PRESENTATION

Commercial finite volume software packages are the most commonly used in the aeronautical and automobile industries. The most important ones are FLUENT, ANSYS CFX and STAR-CCM+. STAR-CCM+ is offered to the end user as a Physics simulation integrated package. Much more than just a CFD solver, STAR-CCM+ aims at being an entire engineering computational procedure for solving problems involving flow (of fluids or solids), heat transfer and stress.

This material consists of a collection of exercises and tutorials used in the ATHENS course UPM41 CFD WORKSHOP, which is given as well as the free configuration subject at the UPM, which has adopted STAR-CCM+ as the reference commercial package.

This course is organized around a set of tutorials through which the main capabilities of the software are demonstrated. These tutorials cover fundamental problems of fluid mechanics (viscous laminar flows, turbulent flows, free surface flows, etc…) as well as consolidating the more fundamental fluid Mechanics core knowledge.

AUTHORS

Antonio Souto Iglesias	antonio.souto@upm.es
Carlos Ariel Garrido Mendoza	carlosariel.garrido@upm.es
Leo Miguel González Gutiérrez	leo.gonzalez@upm.es

INTRODUCTION

In this course, we aim at providing an introduction to the use of CFD codes in Engineering. The specific objectives are:

1. Learn to us a finite volume CFD.
    a) Physical Models (viscous and free surface flows)
    b) Meshing
    c) Execution
    d) Post-Processing.
2. Establish connections with Fluid Dynamics background.

In order to achieve these targets, this document is divided into five tutorials:

1) Tutorial 1: POISEUILLE FLOW

2) Tutorial 2: PLANE POISEUILLE WITH A TURBULENCE MODEL

3) Tutorial 3: FLOW PAST A CIRCULAR CYLINDER

4) Tutorial 4: DAM BREAK

Each tutorial describes in detail the steps to follow on how to appropriately solve the proposed problems thus making it much easier for the reader to follow. A great number of images have been used throughout these tutorials to further aid the reader in understanding the main aspects of the software interface. Once a tutorial has been completed, various similar unsolved exercises are proposed in order to complement and reinforce the recently acquired knowledge. Both the tutorials and exercises often require the use of certain CAD files, which are also provided. Furthermore, in order to carry out some comparisons with reference data, some MATLAB scripts are provided together with references to the scientific literature.

We hope these tutorials will be useful in familiarizing students with the software and its capabilities, but more importantly, helping them understand the nature of the problems as well as their underlying mechanics.

CFD WORKSHOP

UPM 41 ATHENS COURSE
STAR CCM+ TUTORIAL Number 1
POISEUILLE FLOW

CONTENTS

1 NOTATION

BC	Boundary condition.
DC	Double click
F2	Rename
FF	Field Function
LC	Left click
NS	Navier Stokes
RC	Right click

2 RELATED DOCUMENTS

None.

3 DESCRIPTION OF THE PHYSICAL PROBLEM

The general equations that govern unsteady 2D Newtonian incompressible viscous flows are the Navier-Stokes equations in its incompressible form:

$$u_x + v_y = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial x} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial x} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{3}$$

The Poiseuille flow is a laminar horizontal flow between two parallel walls driven by a constant pressure gradient $-\partial p/\partial x$ = ***K***. The steady state x-momentum equation takes the form:

$$0 = \frac{K}{\rho} + \nu\frac{\partial^2 u}{\partial y^2} \tag{4}$$

It is supposed that the channel has a width 2b in the *y*-direction.

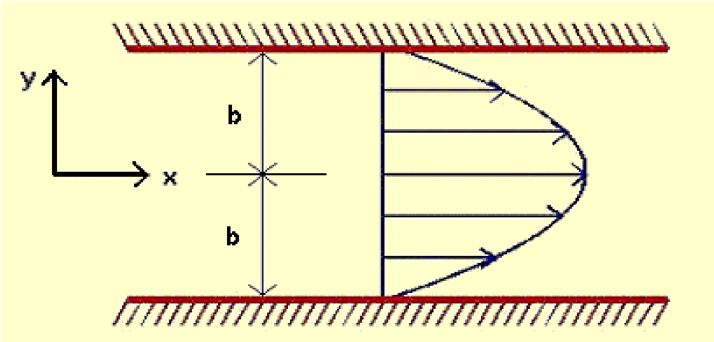

Figure 1: Domain of the Poiseuille problem

The no-slip boundary condition at the top and bottom edges of the channel reads

$$v_x(x,b) = v_x(x,-b) = 0 \tag{5}$$

The steady state solution of (3) and (4) can easily be obtained (see reference [Batchelor, 1967])

$$u(y) = \frac{K}{2\cdot\mu}\left(b^2 - y^2\right) \tag{6}$$

4 PROBLEM SETUP

The STAR-CCM+ version used in this tutorial is 5.04.006.

The parameters documented in section 3 will take the following values in the simulations, with L being the length of the computational domain.

$$K = 1Pa/m,\ b = 0.7m,\ L = 6\text{m},\ \mu = 1Pa.s$$

$$u_{max} = \frac{K\cdot b^2}{2\mu} = 0.5\cdot 0.7^2 = 0.245m/s$$

$$\text{Re} = \frac{u_{\max}\cdot 2b}{\mu/\rho} = 0.245\cdot 1.4 = 0.356$$

The geometric information is provided as a separate file, as will be described in section 6.

Periodic boundary conditions are considered at the left and right side of the computational domain (see figure 1). Density is irrelevant in this problem but is supposed to be 1 kg/m3 for completion of the data.

5 TOPICS COVERED

This is the first tutorial of this series. Therefore all the basic topics related with running a simulation with STAR-CCM+ have to be discussed. A list of these is now provided:

1. Importing a geometrical domain.
2. Building a 3D regular Mesh.
3. Converting a 3D mesh into a 2D mesh.
4. Defining the characteristics of the physical problem, which consists of a mono-phasic viscous flow.
5. Enforcement of no-slip BC.
6. Enforcement of periodic inflow/outflow BC.
7. Creating user-defined auxiliary functions that are necessary to perform specific operations. In the STAR-CCM+ terminology, such functions are referred to as FIELD FUNCTIONS.
8. Defining the setup parameters for starting the solver (time step, convergence criteria, …)
9. Generating output information: Cutting and representation of a section.

6 SIMULATION SETUP AND SOLUTION

6.1 Start STAR-CCM.

Execute STAR-CCM+ either with the corresponding ICON or from the Windows menu.

6.2 Import Geometry 3D

First, we have to load the file with the geometrical information of the computational domain. The file is named *Poiseuille.igs* and can be found in the corresponding folder.

File → New Simulation → OK

File → Import→ Import Surface Mesh ( ) → … /Poiseuille.igs → Open

After clicking on open, the pop-out window shown in figure 1 will appear. Select the options as shown. In *Region Mode:* select the “One region per body” option so that only one computational domain is created. In *Tessellation Density* choose “Fine” so that the geometrical definition, consisting essentially of polyhedrons obtained by tessellation the geometry in Poiseuille.igs, is defined with good quality.

Press OK.

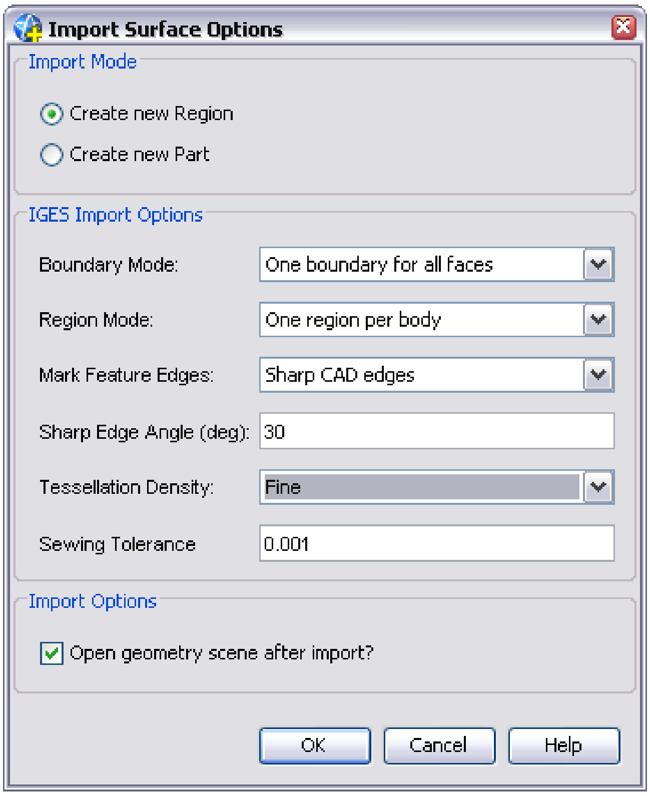

Figure 1 : Input surface options dialog window.

Now that the 3D geometry has been loaded, you can rotate it by left clicking on the *Geometry Scene 1* section of the screen. Now is a good time to save the project in your personal folder (Z:)

Save ( ) → /poiseuille.sim

6.3 Preparing Surfaces

6.3.1 General

When geometrical information is imported, such information constitutes a new *Region* and appears in the Region part of the tree menu (figure 2) as Shell (Shell)

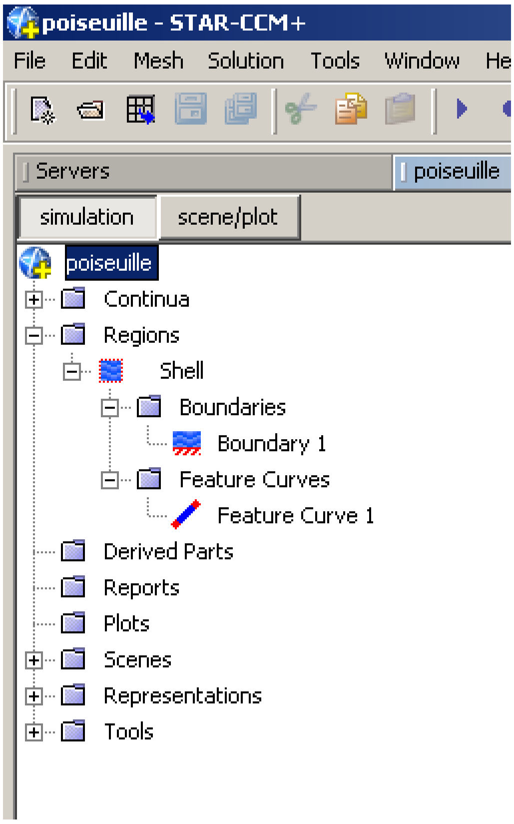

Figure 2 : Simulation Tree.

We will rename it as CHANNEL.

Regions → Shell, F2 → CHANNEL

After renaming the region, press *Enter*. All changes in STAR-CCM+ are really effective only after pressing Enter. Moving the mouse pointer to another region and clicking does not confirm the choice, contrary to what is usual in many codes.

Since the geometry has been imported as a block, it is important to split the different surfaces that represent the 3D object so that different BC's can be applied on the corresponding surfaces.

Regions → CHANNEL → Boundaries → Boundary 1, RC → Split by Angle → Apply → Close

Let us rename the different surfaces that appear in our geometry

Regions → CHANNEL→ Boundaries → Boundary 1, F2 → FRONT
Regions → CHANNEL→ Boundaries → Boundary 1 2, F2 → INLET
Regions → CHANNEL→ Boundaries → Boundary 1 3, F2 → TOP
Regions → CHANNEL→ Boundaries → Boundary 1 4, F2 → OUTLET

Regions → CHANNEL→ Boundaries → Boundary 1 5, F2 → BOTTOM
Regions → CHANNEL→ Boundaries → Boundary 1 6, F2 → BACK

Let's check whether all surfaces have been correctly created in the *Regions* branch of the options tree (figure 3), and their geometrical position is consistent with their names. In case the scene is not open do the following:

Scenes → + → Geometry Scene 1, RC Open.

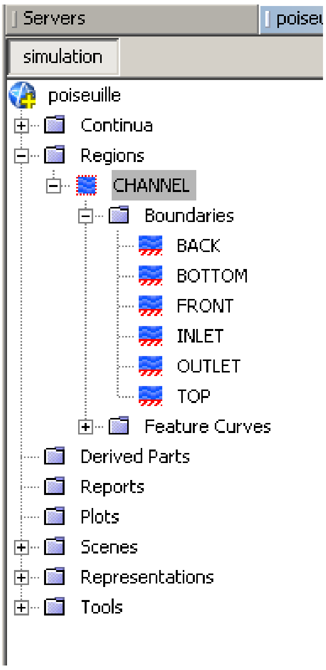

Figure 3 : Simulation Tree after renaming the region and its boundaries.

To orientate the view for an easy way to check the boundaries names, proceed as shown in figures 4 and 5. Clicking on the boundaries names shows the selected boundary in the right window (figure 6). Rotate the view by LC on the geometry scene 1 window.

In order to go back to previous views, click on the icon .

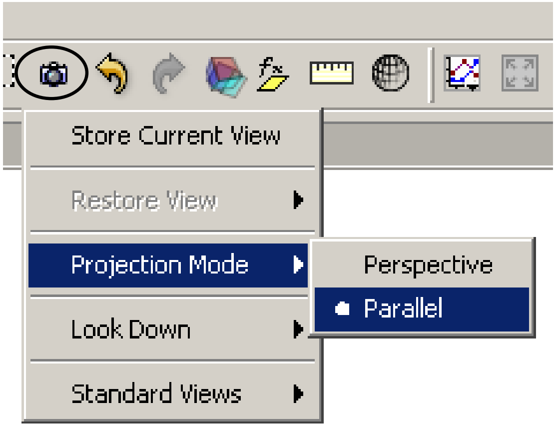

Figure 4 : Standard reference frame orientation (parallel projection mode)

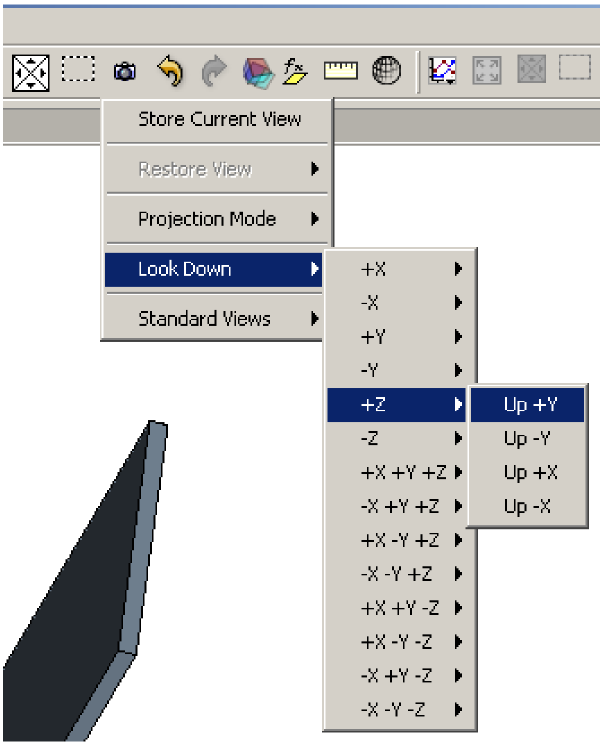

Figure 5 : Standard reference frame orientation.

