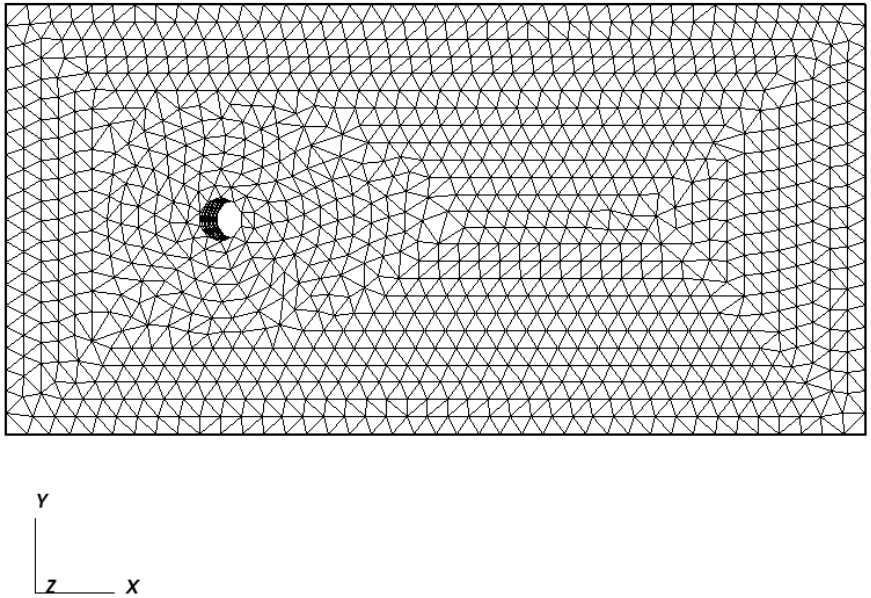
-If everything is OK, try to fine tune the speed convergence of the implicit solver with the value of **seuil_convergence_solver** keyword (look inside the TP_Temp_QC_VEF.out file and check that the number of iterations for GMRES is between 3 and 5, it is enough to quickly converge).

-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:

triou TP_Temp_QC_VEF.data 1> TP_Temp_QC_VEF.out

Periodic 3D channel flow



Fluid : Re=200, initial velocity V_0=1m/s, periodic boundary condition on X and Z directions

-Copy the study named **P1toP1Bulle**. It simulates a 3D incompressible laminar flow (Re=200) with periodic boundary according Z direction only.

-Use **RegroupeBord** keyword to merge *Entree* and *Sortie* boundaries into a single one named *periox*. 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**) and 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.

-Now, the calculation domain is a rotating channel according to Z direction with a constant velocity Ω=1 rad/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.

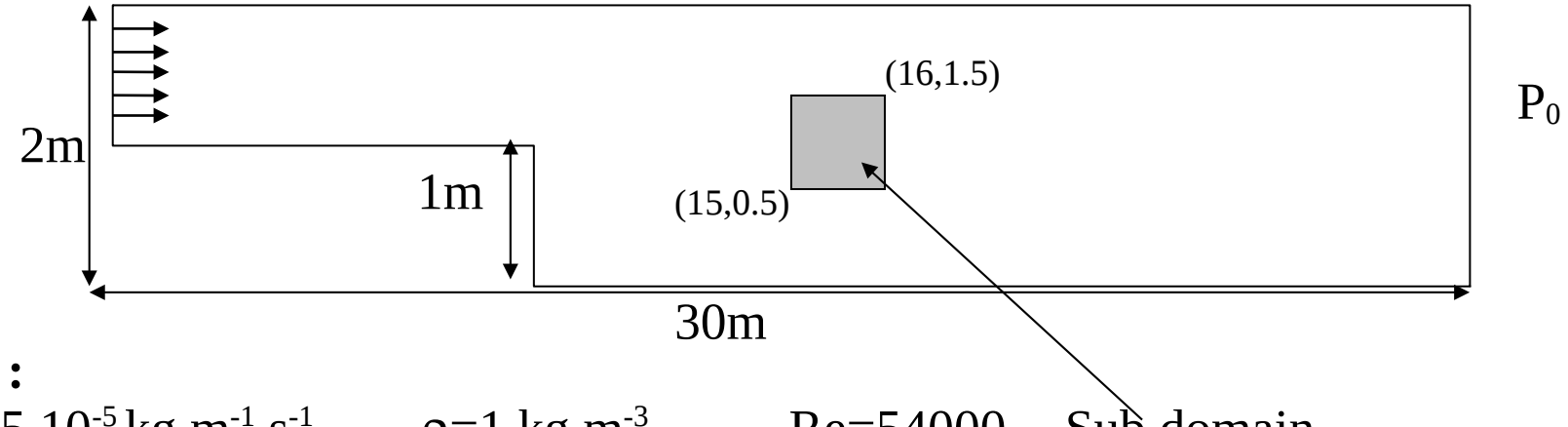
-To improve the calculation speed on this mesh or a bigger one (you can create a uniformly refined mesh using for instance the keyword **Raffiner_Anisotrope**), you could try:

-Use semi implicit scheme (add the keyword **diffusion_implicite** into **Schema_Euler_explicite** scheme)

-Run a calculation on the same mesh with a coarse discretization (**P1**) with less pressure unknowns (on this mesh, it runs 3 times faster than P1Bulle but with less accuracy: 8452 unknowns compare to 49221 unknowns) then stop the calculation and restart the calculation with **VEFPreP1B** discretization by reading the velocity field with **Champ_fonc_reprise** keyword in the initial conditions for the velocity. This will be useful to reach the quasi-stationary state faster.

-Use implicit scheme (change the scheme to **Schema_Euler_implicite** scheme and use the **Implicite** keyword) only if you are looking for the stationary state only.

Constituents and turbulent flow exercise



Fluid :

$\mu=1.85.10^{-5}$ kg.m^{-1}.s^{-1} $\rho=1$ kg.m^{-3} Re=54000 Sub domain

Boundary conditions :

Inlet with fixed velocity: $U_0=1$ m.s^{-1}, $C_1=1$ $C_2=2$ $C_3=3$ (dimensionless values)

Outlet with a fixed pressure: $P_0=0$, $C_1=0$ $C_2=0$ $C_3=0$

Upper and lower walls: No slip wall (U=0) and non-porous wall ($\partial C_i/\partial n=0$)

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

-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 (α=1m/s) and associate the constituents to the problem.

-Define the concentrations 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.

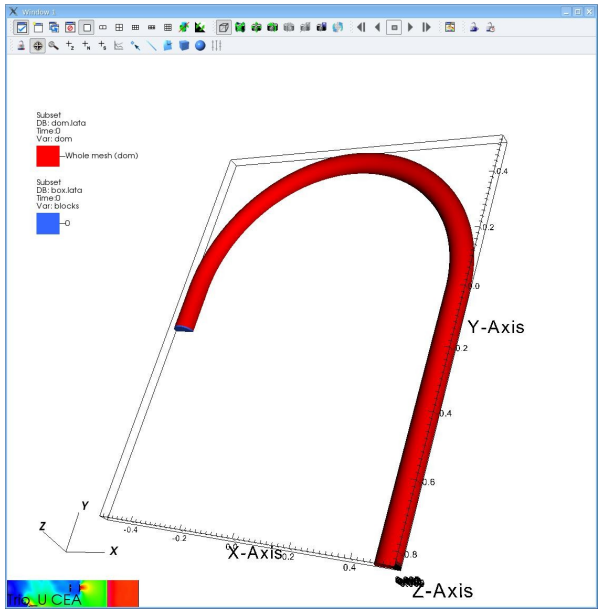
-Use the Schmidt model to close the turbulence modelization

-Define a sub domain (in grey on the previous picture) with the keyword **Sous_Zone** and 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**.

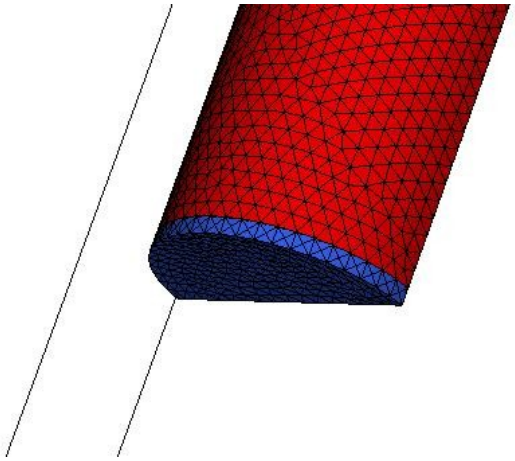
-Add the keyword in the **Champs** option of the post processing definition to write the 3 concentrations into the *.lata* results file

-Run the calculation and check the results.

3D turbulent flow in a curved pipe



Curved pipe



Detail of the box near the inlet of the curved pipe

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

Create a new directory and copy some data:

mkdir –p ~/test/yourname/PeriodicBox

cd ~/test/yourname/PeriodicBox
cp $TRIO_U_ROOT/Validation/Rapports_automatiques/Validant/pas_fini/PeriodicBox/src/* .

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 and the *PeriodicBox* and *DomainFlow* data files according you wish run a RANS or LES simulation.

Then build the meshes:

./prepare
triou BuildMeshes

You can visualize the partitioned meshes with (MODEL=RANS or LES):

triou –mesh PeriodicFlowMODEL

Then, run a 2 cores parallel calculation to initialize the turbulent flow in the box:

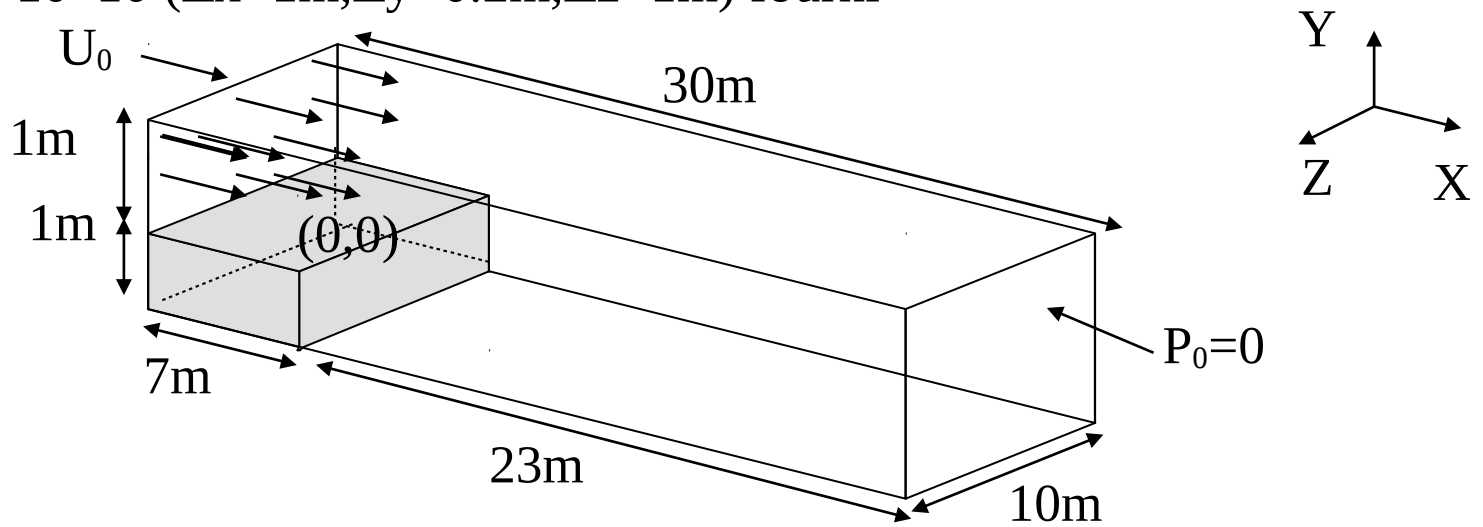
triou PeriodicBoxMODEL 2

And, once finished (the RANS model converges in few minutes, the LES model takes much longer, so you will stop it after few minutes), run a 6 cores parallel calculation of the domain (it will stop by default after 10 times steps):

triou DomainFlowMODEL 6

Turbulent flow on a 3D step exercise

Meshing : 30*10*10 (Δx=1m,Δy=0.2m,Δz=1m) fourni



Fluid :