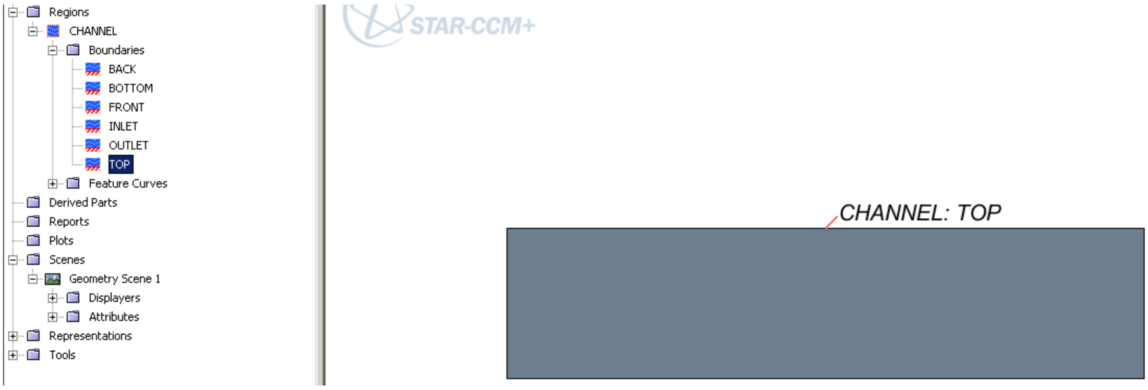

Figure 6 : Boundaries signaling.

For safety,

File → save

6.4 Physics Models

6.4.1 General

The physical model including the initial conditions has to be defined.

Continua, RC → New → Physics Continuum

Continua→ Physics 1→ Models→ DC→ Three Dimensional, Implicit Unsteady, Liquid, Segregated Flow, Constant Density, Laminar→ Close

Note: The Segregated Flow model solves the flow equations (one for each component of the velocity, and one for the pressure) in a segregated or uncoupled manner. The linkage between the momentum and continuity equations is achieved with a fractional step approach. For more information, check reference 5 (pgs. 2040 to 2063).

The liquid properties are tuned so that the convergence to the steady state solution is speeded up.

Continua→ Physics 1→ Models→ Liquid→ H2O→ Material Properties→ Density→ Constant→ 1

Continua→ Physics 1→ Models→ Liquid→ H2O→ Material Properties→ Dynamic Viscosity→ Constant→ 1

In a Poiseuille flow, the motion is driven by a constant pressure gradient. This pressure gradient acts like a body force (similarly to gravity for instance). This term is incorporated into our project by activating the following options.

Regions→ CHANNEL→ Physics Conditions→ Momentum Source Option→ Specified

The pressure gradient is substituted by a body force, and since the pressure gradient has a negative sign in the NS equations, the corresponding body force must have its sign swapped. If we want a negative pressure gradient, an equivalent positive body force must be introduced in our project.

Regions→ CHANNEL→ Physics Values→ Momentum Source→ Constant→ Value→ [1.0, 0.0, 0.0]

6.4.2 Boundary conditions (1)

All surfaces carry a no-slip BC by default. This can be checked by clicking e.g.:

Regions → CHANNEL→ Boundaries → Bottom →

And seeing that Type is activated as Wall and the Shear Stress Specification: no-slip (see figure 7)

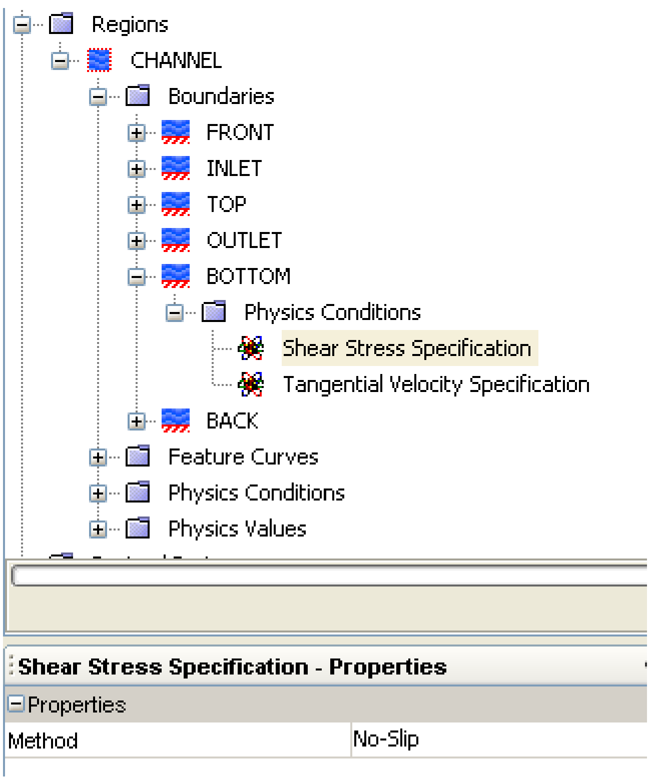

Figure 7: No-slip BC

Now we must change the surfaces (INLET, OUTLET) that have a different boundary condition.

Regions → CHANNEL→ Boundaries → INLET → Type → Velocity Inlet (as in figure 8)

Regions → CHANNEL→ Boundaries → OUTLET → Type → Velocity Inlet

Usually, a *Pressure Outlet* condition is used at the exit. In this particular case, since this flow is periodic, a *Velocity Inlet* condition is used instead.

File → Save

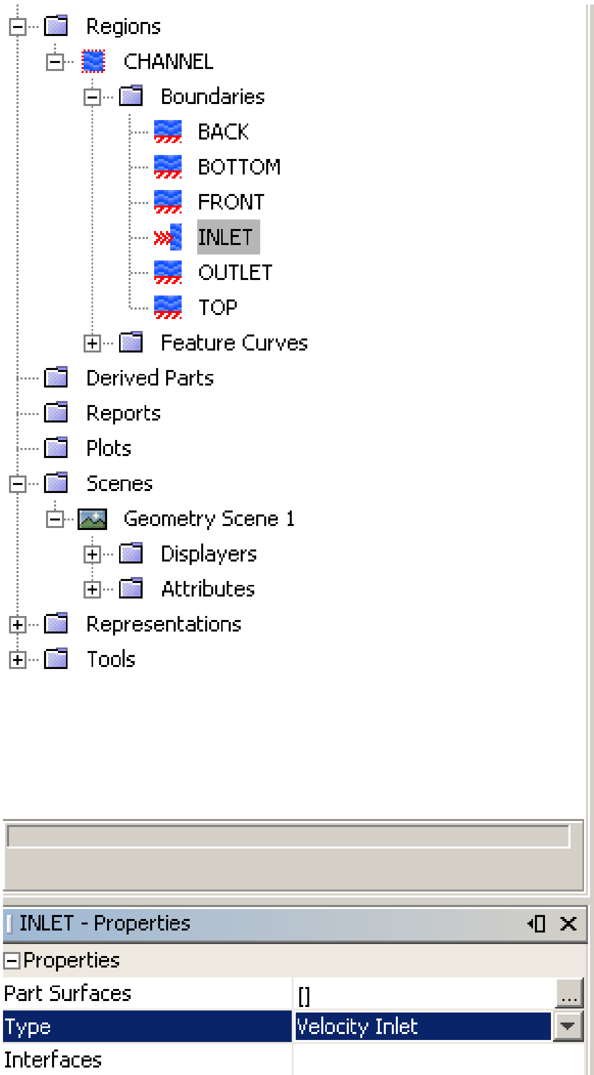

Figure 8: BC setup.

6.5 Mesh

6.5.1 General

The characteristics of the desired mesh must be specified. First, a *Surface Mesh* model is necessary. Since in principle some regions require a mesh refinement (especially those regions closer to the top and bottom boundaries), the option *Surface Remesher* must be selected in the *Meshing model selection* dialog box (figure 9). The *Surface Remesher* model is used to re-triangulate an existing surface in order to improve the overall quality of the surface and optimize it for the volume mesh models. The remeshing is primarily based on a target edge length that you supply and can also include refinement based on curvature and surface proximity. Localized refinement based on boundaries can also be included.

Even if we are simulating a 2D problem, in Star CCM+ the mesh has to be created in 3D and later projected in 2D. The *Trimmer* option is used in order to carry out this procedure. With this option, those cells that extend across the boundaries will be trimmed. The *Trimmer Meshing* provides a robust and efficient method of producing a high quality grid for both simple and complex mesh generation problems. It provides predominantly hexahedral mesh with minimal cell skewness.

The mesh characteristics are inside the Continua branch of the Simulation tree. The same region can have different meshes and therefore, Mesh and Region belong to different branches of the tree.

Continua → Mesh 1 → Models, DC → Surface Remesher, Trimmer → Close

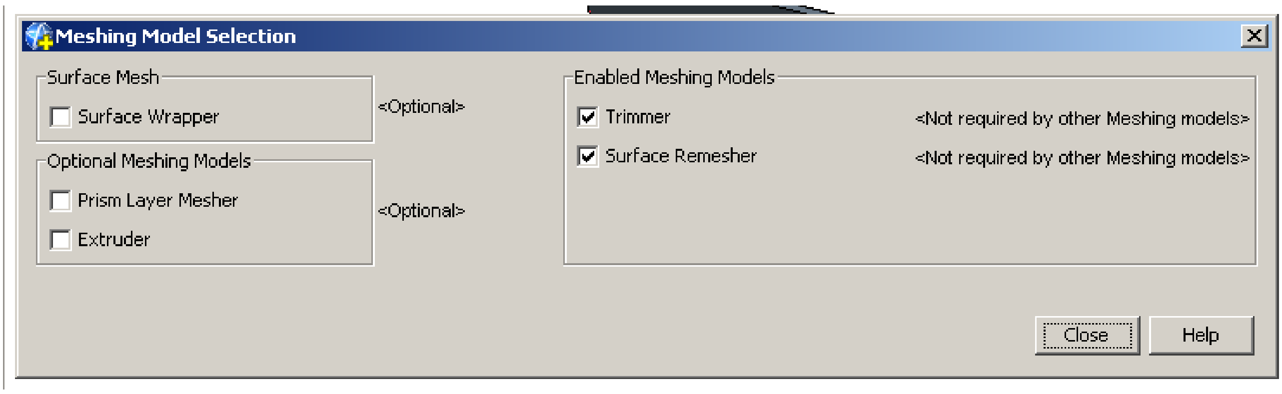

Figure 9: *Meshing model selection* dialog box

A *base size* (reference size) for the mesh must now be selected. This means that all the relative distances will be referred to this dimensional base size. It is important to check the characteristic size of the body that we are studying and make the reference values similar to it. Use the ruler icon (see figure 10) if necessary to have an approximate idea of the characteristic size of the imported object.

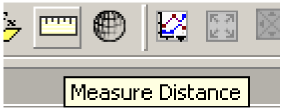

Figure 10: Measure distance functionality

Continua → Mesh 1 → Reference Values → Base Size =1m

A maximum relative size of the mesh elements as a factor of the Base Size is now defined.

Continua → Mesh 1 → Reference Values → Maximum Cell Size → Relative Size (Percentage of base) = 5

A minimum and a target relative size for the surface mesh that is built as a previous step to the volumetric mesh are now provided.

Continua → Mesh 1 → Reference Values → Surface Size → Relative Minimum Size = 5
Continua → Mesh 1 → Reference Values → Surface Size → Relative Target Size = 5

File → Save

6.5.2 Mesh Generation

First, a surface mesh must be generated and visualized (recommended).

Mesh (menu bar) → Generate Surface Mesh (the corresponding icon can be directly used instead)
Scenes, RC → New Scene → Mesh

Second, the volumetric mesh must be generated and visualized (recommended)

Mesh (menu bar) → Generate Volume Mesh.
Over the scene, RC → Apply Representation → Volume Mesh (figure 11)

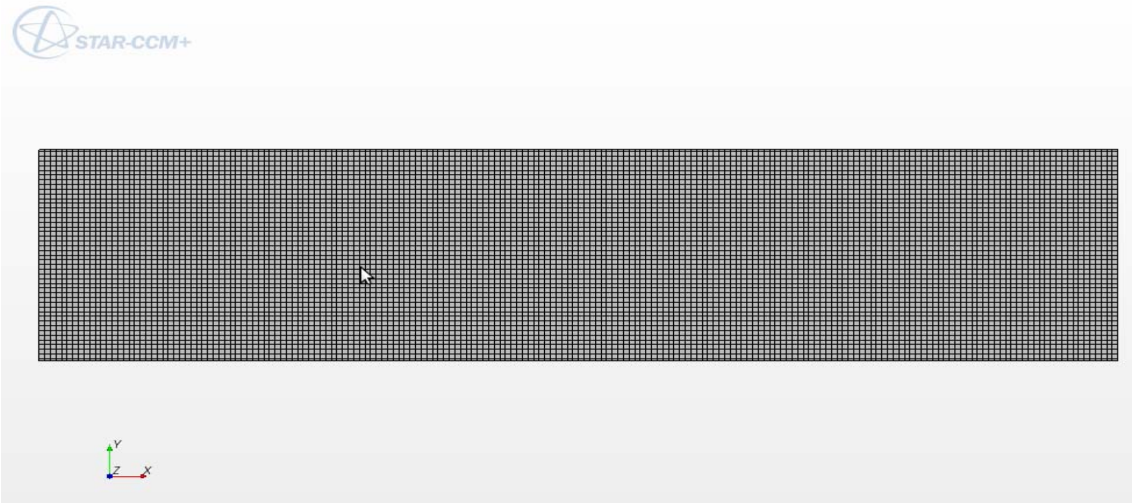

Figure 11: Volumetric mesh.

A regular structured mesh has been created.

File → Save

6.5.3 Boundary conditions (2) (on the mesh)

An interface between the INLET and OUTLET boundaries must be created so that they share the periodic BC. This applies to the mesh elements that lay on those boundaries.

Regions→ CHANNEL→ select INLET and OULET simultaneously by keeping *Ctrl* key pressed→ RC→ Create Interface→ Periodic

There are a range of options to define once a periodic BC is established between two boundaries through and interface. In our particular case, the periodicity is referred to all physical magnitudes, and therefore, the option is that the interface is "Fully Developed" (figure 12).

Interfaces→ Periodic 1→ Type→ Fully-Developed Interface

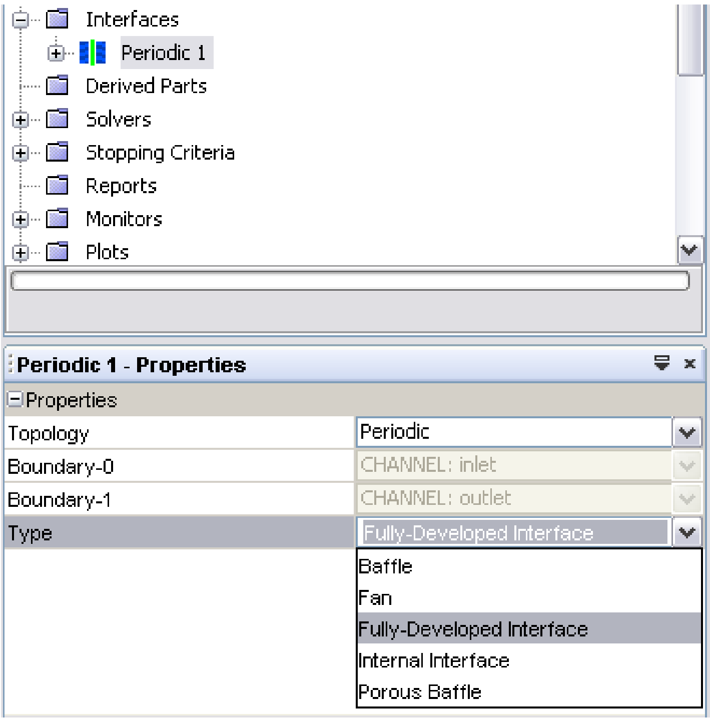

Figure 12: Periodic interface type definition.

6.5.4 Convert to 2D (Field functions)

The mesh region has to be converted to 2D. There are special requirements in STAR-CCM+ for 3D meshes to be converted to 2D ones. These requirements are:

- The grid must be aligned with the X-Y plane.
- The grid must have a boundary plane at the Z=0 location.

The first condition applies but the second does not. This can be appreciated by e.g. visualizing the coordinates system:

Tools → Coordinate Systems →Laboratory

and observing that the origin is in the center of the CHANNEL region.

We need to split the geometry at the plane z=0. In STAR-CCM+, this is achieved with the use of *“FIELD FUNCTIONS”*. These functions are created by the user and are used as auxiliary tools in many instances when preparing cases for computation. In this particular case, a new field function will be created with the aim of checking whether the *z* coordinate of the Centroid of an element is greater than 0. To define this function:

Tools** → **field functions, RC** → **new** → **UserFieldFunction_1, F2 → (rename it as) ***Split***

In the bottom we type in *Function name*, ***Split*** (see figure 13), and in definition we copy this:

($$Centroid[2] > 0.0) ? 1.0 : 0.0

which uses an EXCEL like syntax for conditionals. Note that 3D position coordinates are indexed in STAR-CCM+ as 0,1,2.

The name of the field function in the Field Functions branch is the one used for the scenes and the one in the Function Name box is used to refer to this Field Function from another function.

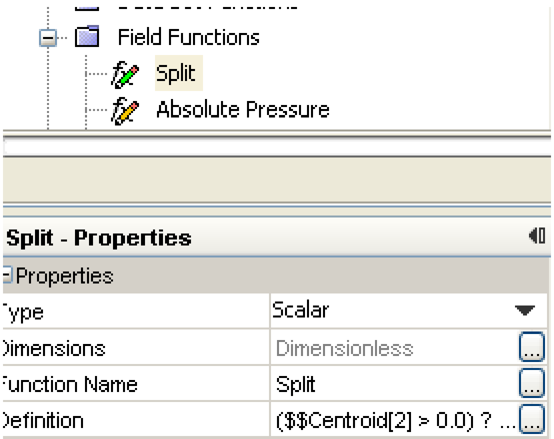

Figure 13:.Field function definition.

Let’s see now whether the zones defined by this split are consistent with the idea of projecting the object onto the Z=0 plane. To do this,

Scenes → **RC** → **new scene** → **scalar**

Drag the Region CHANNEL to this Scene and click *"Add to Scalar 1"*

In the right window RC on the blue rectangle in the bottom and select the function *split*. By rotating the figure (LC), you should be able to see something similar to figure 14. The split function value is either 0 or 1 depending on whether z>0 or z<0, which is consistent with what can be seen in such a figure.

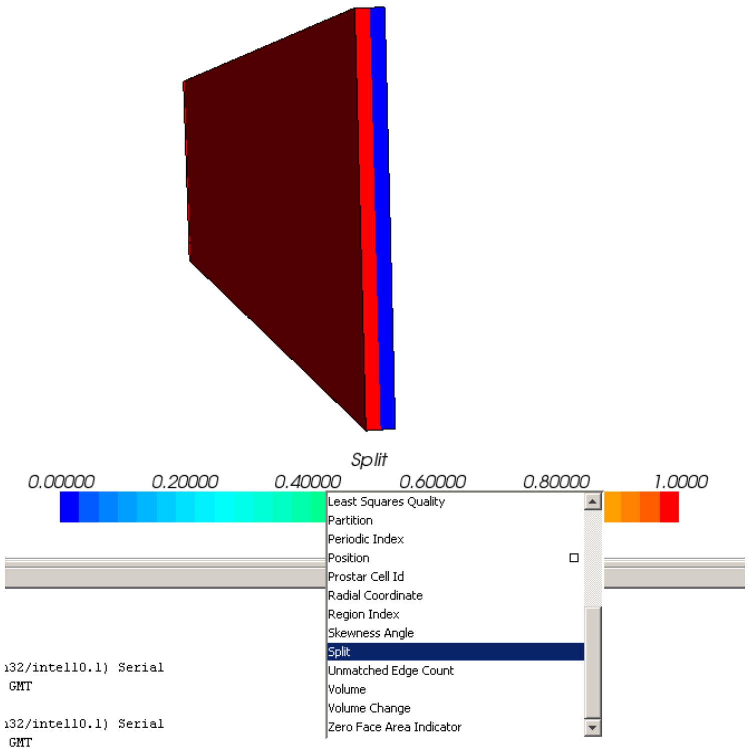

Figure 14:.Split field function scalar scene.

By relying on this *Split* function, it is now possible to trim the tank so we can work in 2D.. In order to do this we go to

Regions → CHANNEL, RC→ Split By Function→ <<Select function>> → Split→ apply→ close.

A region called CHANNEL 2 appears which may be removed by pressing Del after selecting it.

Regions → CHANNEL 2, Del.

File → Save

Now the mesh can be converted to 2D

Mesh (menu bar) → **Convert to 2D→ Ok**

We now have a new region, a 2D one, whose name will be CHANNEL 2D. The project must now be saved with a new name.

File→ Save As→ Poiseuille2D.sim

The simulation tree should look like the one presented in figure 15

We must now unselect Physic and Mesh from the CHANNEL Region. The reason for this is that we aim at running a 2D simulation and the Physics must only apply to the 2D Region.

Regions→ CHANNEL→ Mesh Continuum → None

Regions→ CHANNEL→ Physics Continuum → None

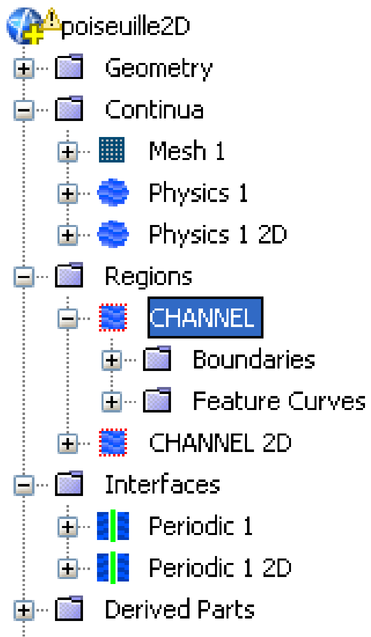

Figure 15:.Simulation tree after mesh conversion to 2D.

6.6 Solver(time-steps)

The project is now ready to be run. Nonetheless, running a CFD simulation requires specifying some characteristics of the solver phase. Specifically the convergence criteria in the pressure coupling algorithm, time step, maximum simulation time, etc… have to be set up. First, the time step is defined.

Solvers → Implicit Unsteady → Time-step → 0.1s

The criterion for defining the time step relies on both the accuracy and the stability of the simulation. This will be discussed in further detail later.

The number of iterations per time step in the coupling algorithm is now defined.

Stopping Criteria → Maximum Inner Iterations → 3
Stopping Criteria → Maximum Physical Time → 60
Stopping Criteria → Maximum Steps → Disable by un-clicking the box

6.7 Run

6.7.1 Preparing a Scalar Scene

Since we are interested in viewing the velocity evolution (velocity module), a new scene is created.

Scenes, RC → New Scene → Scalar
Scenes→ Scalar 1, F2 → Velocity

In the new Scene, RC over the blue line

Velocity → Magnitude

6.7.2 Run

It is now time to run our project.
File → Save

Solution → Run ( )

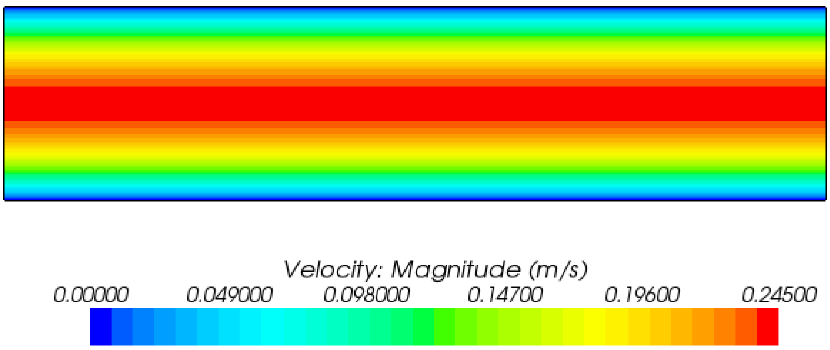

Figure 16:.Poiseuille steady state solution.

Run the simulation until you consider that a steady state solution has been reached, defined as negligible variations in the values of the velocity. An image such as the one in figure 16 should be obtained.

To export an image of the current Scene, follow the next steps figure 17:

RC on the Scene → Hardcopy… → Save in your own folder.

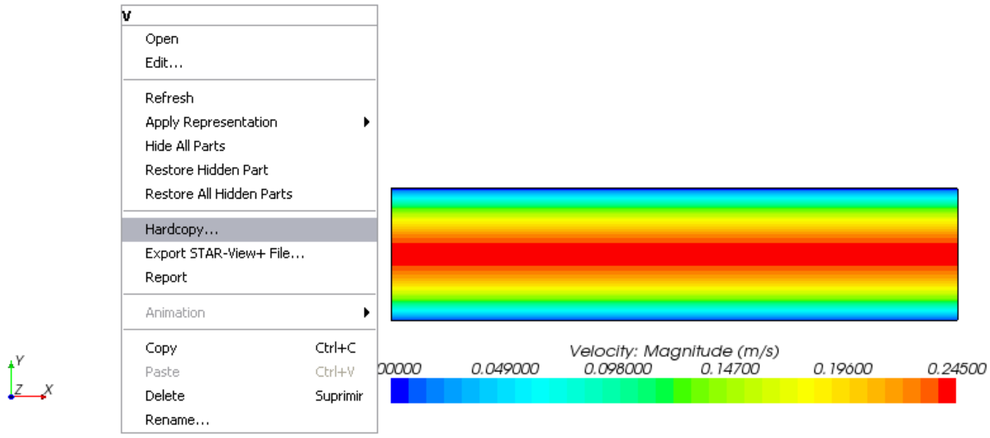

Figure 17: Image Hardcopy.

7 POST-PROCESSING EXTRAS

7.1 Velocity Profile

Running a test case for which an analytical solution is available provides valuable feedback in regards to both the quality of the CFD package and the user ability to use it. Since we have obtained a steady state solution, we can compare it with the exact one. To achieve this, a section of the velocity profile must be created. STAR-CCM+ has some resources to define specific areas where flow fields may be monitored. Such zones are named *"Derived Parts".* A vertical line where the horizontal velocity magnitude will be inspected is going to be created.

Derived Parts, RC → New Part → Probe → Line...
Input Parts→ CHANNEL2D→ OK
Point 1 → [0.0, -0.7, 0.0]
Point 2 → [0.0, 0.7, 0.0]
Resolution → 15 (to set the number of points to define the probe line)
Create → Close.
Derived Parts → line-probe → F2 → x-velocity probe

The probe has now been created and a plot fed from that probe will be the main tool used to see and eventually export that information.

Plots, RC → New Plot → XY
Plots → XY Plot 1, F2 → Vx
Plots → Vx, RC → Edit →
 [ ] → x-velocity probe (see figure 18**)**
 X Type → Type → Scalar
 X Type → Scalar → Velocity[i] (it may take a few seconds to update)
 Y Types → Y Type 1 → Type → Position

Y Types → Y Type 1 → Position → Direction → [0.0, 1.0, 0.0]
Y Types → Y Type 1 → x-velocity probe → Line Style → Style → Solid
Close

A similar figure to 19 should be obtained.

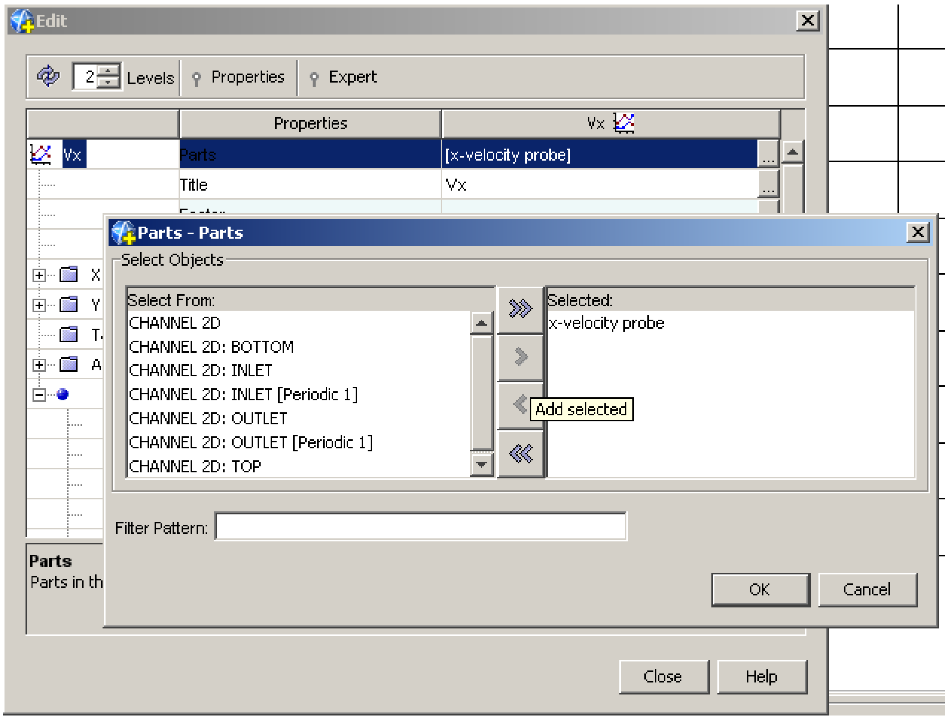

Figure 18:.Assign the x-velocity to the probe.