$\mu=2{,}5.10^{-5}$ kg.m^{-1}.s^{-1} $\rho=1$ kg.m^{-3}

Boundaries conditions: (Re=40000)

Inlet: $U_0=1$ m.s^{-1} *Outlet*: $P_0=0$

-Copy the study named **Marche3D**

triou –copy Marche3D

-Edit the data file and:

- Modify the fluid characteristics to perform a calculation at Re=40000
- Define a turbulent hydraulic problem (see user manual)
- Select the sub-grid Smagorinsky turbulence model with standard wall law
- Select the Quick convection scheme
- Post-process of velocity, pressure, vorticity, ν_t at the nodes and elements

-Run the calculation. Post-process of the main calculated fields.

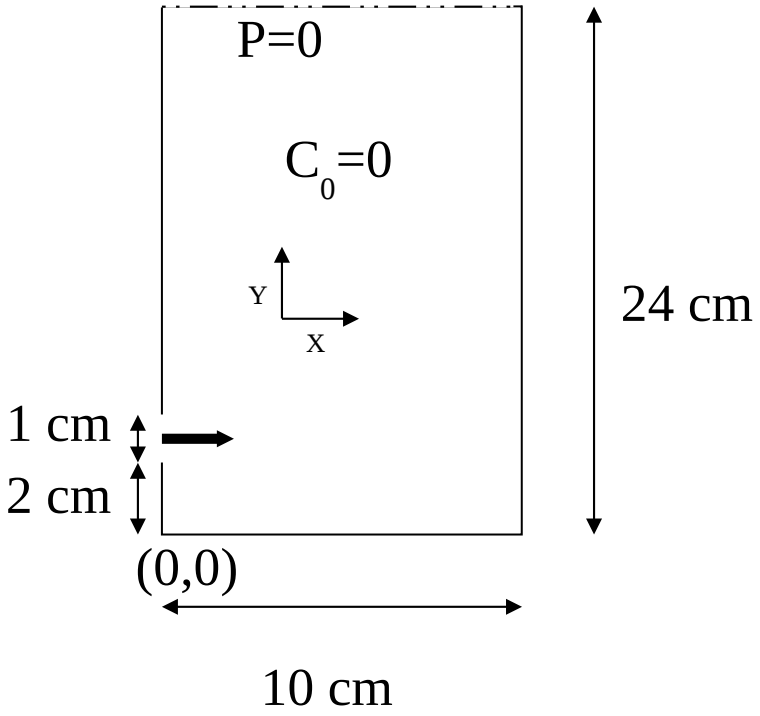
-Replace the sub-grid model by the standard k_eps model.

Tank filling exercise (2D-single phase flow)

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

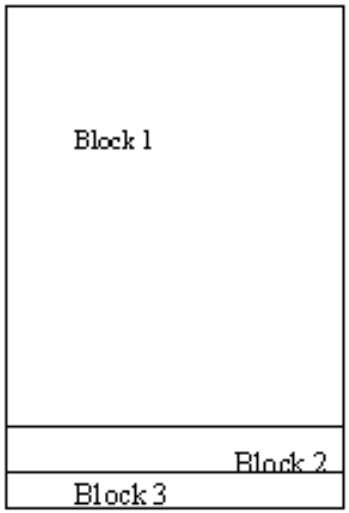
Boundary condition:
Inlet: Velocity $V_x=V(t)$ $V_y=0$ with
$V(t)=1-(y-0.025/0.005)^2$ (t<=0.5s), V(t)=0 (t>0.5s)
Inlet: Concentration C=1 (t<=0.5s), C=0 (t>0.5s)
Outlet: Pressure P=0
Wall: Velocity $V_x=0$, $V_y=0$

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



-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 create the corresponding mesh with 3 blocks (start with dx=dy=0.2cm which gives a total node number N_x=51 and N_y=121). You could also use **facteurs** and **symx**, **symy** keywords to define a refined mesh near the walls. Define the 3 boundaries (inlet, outlet, and wall).



-Check the mesh

-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 Trio_U calculates itself its time step.

-Change the values for the fluid and the constituent. The **beta_c0** keyword may be useful in order to have a Boussinesq coupling between momentum and concentration equations ($\beta_{C0}g(C-C_0)$ source term added in Navier Stokes equation). Change values for the gravity.

-Change the initial and boundary conditions for Navier Stokes equation. You will use the **Champ_Front_Fonc_txyz** keyword for the velocity. To write the time dependant condition for velocity you will 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)

-Change the initial and boundary conditions for the constituent equation. You will use also **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.

The keyword for impermeable boundary condition for concentration is **paroi**.

-Check you have high order schemes 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:

Diffusion { **negligeable** }

-Add a concentration probe near the inlet (e.g.: at (0,0.025)) and 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). Change the period to 0.1s in the post processing definition (keyword **dt_post**).

-Run the study and follow the time evolution with the probes.

-Check the flow rate in inlet boundary in the *diagonale.out* file. You should find a value near $6.66.10^{-3}$ $m^2.s^{-1}$.

-Use VisIt to post process the results at t=0.2 (Time_0002), t=0.4s (Time_0004) and t=0.7s (Time_0007). VisIt tool will be the same work, but VisIt has some interesting feature for this study. It can give concentration histogram to check the numerical diffusion in the concentration equation. 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.