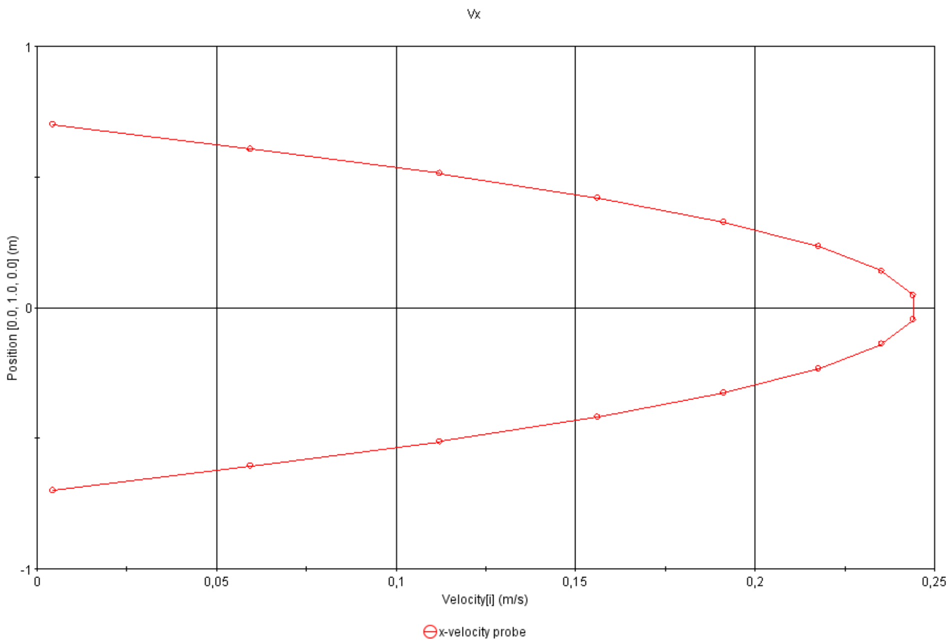

Figure 19: Plot of the x-velocity to the probe.

The data in this plot can be exported as an ASCII file and compared in MATLAB for instance with the exact solution.

RC on the plot → export → vx (select your working folder) and select Field Delimiter *Space ()*.

A file named vx.csv with this shape will appear now in your folder:

"Velocity[i] (m/s)-x-velocity_probe"	"Position [0.0, 0.1, 0.0]-x-velocity_probe (m)"
0.0175	-0.7000
0.0500	-0.6067
0.1075	-0.5133
0.1550	-0.4200
0.1925	-0.3267
0.2200	-0.2333
0.2375	-0.1400
0.2450	-0.0467
0.2450	0.0467
0.2375	0.1400
0.2200	0.2333
0.1925	0.3267
0.1550	0.4200
0.1075	0.5133
0.0500	0.6067
0.0175	0.7000

Open MATLAB and use script **uy_poiseuille.m** (available in the course materials folder) to compare the STAR-CCM+ graph with that of the exact solution.

7.2 Forces

The force on the bottom plate can be compared with the exact solution, which is defined by the following formula:

$$F = \tau \cdot Area \tag{7}$$

where,

$$\tau = \mu \cdot \left.\frac{\partial u}{\partial y}\right|_{y=-b} = K \cdot b = 1 \, {}^{Pa}\!/_{m} \cdot 0.7m = 0.7Pa \tag{8}$$

then, replacing the τ value and the length of the computational domain *L* in the previous formula we obtain the force *F* expressed per unit length.

$$F = 0.7Pa \cdot 6m = 4.2 \, {}^{N}\!/_{m}$$

In Star CCM+ we can extract information from the simulation. The functionality that helps the user in this is the *"Report"*. Each report is represented by a node in the simulation tree, with its own pop-up menu.

The Reports are useful for post-processing, and enable us to track the evolution of relevant engineering quantities such as drag, lift, torque, or mass flow. These reports are also useful in computing diagnostic quantities such as minimum cell volume or average wall.

To create any new report, do the following,

Report, RC→ New Report → Force (figure 20)

The reports can also be renamed, and once a report is created it will be saved with the simulation.

Report → Force 1, F2 → Bottom Force

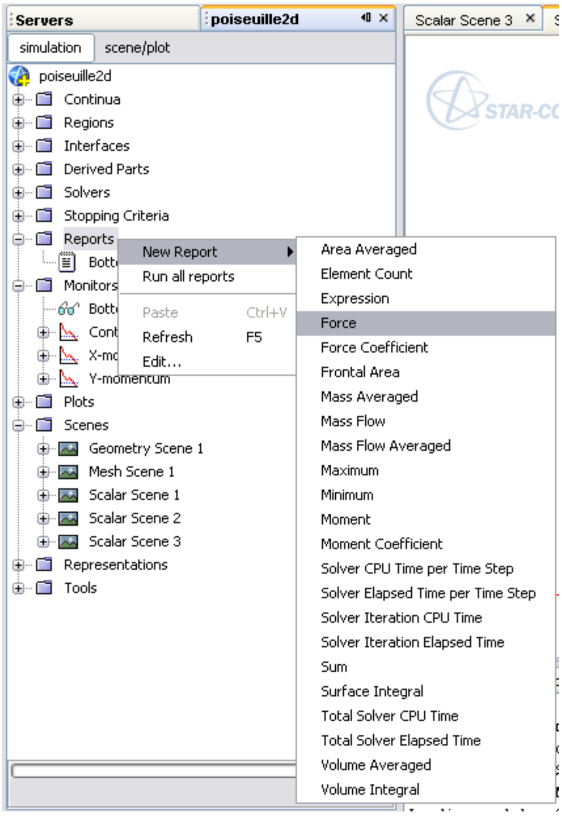

Figure 20: Adding reports.

The new force report has default properties where usually some of them must be changed; after configuring the report we should have something as shown in figure 21.

Report → Bottom force → Force Option → Shear

Report → Bottom force → Parts → [ ] → Select CHANNEL 2D: BOTTOM

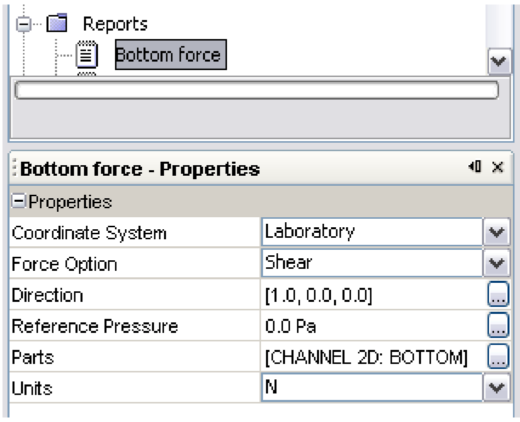

Figure 21: Report setup.

The *Reports* can be observed *in real time* while the solution is iterating, by creating a monitor. This is useful when you are interested in the evolution of a quantity over time, or if you want to use this report as one of your stopping criteria. The *report monitors* can also be plotted (Figure 22)

Report → Bottom force, RC → Create Monitor and Plot from Report
Plots →Bottom force Monitor Plot, RC → Open

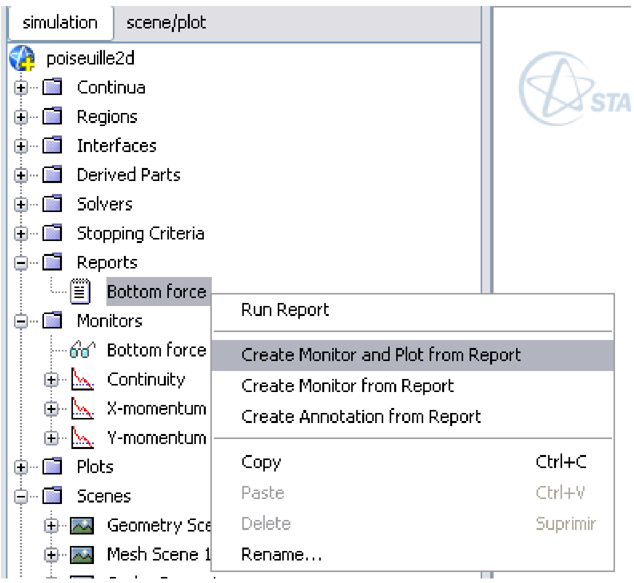

Figure 22: Report Monitor and Plot.

After creating the monitor and plot from your report, you should see something like figure 23 in your simulation tree. Check Reports, Monitors and Plots

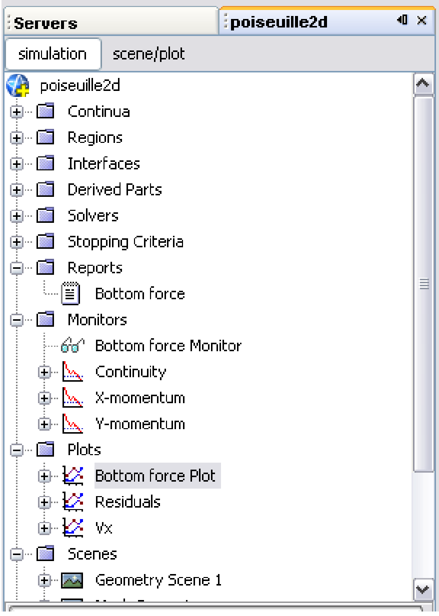

Figure 23: Simulation tree with monitor and plot from report.

Clear solution and run again the simulation in order to compute this report and the corresponding monitor and plot.

Solution → Clear Solution → Fields (selected)

Solution → Run

8 EXERCISES

1 Plane Poiseuille flow over a slope

The pressure drop that drives the motion is substituted in this exercise by gravity applied to a fluid running on a slope between two plates. The geometry corresponding to this exercise can be imported from file *poiseuille_exercise_1.igs*. The figure 24 shows the constructive characteristics of the geometry, where *L = 6m*, *h = 1.4m* and *α = 15º*. Repeat the tutorial for this case, using the same value for all parameters, and check the accuracy of the results analogously to the horizontal plane Poiseuille case.

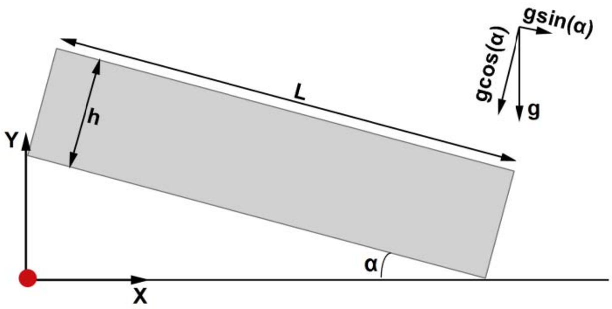

Figure 24: Configuration for exercise 1 Plane Poiseuille flow over a slope.

Tips:

1. Use 1.5 for the density ρ and 1.0 for the dynamic viscosity μ.
2. Do not use any momentum source. The momentum changes are induced by gravity.
3. Gravity is included as another option when defining the setup for the physics of the simulation, like in the initial condition part of section 6.4 of the tutorial.

2 Couette flow.

Couette flow is frequently used in undergraduate Physics and Engineering courses to illustrate shear-driven fluid motion. The simplest conceptual configuration finds two infinite, parallel plates separated by a distance *2b*. One plate, say the top one, translates with a constant velocity *U* in its own plane (figure 25). The steady state profile is a parallel velocity field that matches the wall velocities.

Taking U=1m/s, b=0.7m, and use 1 for the density ρ and dynamic viscosity μ, the steady state solution for this particular configuration is:

$$u(y) = \frac{U}{2}\left(\frac{y}{b} + 1\right) = 0.7142\, y + 0.5$$

The geometry corresponding to this exercise can be imported from file *couette_exercise.igs*.

Run the simulation until the steady state is found using the same mesh configuration as for the Poiseuille case.

Compute the force on the bottom plate and compare with the exact solution.

Tips:

1. As aforementioned, in a Couette flow, the motion is driven by a constant wall velocity. This BC is incorporated into our project by activating the following options.

Regions → CHANNEL → Boundaries → TOP → Physics Conditions → Tangential Velocity

Specification→ Method → Vector
Regions → CHANNEL → Boundaries → TOP → Physics Values → Velocity → [1.0, 0.0, 0.0]

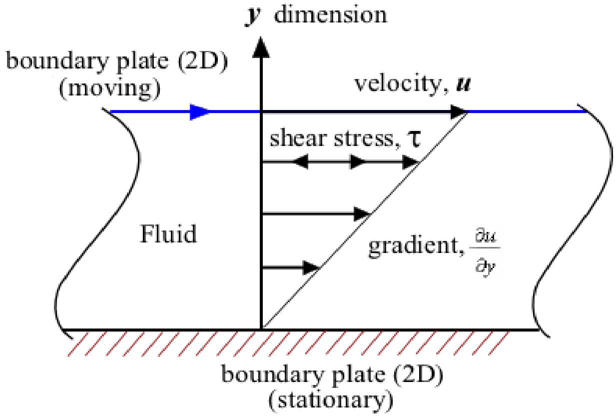

Figure 25: Couette flow

3 Vortex Spin down.

This exercise consists in simulating the spin down of an initially rigidly rotating fluid in a cylindrical container (figure 26).

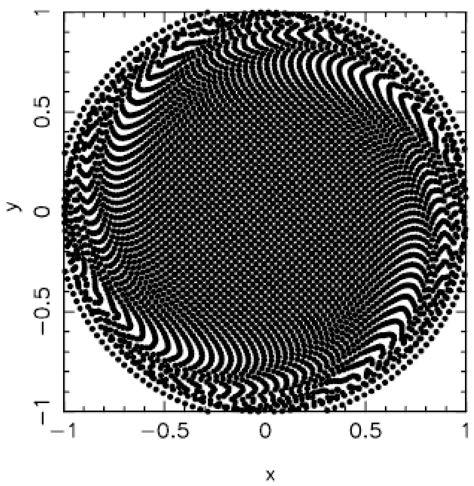

Figure 26: Vortex Spin-down

For this problem, the time variation of the θ component of the velocity can be obtained from the formula given by Batchelor (1967, page 204).

$$v_\theta = -2\Omega R \sum_j \frac{J_1(\lambda_j r/R)}{\lambda_j J_0(\lambda_j)} e^{-\lambda_j^2 \nu t/R^2}$$

where *R* is the radius of the cylinder, Ω is the angular velocity, J_1 and J_0 are Bessel functions, and λ_j is the jth zero of J_1.

For the present calculations, Ω =1rad/s and R = 1m. Assume the fluid is water, as in the tutorial example, but assume the dynamic viscosity is 0.0308 Pa·s and density is 1 Kg·m^3. The geometry corresponding to this exercise can be imported from file *poiseuille_exercise_spindown.igs*. This is the same case presented in reference [Monaghan, 2005]. Use the same base size for the mesh as for the Poiseuille case and the same cell size.

Compute with STAR-CCM+ the radial distribution of the *θ* component of the velocity for t=1.14s using a time step of 0.01s. Compare it to the exact solution as provided by MATLAB file *spindown.m* by creating a probe line to measure the velocity and exporting the graph in file *'vtheta.csv'*. A plot similar to the one in next figure should be obtained:

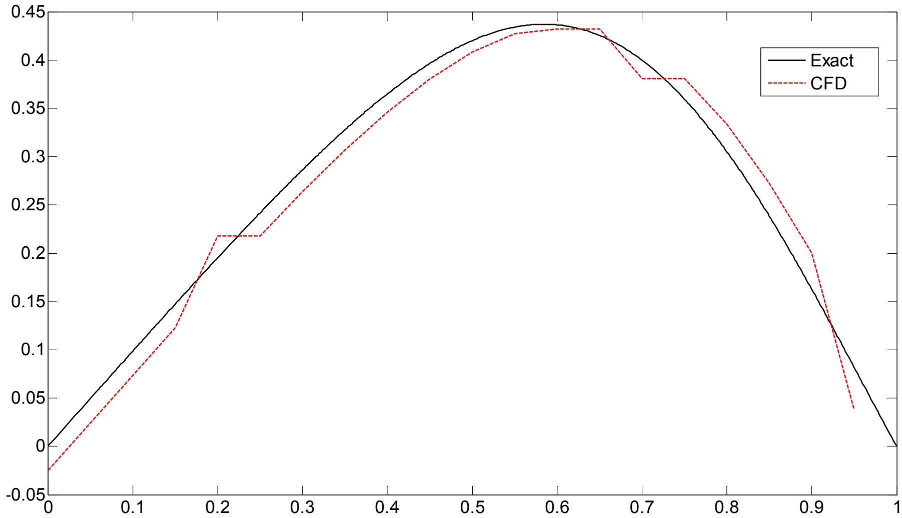

Figure 27: Vortex Spin-down radial distribution of *θ* component of the velocity

This plot is consistent with the velocity field, as can be seen in the next figure, where the line probe for plotting the previous graph is also shown:

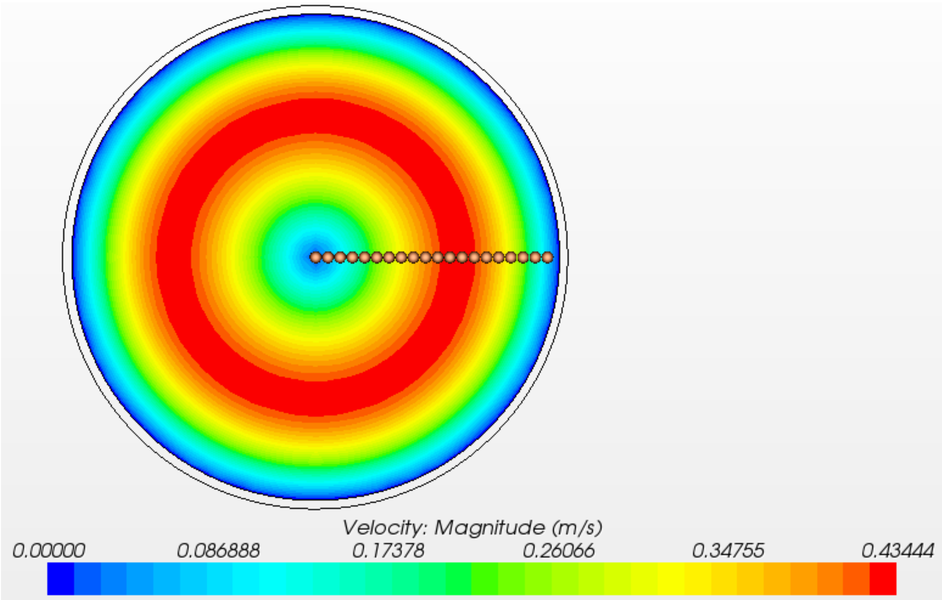

Figure 28: Vortex Spin-down velocity plot (smooth view by changing to *Smooth Filled* the *Contour Style* option in the branch Displayers – scalar 1 of your scalar scene)

It is left as an extra exercise to refine the mesh and compute again the solution, checking convergence to the exact one.

It is left also as exercise the possibility to generate a circular type mesh, which is aligned with the exterior boundary of the problem.

Tips:

1. This geometry has one plane in the Z=0 plane. Therefore, no splitting is needed when converting the mesh to 2D.
2. When converting to 2D, do not erase neither the 3D region nor the 3D mesh.
3. Field functions for the initial velocity: create two field functions (scalar) called *VX0* and *VY0*, and define them as:

$$VX0 = -1 \cdot \$\$Centroid[1]$$
$$VY0 = 1 \cdot \$\$Centroid[0]$$

Check that the units of the field functions correspond to the velocity ones.

4. Use these field functions as initial conditions for the velocity field (see figure 29)

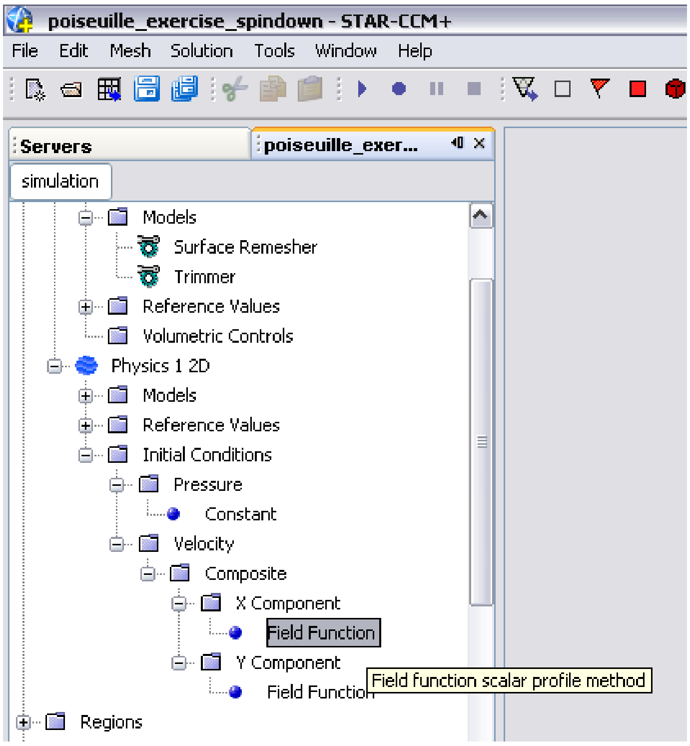

Figure 29: Vortex Spin-down initial velocity setup with FFs

4 Two-Vortex

In this exercise the evolution of a co-rotating vortex couple will be studied. The computational domain will be a scaled version of the one used for the vortex spin down, where the scale factor is 40.

Region → Shell, RC → Transform → Scale → 40

Use the real values of water for the density and the dynamic viscosity.

Run the simulation until time = 200 s, with a time step of 0.1s.

Figure 30: Airplane vortex.

Tips:

1. This geometry has one plane in Z=0 plane. Therefore, no splitting is needed when converting the mesh to 2D.
2. Create a set of field functions (scalar) to simulate two vortices as shown in figure 31. The field functions are defined as follow:

$$ar = 5$$

$$al = -5$$

$$Q = 5$$

$$rr = sqrt\left(pow\left((\$\$Centroid[0] - \$ar), 2\right) + pow(\$\$Centroid[1], 2)\right)$$

$$rl = sqrt\left(pow\left((\$\$Centroid[0] - \$al), 2\right) + pow(\$\$Centroid[1], 2)\right)$$

The next equations are for the right side:

$$Vxr = -\$Q * \$\$Centroid[1] * \left(1 - \exp\left(-pow(\$rr, 2)\right)\right) / \left(pow(\$rr, 2)\right)$$

$$Vyr = \$Q * (\$\$Centroid[0] - \$ar) * \left(1 - \exp\left(-pow(\$rr, 2)\right)\right) / \left(pow(\$rr, 2)\right)$$

The next equations are for the left side:

$$Vxl = -\$Q * \$\$Centroid[1] * \left(1 - \exp\left(-pow(\$rl, 2)\right)\right) / \left(pow(\$rl, 2)\right)$$

$$Vyl = \$Q * (\$\$Centroid[0] - \$al) * \left(1 - \exp\left(-pow(\$rl, 2)\right)\right) / \left(pow(\$rl, 2)\right)$$

Figure 31: Wing Vortex.

The total velocity for X and Y components are:

$$Vxtot = \$Vxr + \$Vxl$$

$$Vytot = \$Vyr + \$Vyl$$

Remember, rename the field functions and write the same name in the *Function Name* field. Check the units of the field function.

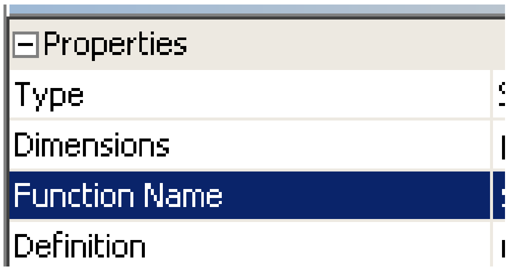

Figure 32: Field Function properties.

3. Use these field functions *Vxtot* and *Vytot* as initial conditions for the velocity field (see figure 29)

5 Lid-driven cavity

The lid-driven cavity problem has long been used as a test or validation case for new codes or new solution methods. The geometry of the problem is simple and in 2D with simple BC's. The standard case is a fluid contained square domain with Dirichlet (no-slip) BC's on all sides, with three stationary sides and one top moving side (figure 33). Take *H = 1m* and *V = 1m/s*. The top side velocity is defined as in the Couette case, and the geometric data can be imported from file *lid_driven_cavity_exercise.igs.*

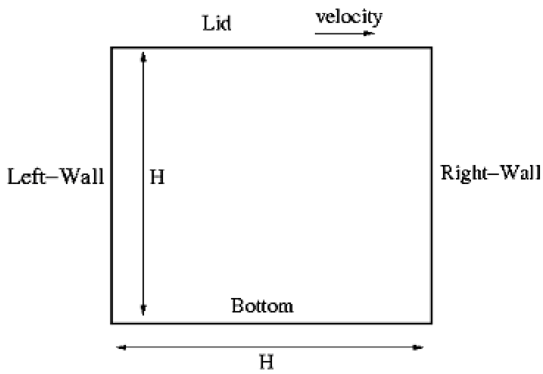

Figure 33: Lid Driven Cavity setup.

Let's take the Reynolds number as:

$$\mathrm{Re} = \frac{V \cdot H}{\nu} = 100 \quad \rightarrow \nu = 0.01 \mathrm{m}^2/\mathrm{s}$$

Set density as 1, and dynamic viscosity as 0.01. For this *Re* number, Ghia et al, 1982, devised a reference steady state solution for this problem (figure 34). Provide your own and perform a qualitative comparison.

Tips:

1. This geometry has one plane in Z=0 plane. Therefore, no splitting is needed when converting the mesh to 2D.

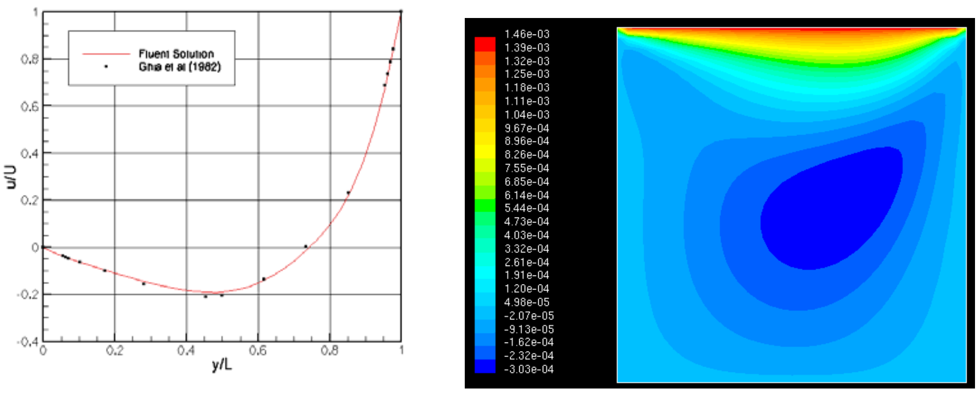

Figure 34: Lid Driven Cavity X velocity, Re=100, [Ghia et al, 1982] and Fluent solution [CFD Online]

6 Circular Couette.

An interesting exact solution of the NS equations is the the steady state flow between two rotating concentric cylinders (figure 35). Where R_1 *= 0.2m*, R_2 *= 0.5m*, Ω_1 *= 5 rad/s* and Ω_2 *= 10 rad/s*. We assume that the pressure gradient is zero and the only forcing is due to the movement of the cylinders. The geometry corresponding to this exercise can be imported from file *circular_couette_exercise.igs*.

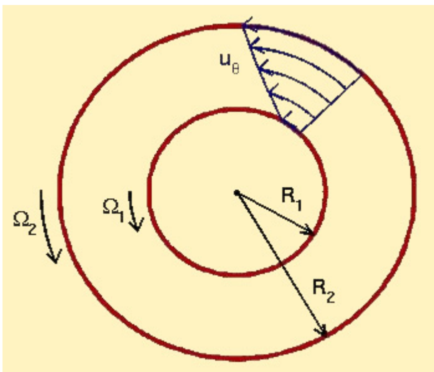

Figure 35: Circular Couette

The steady state solution of this problem is a linear velocity profile:

$$u_\theta = Ar + \frac{B}{r}$$

$$A = \frac{\Omega_2 R_2^2 - \Omega_1 R_1^2}{R_2^2 - R_1^2},\ B = \frac{(\Omega_1 - \Omega_2)\, R_1^2 R_2^2}{R_2^2 - R_1^2}$$

Set density as 1, and dynamic viscosity as 0.01

Tips:
1. The new terms to incorporate into our project can be introduced activating the following options.