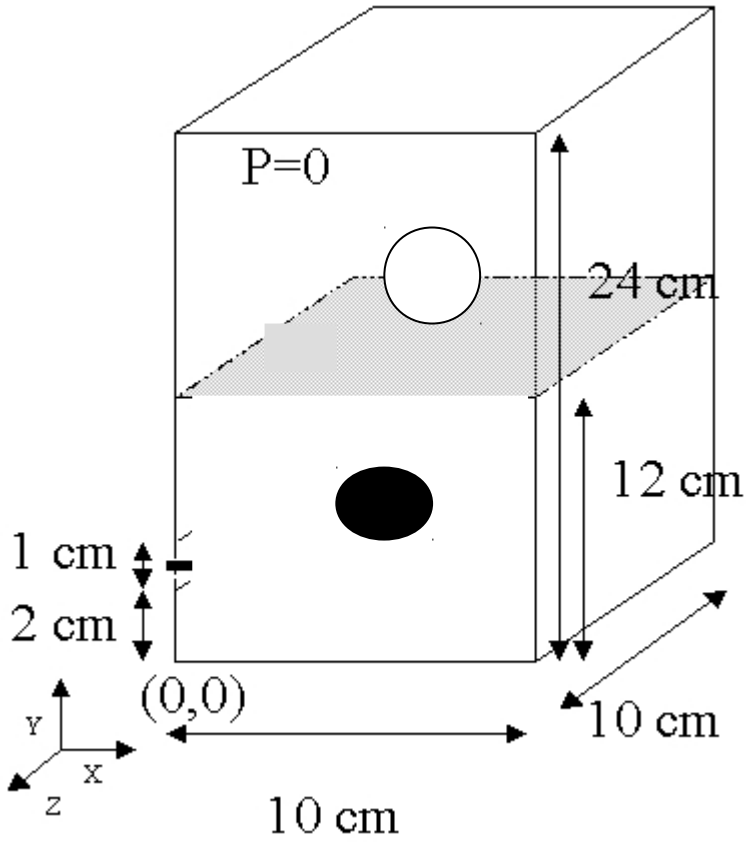
-Copy *diagonale.data* to *diagonale_VEF.data* and in this new file, change the discretization (**VEFPreP1B** instead of **VDF**), triangulate your mesh (**trianguler** keyword), and use **muscl** instead of **quick** scheme. You can switch **GCP** solver by **Cholesky** solver (direct method which may 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 and compare the results between **VDF/quick** and **VEFPreP1B/muscl**.

Tank filling exercise (3D-two phases flow)

Liquid:
ρ= 1000 kg.m^{-3}, μ=2,82.10^{-4} kg.m^{-1}.s^{-1}
σ=0.05 N.m^{-1} D=10^{-6} m^2.s^{-1}
Gas:
ρ=100 kg.m^{-3}, μ=2,82.10^{-4} kg.m^{-1}.s^{-1}
Boundary conditions:
Up : Free outlet
Down: V(x,y,z)=(0,0, 10^{-4} m.s^{-1})
Wall : V=0
Initial conditions:
V=0, C=exp(-((x-0.02)^2+(y-0.02)^2+(z-0.03)^2)/0.03^2)
NB : Sizes of the tank on the picture are false



-Copy the study named **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 the 2D Discontinuous Front Tracking method has not been intensively tested yet.

-Notice the name of the problem : **Probleme_FT_disc_gen**. This refers to the Discontinous Front Tracking method.

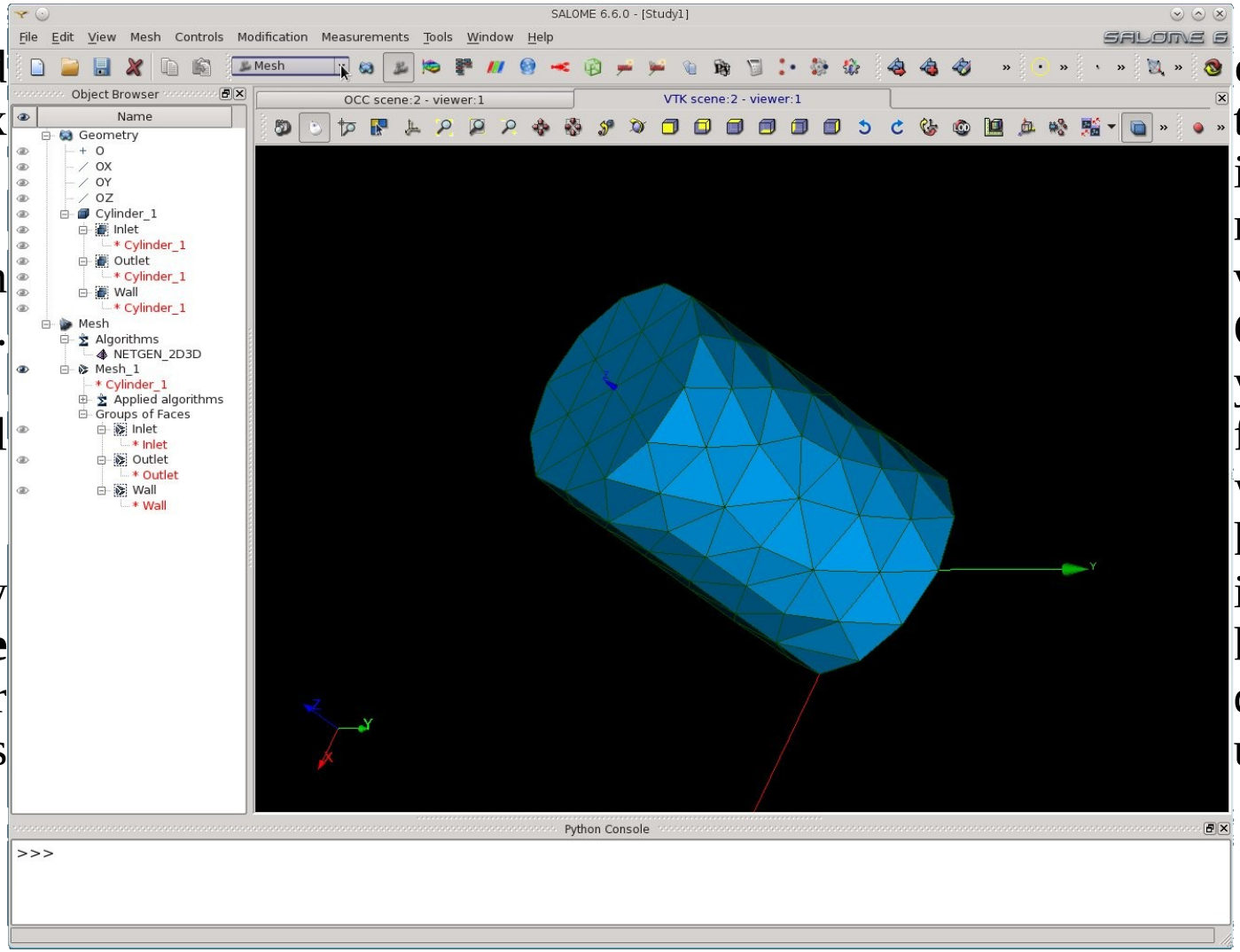
-Notice the keyword **modele_turbulence**. Navier Stokes equation of the Discontinuous Front Tracking problem needs the read of this keyword even if the flow is laminar. In this case, use the **nul** keyword just after **modele_turbulence**. Else, specify the turbulence model chosen.

-Increase the height of the tank (0.06 to 0.12) and add a second drop above the first one, at z=0.08 (keywords **ajout_phase0**/**ajout_phase1** could be useful to add other interfaces).

-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” 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** 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.

-The second one 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.

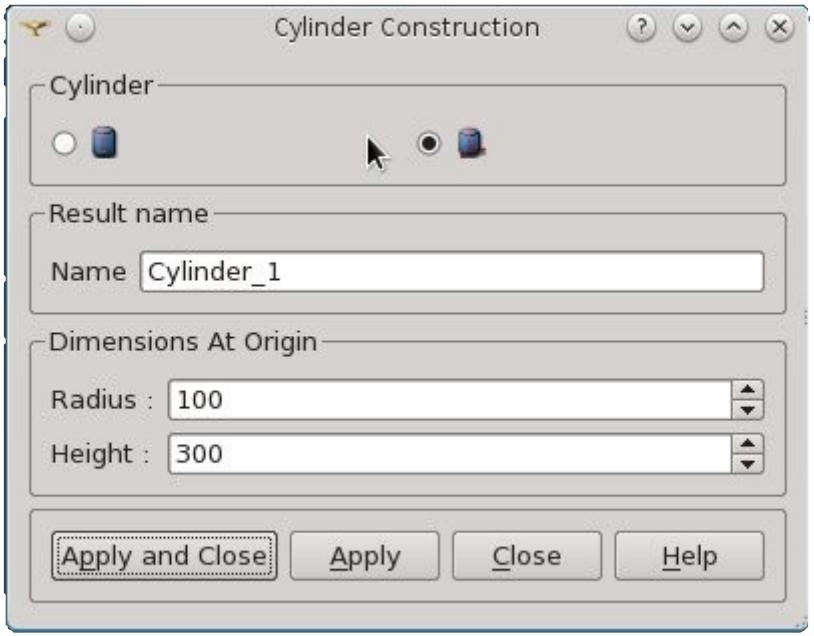
Salomé to create a 3D VEF mesh

-Create a directory and run Salomé (we suppose it is installed):
$ mkdir -p ~/your_name/salome
$ cd ~/your_name/salome
$ runAppli *# Or runSalome, it depends of the Salomé installation/configuration #*

-Create a new study:
File->New

-Select the Geometry module into the SALOME menu (contains all the modules)
New Entity->Primitives->Cylinder

-Specify Radius and Height for the cylinder. Then Apply and Close.
-Rotate, zoom, move the geometry by switching to “Interaction style” : Mouse icon.



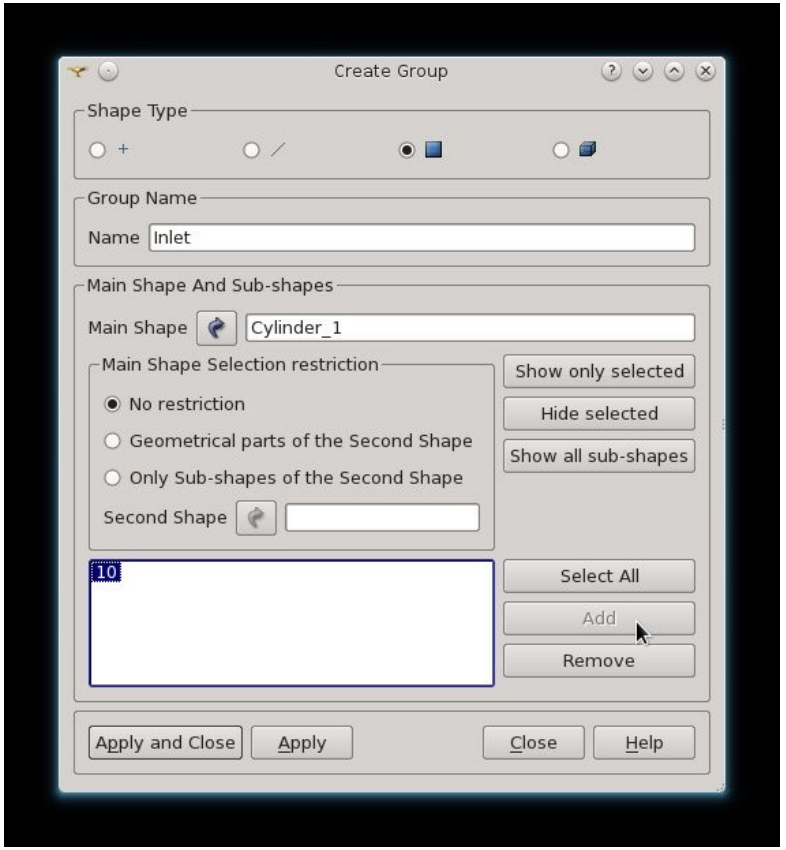
-Create groups for the geometry to define the top, the bottom and the lateral parts of the cylinder.
New Entity->Group->Create

-Select the good Shape Type (surface)

-Give a Group Name for each part:
For the top : “Inlet”
For the bottom : “Outlet”
For the lateral : “Wall”)

-Select the shape defining the part on the window then Add->Apply. **Warning**: Mouse icon should be unselected to select a shape.