Regions → CYLINDER 2D→ Boundaries → Ext_Cylinder→ Physics Conditions → Tangential Velocity → Method → Rotation Rate
Regions → CYLINDER 2D→ Boundaries → Ext_Cylinder → Physics Values → Wall rotation → Value → 10
Regions → CYLINDER 2D→ Boundaries → Int_Cylinder→ Physics Conditions → Tangential Velocity → Method → Rotation Rate

Regions → CYLINDER 2D→ Boundaries → Int_Cylinder → Physics Values → Wall rotation → Value → 5

9 REFERENCES

[Batchelor, 1967] G. K. Batchelor, An Introduction to Fluid Dynamics. (Cambridge Univ. Press, Cambridge, UK, 1967)

[CD, 2010] CD-ADAPCO, 2010, UserGuide_5.04 of StarCCM+.

[CFD Online] http://www.cfd-online.com/Wiki/Lid-driven_cavity_problem

[Ghia et al, 1982] Ghia, Ghia, and Shin (1982), High Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, Journal of Computational Physics, Vol. 48, pp. 387-411.

[Monaghan, 2005] Monaghan, J., 2005, Smoothed particle hydrodynamic simulations of shear flow, Monthly Notices of the Royal Astronomical Society, Vol. 365 (2005), pp. 199-213.

[Morris, 1997] Joseph P. Morris, Patrick J. Fox, and Yi Zhu, Modeling Low Reynolds Number Incompressible Flows Using SPH. JOURNAL OF COMPUTATIONAL PHYSICS 136, 214–226 (1997)

[Sigalotti et al, 2003] Leonardo Di G. Sigalotti, Jaime Klapp, Eloy Sira, Yasmin Meleán, Anwar Hasmy. SPH simulations of time-dependent Poiseuille flow at low Reynolds numbers. Journal of Computational Physics 191 (2003) 622–638

[Urroz, 2004] Gilberto E. Urroz . Numerical solution to unsteady two-dimensional Poiseuille flow, November 2004 (donwloadble from USU OCW – google it).

CFD WORKSHOP

UPM 41 ATHENS COURSE

STAR CCM+ TUTORIAL Number 2

PLANE POISEUILLE WITH A TURBULENCE MODEL

CONTENTS

1 NOTATION

2 RELATED DOCUMENTS

Tutorial **1** (Poiseuille flow)

3 DESCRIPTION OF THE PHYSICAL PROBLEM

Using dynamical systems theory, it has been shown [Fortin&Jardak, 1994] that a Poiseuille flow is potentially unstable for Re>2645, when flow can separate into a pattern of recirculation bubbles on the sides of the channel and a weaving central jet (figure 1). We will take Re=3430. The setup will be the same as in tutorial 1, but a k-ε turbulence model will be introduced in order to simulate this flow and compare the turbulent velocity profile with the laminar solution.

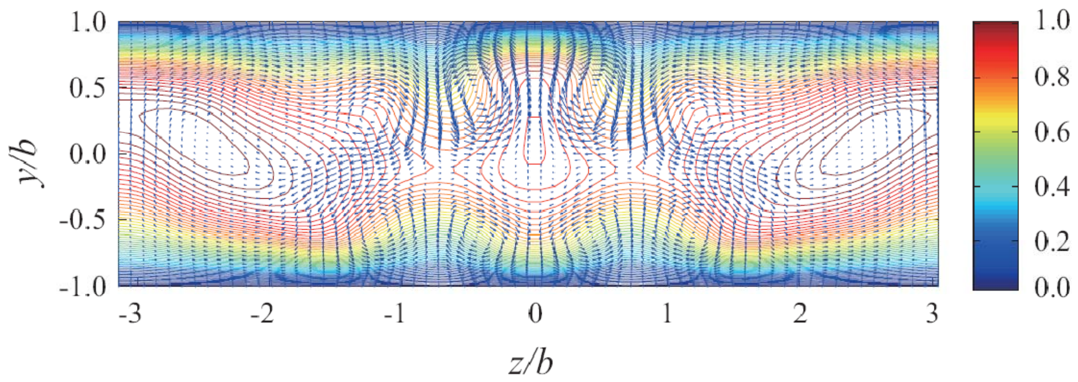

Figure 1: Transition state in Poiseuille flow, Re=1100, from [Tanaka et al, 2008]

4 PROBLEM SETUP

The Reynolds number is defined taking the channel width as characteristic length and the maximum velocity of the laminar solution as characteristic velocity:

$$\mathrm{Re} = \frac{u_{max} \cdot 2b}{\mu / \rho};$$

The pressure gradient K will be taken as 1E4.

$$u_{max} = \frac{K \cdot b^2}{2\mu} = 1E4 \cdot 0.5 \cdot 0.7^2 = 2450\ m/s$$

The initial condition in velocity will be the steady state solution.

5 TOPICS COVERED

1. Including a turbulence model in Star CCM+ Physics.
2. Setting up initial conditions for the velocity field.
3. Creating an exact solution curve for setting up initial conditions for the velocity field.

6 SIMULATION SETUP AND SOLUTION

6.1 Physics Model

The procedure up until this point is the same as the one presented in the laminar plane Poiseuille (tutorial **1**). We will depart from the simulation file corresponding to the Poiseuille case and modify the Physics of the 2D simulation. The physical model now incorporates a turbulence model which will be taken as the *k-ε*.

Continua → Physics 1 2D→ Models → DC → Implicit Unsteady, Liquid, Segregated Flow, Constant Density, Turbulent, K-Epsilon Turbulence → Close

We must create a field function for the initial velocity of the flow, defined as the maximum velocity for the laminar solution.

Tools → Field Function, RC → new
Tools → Field Function → User Field Function 1, F2 → Laminar_Velocity
Tools → Field Function → Velocity_Laminar → Type: Scalar
Tools → Field Function → Velocity_Laminar → Function Name: Laminar_Velocity
Tools → Field Function → Velocity_Laminar → Dimensions: Velocity
Tools → Field Function → Velocity_Laminar → Definition:

1e4/(2*$DynamicViscosity)*(pow(0.7,2)-pow($$Centroid[1],2))

Continua→ Physics 1 2D → Initial Conditions → Velocity → Method → Composite
Continua→ Physics 1 2D → Initial Conditions → Velocity → Composite → X Component → Method → Field Function
Continua→ Physics 1 2D → Initial Conditions → Velocity → Composite → X Component → Field Function → Velocity_Laminar

The rest of steps are the same but the momentum source is taken as:

Regions→ CHANNEL 2D → Physics Values→ Momentum Source→ Constant→ Value→ [1e4, 0.0, 0.0]

6.2 Solver

Solvers → Implicit Unsteady → Time-step → 1e-4s
Stopping Criteria → Maximum Inner Iterations → 5
Stopping Criteria → Maximum Physical Time → 10s

Solution → Clear Solution

Solution → Run ( )

7 POST-PROCESSING EXTRAS

We will learn how to use the field function **Velocity_Laminar** to compare, in the same plot, the calculated turbulent solution by STAR-CCM+ and the laminar one.

Tools → Tables, RC → New Table → XYZ Internal Table
Tools → Tables → XYZ Internal Table, F2 → Laminar_Solution
Tools → Tables → Laminar_Solution → Parts → [ ] → x-velocity probe → Ok
Tools → Tables → Laminar_Solution → Scalars → [ ] → Velocity_Laminar

As in the tutorial 1 section 7, a plot is now created for the horizontal velocity profile, or the one used there is re-utilized.

Tabulated data of the laminar solution is now added to this plot.

Plots → Vx → Tabular, RC → New Tabular Data Set
Plots → Vx → Tabular → tabular, F2 → Laminar_Velocity

Plots → Vx → Tabular → Laminar_Velocity → Table → Laminar_Solution
Plots → Vx → Tabular → Laminar_Velocity → X Column → Laminar_Velocity
Plots → Vx → Tabular → Laminar_Velocity → Y Column → Y (as shown in figure 2)

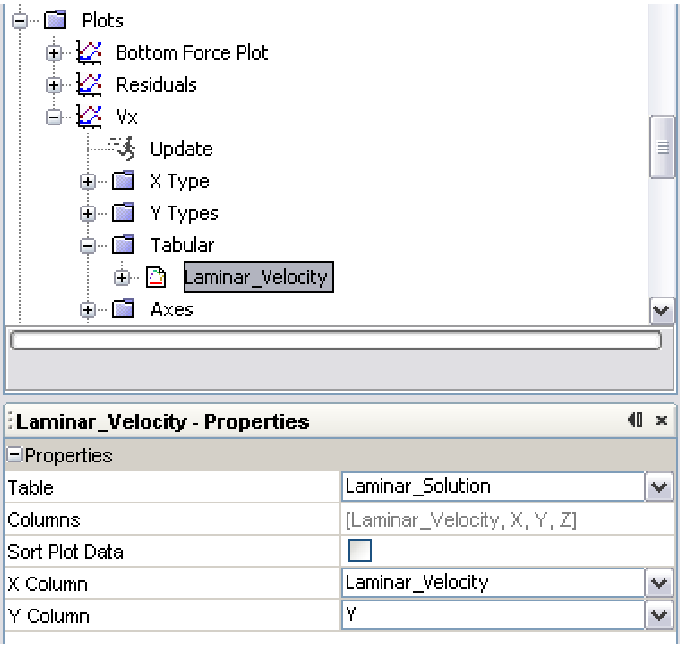

Figure 2: Tabular properties setup

Plots → Vx → Tabular → Laminar_Velocity → + → Line Style → Style → none
Plots → Vx → Tabular → Laminar_Velocity → + → Symbol Style → Color → blue
Plots → Vx → Tabular → Laminar_Velocity → + → Symbol Style → Shape → Dot
Plots → Vx, DC

Solution → Clear Solution

Solution → Run ( )

Something like figure 3 should be obtained.

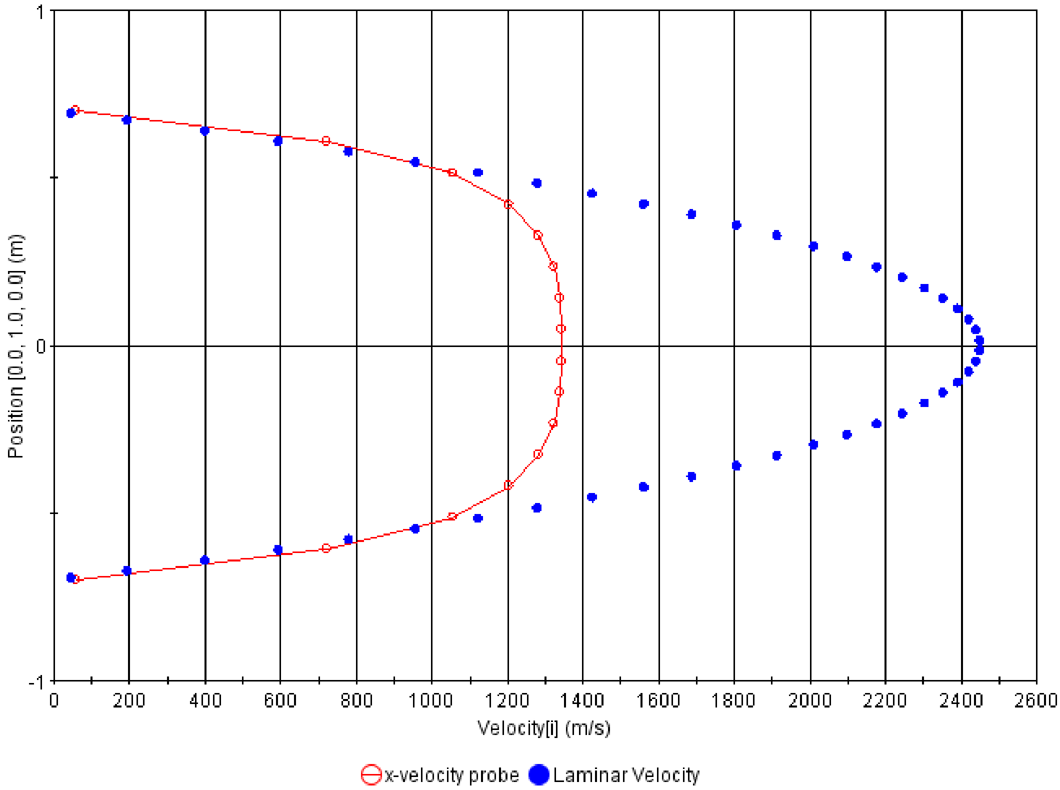

Figure 3: Velocity Profile of the Laminar and Turbulent Solution

8 EXERCISES

1. Export the obtained velocity profile to an ASCII file and compare it with the laminar solution in MATLAB, using the script uy_poiseuille.m.
2. Compare the forces on the plates with the laminar one.

9 REFERENCES

[Fortin&Jardak, 1994] Fortin A., Jardak M., Pierre R., Gervais J.J., Old and New Results on the Two-Dimensional Poiseuille Flow, Journal for Computational Physics, Vol. 115, No. 2, pp.455-469, (1994).

[Siddique&Naser, 2002] Siddique, H, Naser, J., 2002, A 2D CFD code for teaching, combined SPH-experimental systematic approach to resonant free surface movements. International Conference on Computational Fluid Dynamics 2, Sydney, July 15-19, 2002.

[Tanaka et al, 2008] Tanaka, Y., Yoshino, M., Matsubara, M., Aota, N., Direct Numerical Simulation of

Reverse Transition from Turbulence in Plane Poiseuille Flow. 8th. World Congress on Computational Mechanics (WCCM8), Venice, Italy, June 30 – July 5, 2008.

CFD WORKSHOP

UPM 41 ATHENS COURSE

STAR CCM+ TUTORIAL Number 3

CYLINDER

CONTENTS

1 NOTATION

VS VOLUME SHAPES

2 RELATED DOCUMENTS

Tutorial **1** (Poiseuille flow)
Tutorial **2** (Poiseuille flow with a turbulence model)

3 DESCRIPTION OF THE PHYSICAL PROBLEM

A flow past a circular cylinder is one of the classical problems of Fluid Mechanics. The geometry suggests a steady and symmetric flow pattern for low value of Reynolds number. Any disturbance introduced at the inlet gets damped by the viscous forces. As the Reynolds number is increased, the disturbance in the upstream flow may not be damped. This leads to a very important periodic phenomenon downstream of the cylinder, known as "vortex shedding" (figure 1). The vortices become stronger and larger with increases in the Re number. This arrangement becomes unstable beyond a certain critical Reynolds numbers (Re ~ 47) and von Karman vortex shedding takes place. At this point the flow is still two-dimensional and laminar.