-Close the window once the 3 groups has been created. Check they appear in the Object Browser.

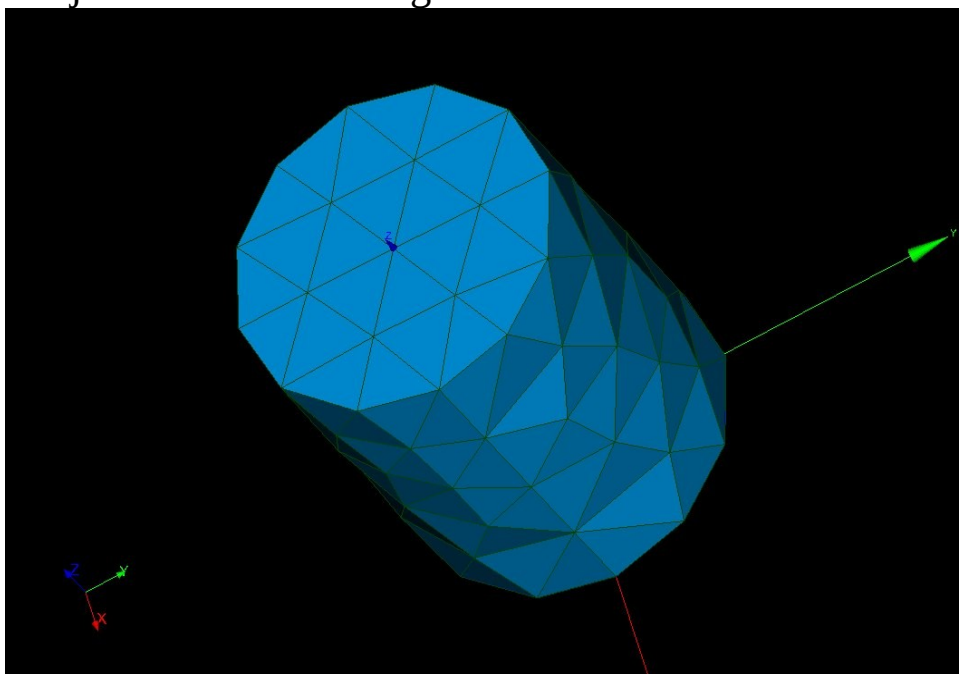


-Now, switch to the Mesh module.
-Select the Cylinder_1 in the Object Browser and Right Click->Show to visualize the geometry.

-Create a mesh with:
Mesh->Create Mesh

-Select the Geometry used for the mesh if not selected.
-Choose Netgen 1D-2D-3D algorithm.
-Apply and Close

-Select the object Mesh_1 in the Object Browser and Right Click->Compute (or Mesh->Compute)
-Select the Cylinder_1 in the Object Browser and Right Click->Hide

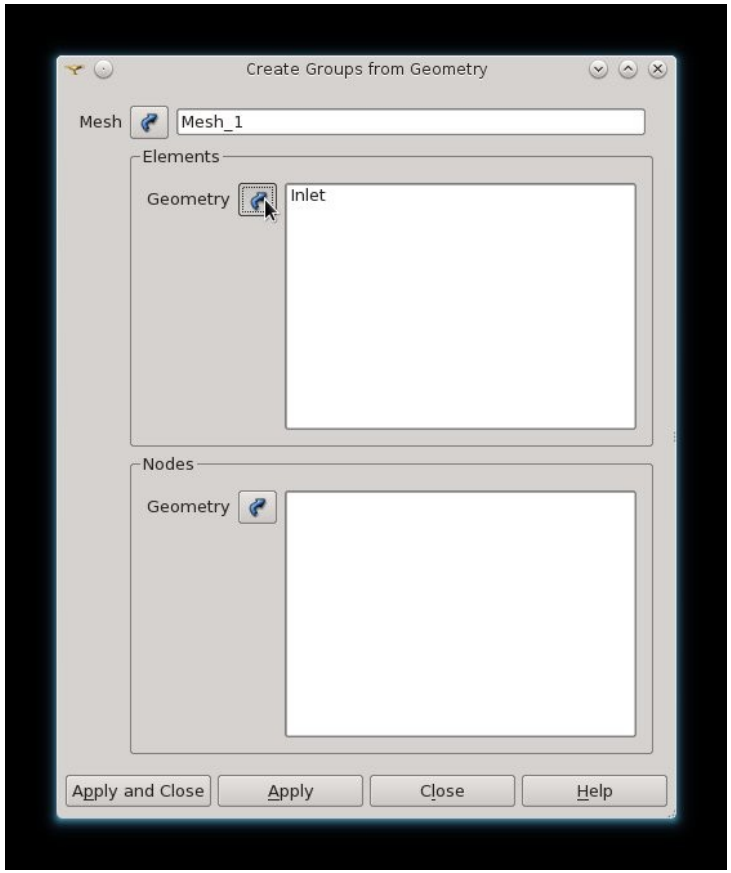


-Now, we are going to create groups for the mesh, by using the geometry groups:
Mesh->Create Groups from Geometry

-Click on the arrow of Elements->Geometry

-Select together the 3 groups (CTRL+left click) in the Object Browser, then press Apply and Close

-Check 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)