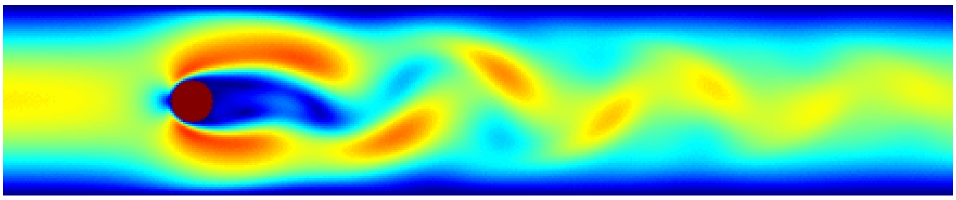

Figure 1: Vortex shedding behind a circular cylinder

4 PROBLEM SETUP

The cylinder of diameter *D=1*m resides in a rectangular computational domain whose upstream and downstream boundaries are located at 3.5D and 10.5D from the center of the cylinder, respectively. The upper and lower boundaries are placed at 3D, each, from the cylinder. A schematic view of the domain is shown in Figure 2 and the geometrical data can be imported from the file *Cylinder.igs*.

No-slip BC is set for the velocity on the cylinder surface while free-stream values (velocity Inlet BC) are assigned for the velocity at the upstream boundary. At the downstream boundary a "Pressure Outlet" BC is imposed, which is equivalent, we believe, to a Neumann BC on the velocity. On the upper and lower surface boundaries a free-slip BC (zero normal velocity) is imposed.

The Reynolds number is based on the diameter of the cylinder, free-stream velocity and viscosity of the fluid.

$$\mathrm{Re} = \frac{U \cdot D \cdot \rho}{\mu}$$

The Reynolds Number Re will be taken as 100, μ=1 Pa.s and ρ=1 kg/m3, therefore the value of U is defined as:

$$U = \frac{\mathrm{Re} \cdot \mu}{D \cdot \rho} = \frac{100 \cdot 1}{1 \cdot 1} = 100\ m/s$$

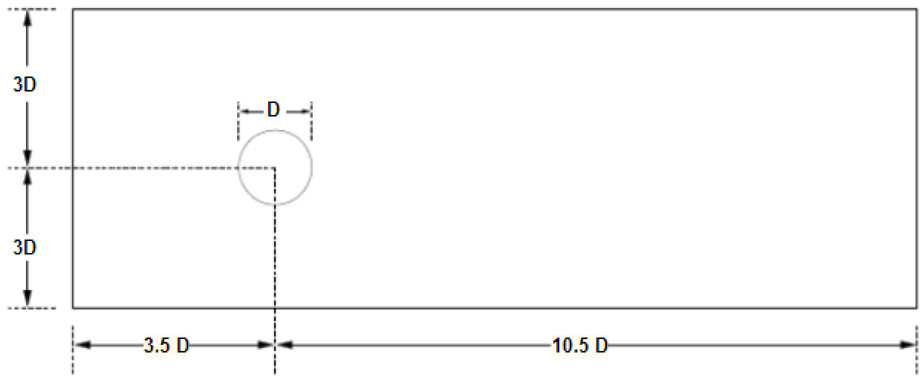

Figure 2: Flow past cylinder: schematic of the computational domain for the flow analysis

5 TOPICS COVERED

This is the second tutorial of this series. We will rely on the previous ones for most of the simulation steps. A list of specific topics covered with this tutorial follows:

1. Blocks for mesh refinement.
2. Work with a refined Mesh.
3. Free slip BC.

6 SETUP AND SOLUTION

6.1 General

The geometrical information resides in the file *Cylinder.igs*. The importing phase is analogous to section 6.2 of tutorial 1.

6.2 Preparing Surfaces

We must now change the default name used for the whole geometry like in the Poiseuille tutorial, but in this problem we have a new boundary that we will call *Cylinder*. After "splitting by angle" the boundaries let's rename them as shown in figure 3.

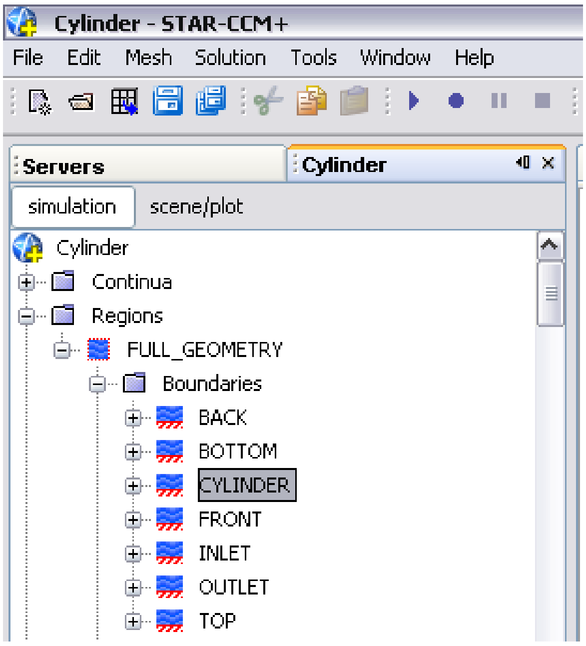

Figure 3: Simulation Tree after renaming the region and its boundaries.

The boundary conditions are given at the inlet by prescribing an incoming velocity (*Velocity Inlet)*. At the outlet, far downstream, a *Pressure Outlet BC* is imposed.

6.3 Physics Models

Now, we must define our physical model and the initial conditions. If our reference system is fixed, the Stationary option should be selected.

Continua, RC → New → Physics Continuum

Continua → Physics 1 → Models, DC → Three Dimensional, Implicit Unsteady, Liquid, Segregated Flow, Constant Density and Laminar → Close

Continua → Physics 1 → Initial Conditions → Velocity → Constant → [100, 0.0, 0.0] m/s

The liquid properties are tuned so that the convergence to the steady state solution is speeded up like in section 6.4 of tutorial 1. Set density as 1, and dynamic viscosity as 1.

Now, Free-Slip BCs are applied to the top and bottom walls and a No-slip BC at the cylinder surface.

Regions → FULL_GEOMETRY → Boundaries → INLET → Type → Velocity Inlet

Regions → FULL_GEOMETRY → Boundaries → OUTLET → Type → Pressure Outlet

Regions → FULL_GEOMETRY → CYLINDER → Physics Conditions → Shear Stress Specification → no-slip

Regions → FULL_GEOMETRY → TOP → Physics Conditions → Shear Stress Specification → slip

Regions → FULL_GEOMETRY → BOTTOM → Physics Conditions → Shear Stress Specification → slip

Regions → FULL_GEOMETRY → INLET → Physics Values → Velocity Magnitude → Constant → 100 m/s

File → Save

6.4 Preparing the Mesh

For this tutorial we are going to use a method in order to refine the mesh in specific areas of the geometry using *Volume Shapes – VS. M*esh blocks will are used to refine the mesh at specific locations. As usual, *Surface Remesher* model is used to mesh de boundaries and and *Trimmer Meshing* model for the volumes.

Continua → Mesh 1 → Models, DC → Surface Remesher, Trimmer → Close
Continua → Mesh 1 → Reference Values → Base Size = 1m
Continua → Mesh 1 → Reference Values → Surface Size → Relative Minimum Size = 100
Continua → Mesh 1 → Reference Values → Surface Size → Relative Target Size = 100

Now, we must create a *Block* (VS see Figure 4) to refinethe specific area where the vortices were developed.

Tools → Volume Shapes, RC → New Shape → Block

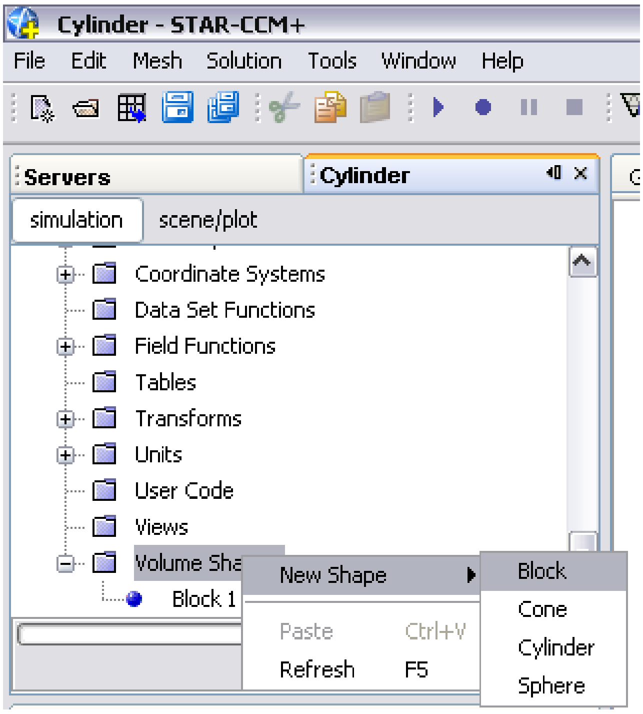

Figure 4: Creating Volume Shapes

After that, the edit window appears and we must edit the maximum and minimum coordinates as shown in figure 5.

Step 1 **Corner 1→ X = 0m**
Y = -2m
Z = 0m
Corner 2→ X = 10.5m
Y = 2m
Z = 0.1m

Step 2 **Create**

Step 3 **Close**

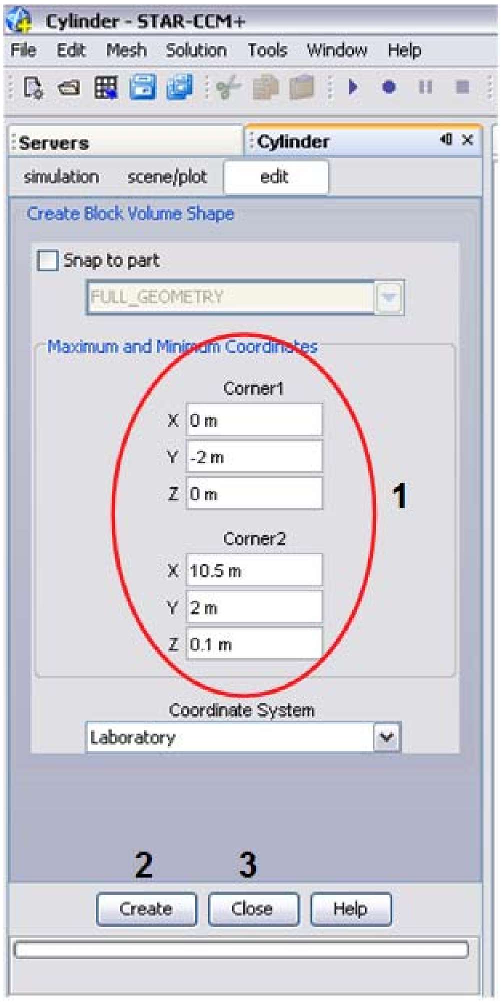

Figure 5: Volume Shapes Setup

Once the VS has been created, the next step is to setup its mesh parameters following the steps below:

Continua → Mesh 1 → Volumetric Controls, RC → New

Continua → Mesh 1 → Volumetric Controls → Volumetric Control 1 → Shapes (as in figure 6) **→ [ ] → Select the Block 1 → Ok**

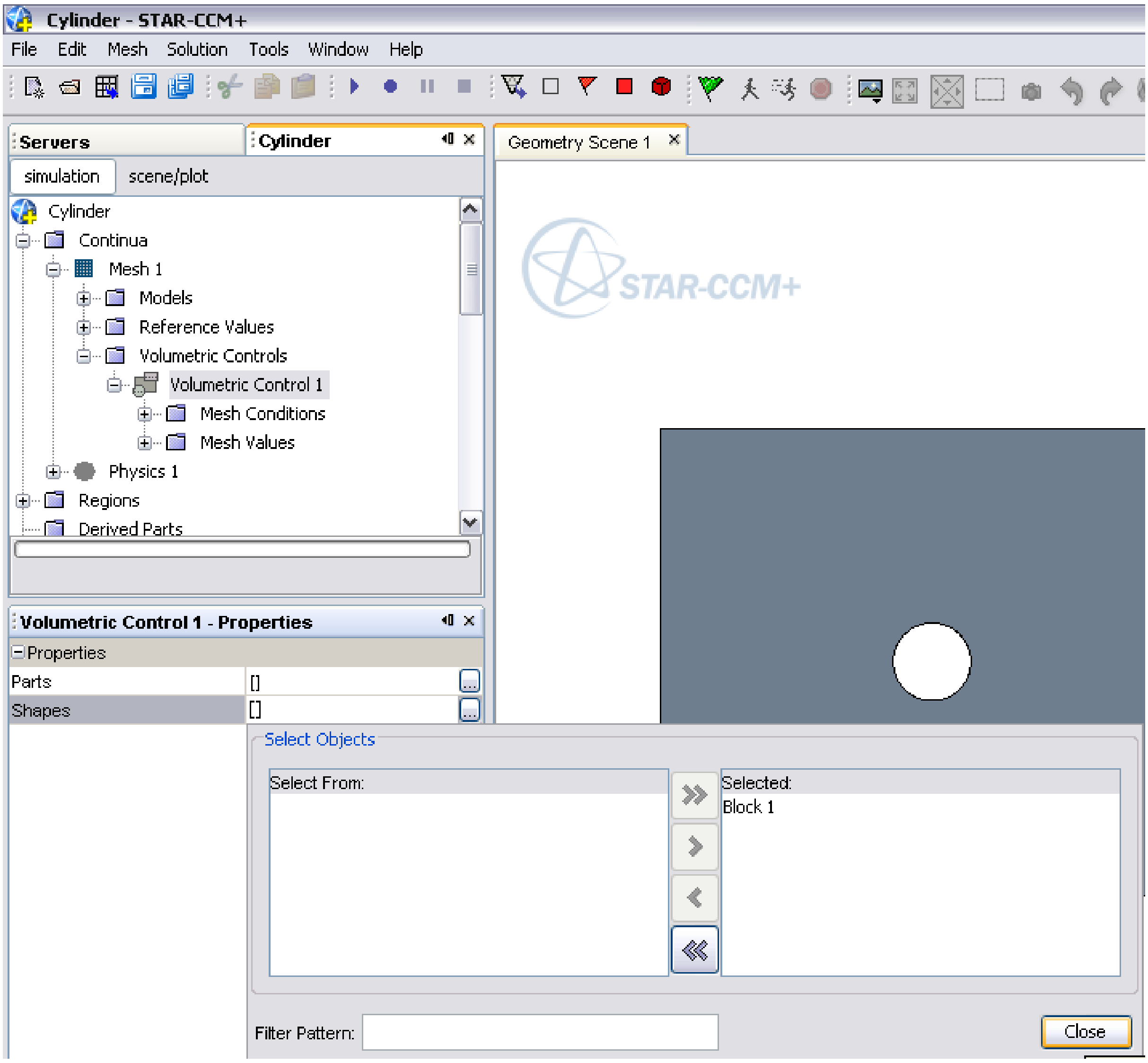

Figure 6: Volume Shapes Setup

Continua → Mesh 1 → Volumetric Controls → Volumetric Control 1 → Mesh Conditions → Trimmer → Customize anisotropic size → Enable by clicking the box