-Now build a data file named *dom.data* for Trio_U:

```
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:
$ triou 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 Trio_U is printed and detected during the discretization where all the faces of the mesh (in particular the boundary faces) are built.

STEP 2:
Goal: Improve the mesh for Trio_U near the wall by using viscous layers.

-Create a new mesh (named “Refined mesh” with):
Mesh->Create Mesh
-Select the Cylinder_1 geometry

-Select the Tetrahedron (Netgen) 3D algorithm

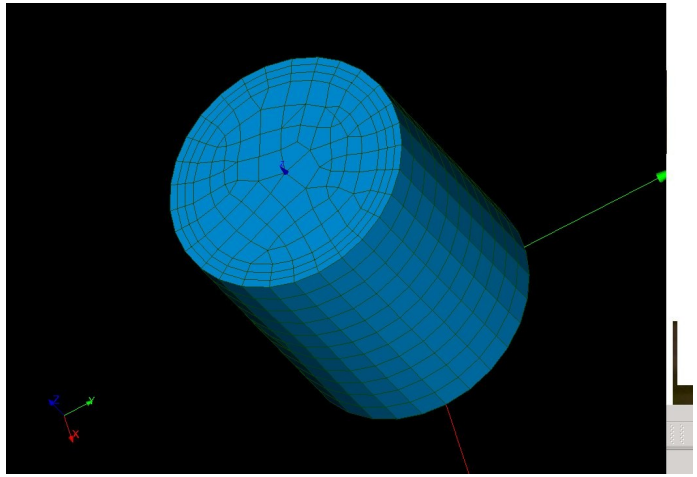
-Add Hypothesis->Viscous Layers with:

- Total thickness : 30
- Number of layers : 3
- Stretch factor : 1.1
- Add to Faces without layers the 2 geometry groups “Inlet” and “Outlet”
- Click OK

-Add 2D algorithm : Netgen 1D-2D
-Add Hypothesis-> Netgen 2D parameters

- Change Fineness from Moderate to Very Fine
- Select Allow Quadrangles
- Click OK

-Apply and Close the Create mesh window
-Select the Refined_Mesh object in the Object Browser and Right click->Compute
-You should have a refined mesh with a mix of tetra, hexa, pryramid, prism elements:



-As Trio_U accepted only tetras elements, you can quickly tetraedrize:
Select the Refined_Mesh in the Object Browser
Modification->Split into Tetrahedra
Don't change the parameters, and click Apply and Close

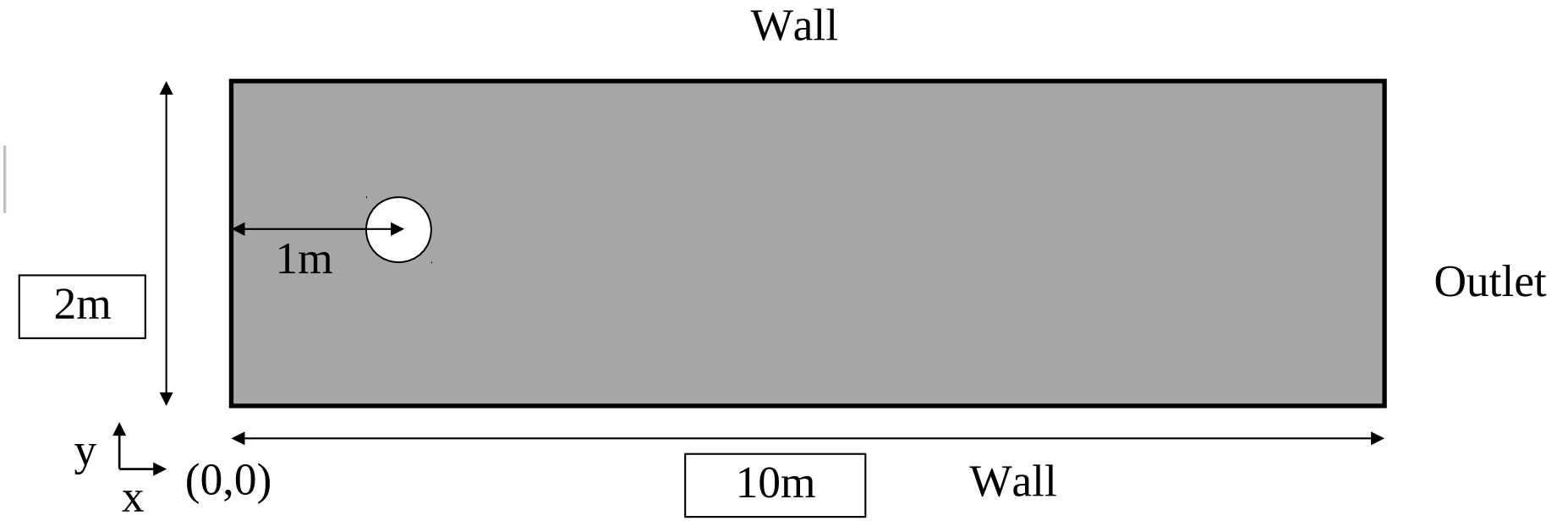
-As usual, define the mesh boundaries with Groups:
Select the Refined_Mesh, Right click->Create Groups from Geometry
Select the 3 Geometry groups “Inlet”, “Outlet”, “Wall”, then Apply and Close

-Export the mesh:
Select the Refined_mesh, Right click->Export->MED file
Save into a Refined_Mesh.med file

-Change or create a new data file for Trio_U to read and visualize your refined mesh.

NB: The solutions of the exercise (*mesh.hdf* file for step 1 and *prism.hdf* file for step 2) are located here:
$TRIO_U_ROOT/doc/Trio_U/exercices/salome

Gmsh to create a 2D VEF mesh



A) Create a directory and copy an example:

```
$ mkdir ~/your_name/gmsh
$ cd ~/your_name/gmsh
$ cp $TRIO_U_ROOT/Validation/Rapports_automatiques/Validant/pas_fini/Drag/src/shape.geo file.geo
$ nedit file.geo &
$ gmsh file.geo
```

-First configure **gmsh** to show points, lines, and surface numbers of the geometry. In menu Tools->Options->Geometry->Visibility, select Lines, Surfaces, Point numbers, Line numbers, Surface numbers. Save definitively your choices with File->Save Options->As default.

-Now, look at the file.geo file and change the variables and add the circle. Rename the boundaries. You will press Reload in the gmsh GUI to update the geometry visualization.

-Mesh in 2D, export it to a MED file then build a Trio_U data file to read the mesh. Visualize the mesh with the **Postraiter_domaine** keyword.

-Try then 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

-In 3D, you will use “Plane Surface” to define the boundaries of your domain and “Physical Surface” (instead of Physical Line in 2D) to name these boundaries:

```
Line Loop(1) = {15,6,1,14};                 // Line loop of several lines
Line Loop(2) = {15,16,17,18};               // Line loop of several lines
Plane Surface(1) = { 1,2 } ;                // Surface/boundary defined
Physical Surface(“Inlet”) = {1};            // Naming boundary is MANDATORY
```

-You will need to define a volume and a physical volume at the end of .geo file:

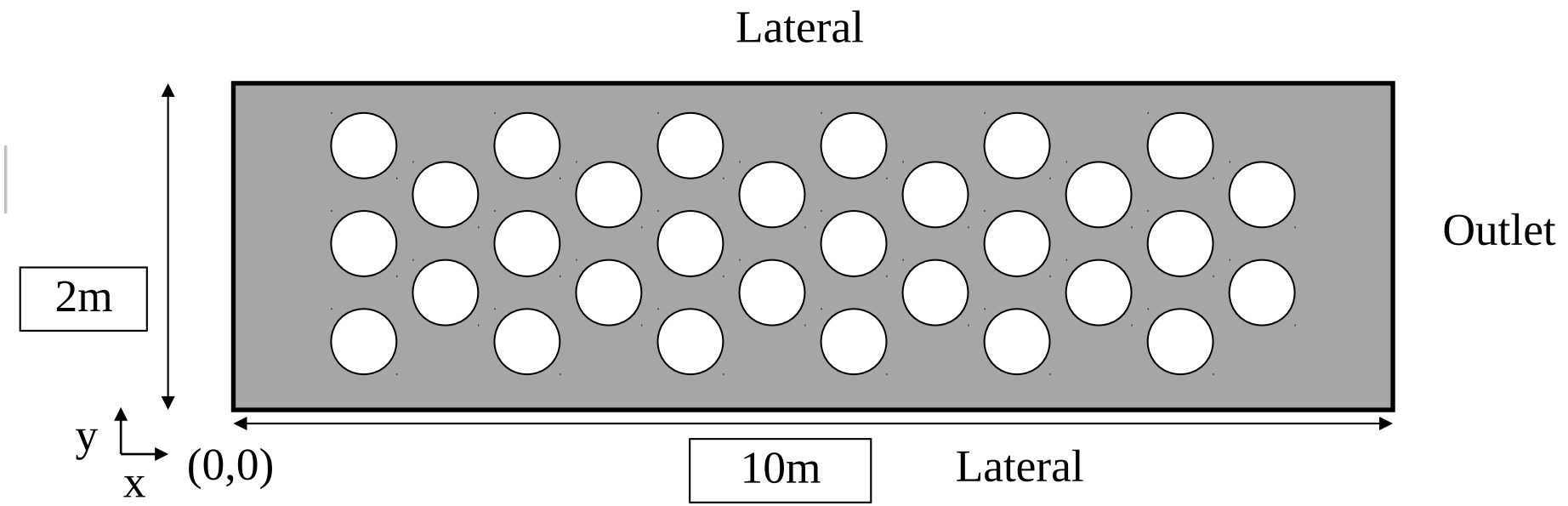
```
Surface Loop(1) = { 1,2,8,12,16,20 } ;        // Surface Loop of Plane Surfaces
Volume(1) = { 1 } ;                           // Volume defined
Physical Volume("domain") = 1;                // Naming domain is MANDATORY
```

-Mesh and export it 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 Trio_U. 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.

Xprepro to create a 2D VDF mesh



The radius of the cylinders is 0.2m and the distances between cylinders are 0.2m in the y direction and 0.6m in the x direction.

In this exercise, you will learn how to create a 2D mesh with Xprepro. You will have a look at the “virus” example (button Examples) to see how to build a loop in the viewlist.

-Run Xprepro in the Trio_U environment :

Xprepro

-First, have a look at the Examples (Copy and Read button on virus example) to understand how you will use fortran code in the viewlist to create loops for building your cylinders.

-Then, click on the button Default to define the initial block and edit the dimensions file. It is a 2D geometry, so we will define zomin=0. and zomax=0.01. The other dimensions of the initial block are xomin=0., xomax=7.8, yomin=0., yomax=2., eps=0.0001 (tolerance).
The cylinder radius is R=0.2 , the coordinates of the centre of the first cylinder, which after will be duplicated, are xoc=0.6 and yoc=0.4. Create 2 Fortran loops to copy the first cylinder in the two directions X and Y.

-With the button Add , build the boundary blocks named **inlet** (index -1000) **outlet** (index –2000), and **lateral** (index 0). Don’t forget to name the blocks.

-Create a 2D cut slice in the XY plane, you will add and define (set IDIR=3) with Add button an object « Meshing creation 2D », and its index will be negative.

-Click Modify on the viewlist to modify the nodes number NX=203, NY=43, NZ=4.

-Check the other objects in the viewlist and save your work with the button Save file prepro then check the name of the boundaries with Boundaries informations.

-Run the model with Modeling run…. A window is opened where you will check the nodes coordinates of the mesh.

-Click on Pre-mesh visualization to check your pre-mesh. Warning, it is always a 3D model in Xprepro, even if you wish a 2D mesh. By default, you see the indexes between 1000 and -3000. If you don’t see the cylinders, create a 2D slice in the XY plane in VisIt to see inside the 3D pre-mesh.

-Zoom onto the boundaries to check the boundary blocks are all defined.
Click Pre-processing run to create the final mesh (check there is no error messages).

-Click on Get geom to generate the Trio_U .geom file in your study.

-If you wish, build a data file to read your mesh and run the flow around the cylinders.