Continua → Mesh 1 → Volumetric Controls → Volumetric Control 1 → Mesh Values → Trimmer Anisotropic Size → Custom X Size → Enable by clicking the box

Continua → Mesh 1 → Volumetric Controls → Volumetric Control 1 → Mesh Values → Trimmer Anisotropic Size → Custom Y Size → Enable by clicking the box

Continua → Mesh 1 → Volumetric Controls → Volumetric Control 1 → Mesh Values → Trimmer Anisotropic Size → Relative X Size → Percentage of base → 10
Continua → Mesh 1 → Volumetric Controls → Volumetric Control 1 → Mesh Values → Trimmer Anisotropic Size → Relative Y Size → Percentage of base → 10

We then configure the mesh around the cylinder. To create the refined mesh:

Regions → FULL_GEOMETRY → Boundaries → CYLINDER → Mesh Condition → Custom Surface Size → Enable by clicking the box

Regions → FULL_GEOMETRY → Boundaries → CYLINDER → Mesh Values → Surface Size → Relative Minimum Size = 1

Regions → FULL_GEOMETRY → Boundaries → CYLINDER → Mesh Values → Surface Size → Relative Target Size → 1

We now generate the mesh (section 6.5.2 tutorial 1) and convert it to 2D (section 6.5.4)

Now, we create a Report (section 7 tutorial 1) to monitor the lift produced on the cylinder.

Report, RC→ New Report → Force Coefficient (figure 7)

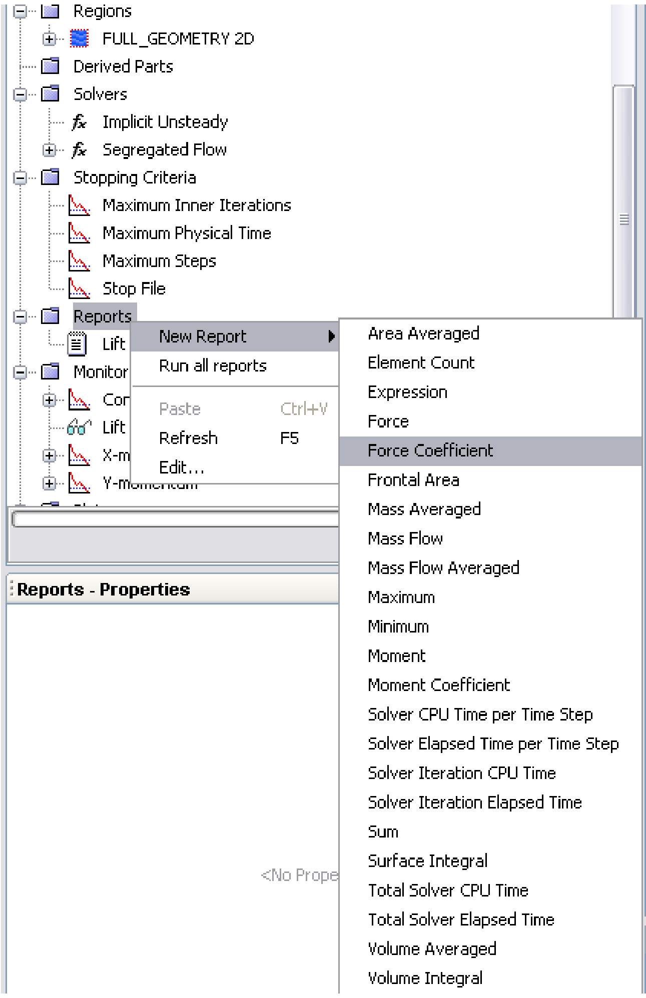

Figure 7: Adding Force Coefficient Report. Lift Coefficient Report

Report → Force Coefficient Monitor, F2 → Lift_Coefficient

Report → Lift Coefficient → and set the property values as shown in figure 8.

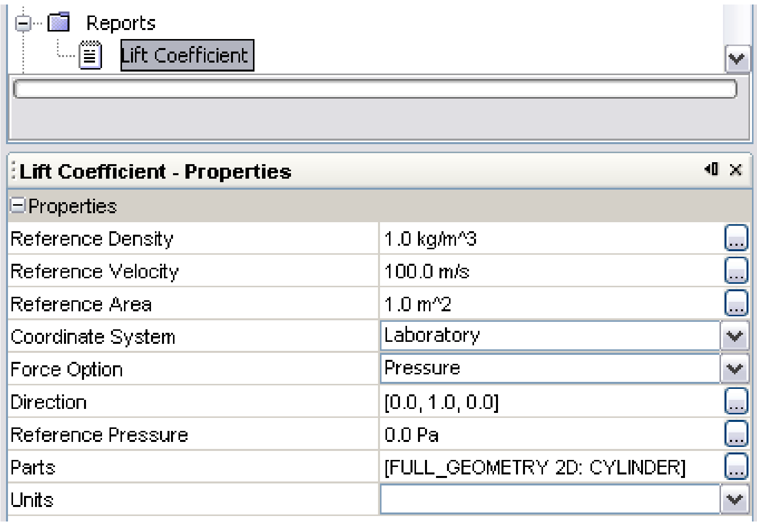

Figure 8: Lift Coefficient Report Properties

Report → Lift Coefficient, RC → Create Monitor and Plot from Report

We change the *Trigger* option of the monitor to adjust the plot by *Time-Step*

Monitors → Lift Coefficient Monitor → Trigger → Time Step

6.5 Solver

Solvers → Implicit Unsteady → 0.001

Stopping Criteria → Maximum Inner Iterations → 3

Stopping Criteria → Maximum Physical Time → 2

Scenes, RC New Scene Scalar

In the scene, RC over the blue line

Velocity→ Magnitude

Solution → Run ( )

Run the simulation until you consider that a steady state solution has been found, obtained when only negligible variations in the values of the velocity are present. An image such as the one in figure 9 should be obtained.

Compare the period of the lift coefficient curve with the Strouhal number for the Reynolds = 100 (should be around 0.16, see e.g. [WILLIAMSON, 1996]). The Strouhal number S is defined as

$$S = \frac{f\,D}{U}$$

with f being the frequency of the shedding vortex and D and U the diameter and the free stream velocity respectively.

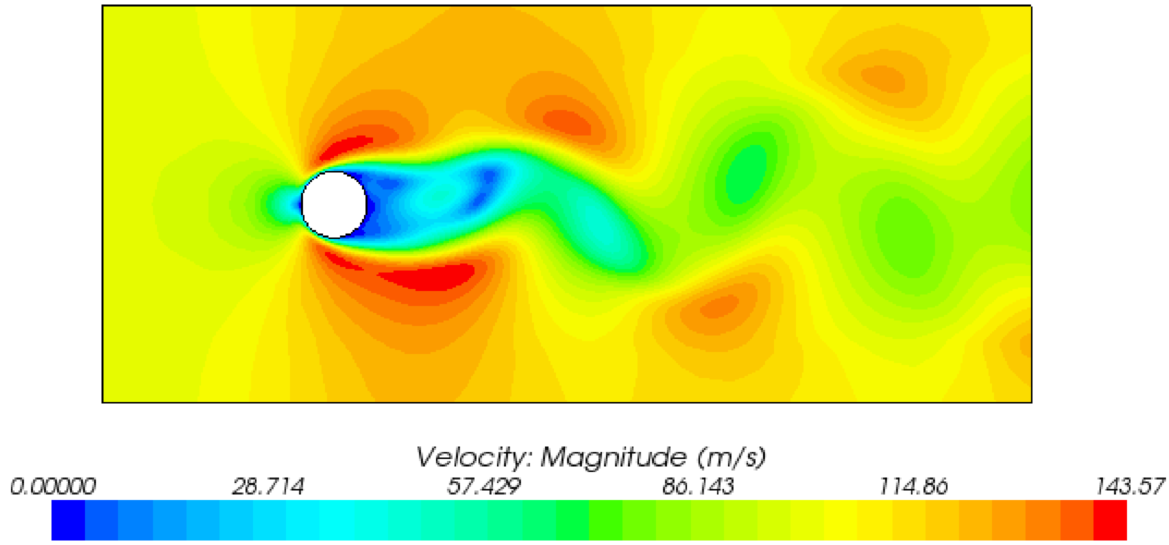

Figure 9: Flow around the cylinder

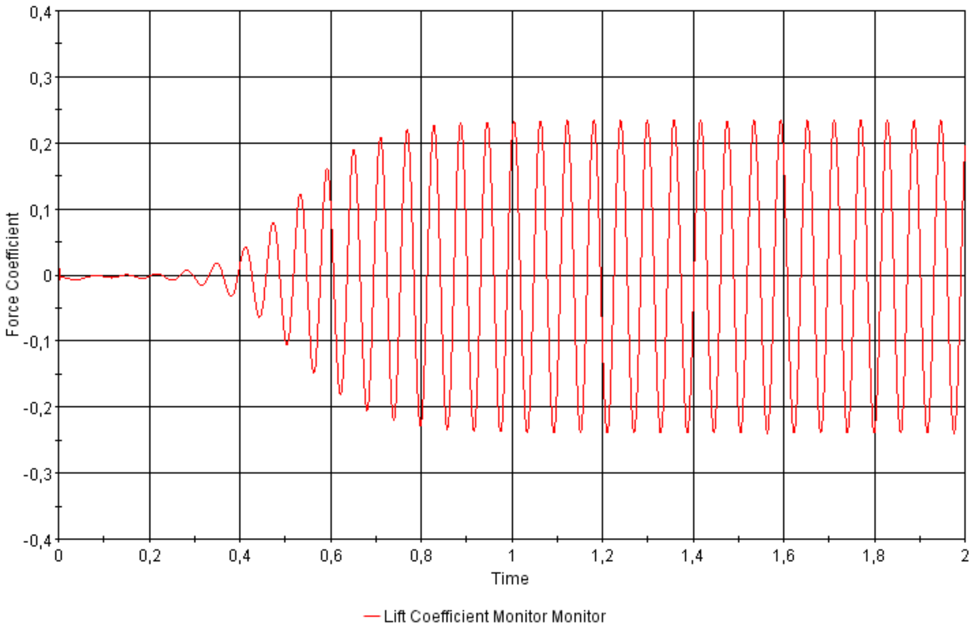

Figure 10: Lift Coefficient Monitor Plot

7 EXERCISE

1 Modify the previous example so that the Drag force coefficient (x-component of the force) is also monitored and drawn in a new plot (C_D based on the diameter should be around 1.39, see, e.g., [LIMA ET AL., 2003])

2 Flow past a cylinder with a Rotation Rate (Magnus effect – figure 11).

In this exercise, we are going to make a simple change in our basic problem which consists in assigning a boundary condition to the cylinder defined as the possibility of rotation "Rotation Rate".

To solve this exercise, we will follow analogous steps as with the cylinder, with the following tips:

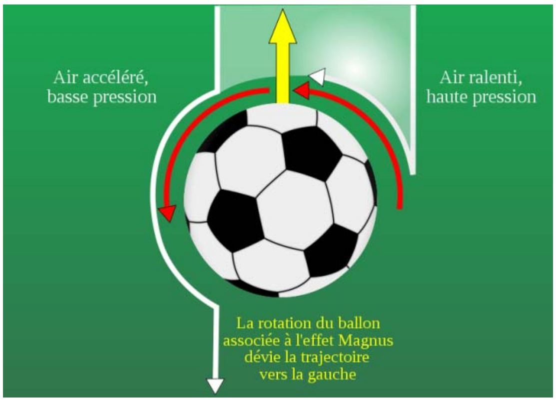

Figure 11: Magnus Effect

Regions → FULL_GEOMETRY 2D → Boundaries → CYLINDER → Physics Conditions → Tangential Velocity Specification → Method: Rotation Rate, and Reference Frame: Absolutes

Regions → FULL_GEOMETRY 2D → Boundaries → CYLINDER → Physics Values → Wall Rotation → Constant → Value → 100 rad/s

Once implemented, the changes should produce something like figure 12.

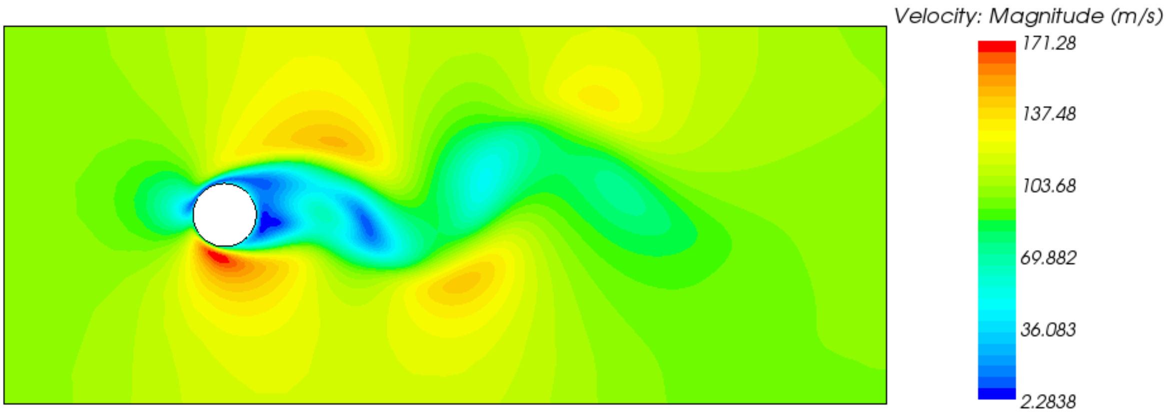

Figure 12: Flow around the cylinder with a rotation rate in the cylinder

Repeat the tutorial and compare the lift coefficient solutions to find that there is a net component of the lift in this case.

8 REFERENCES

[WILLIAMSON, 1996] Williamson, C.H.K., 1996, Vortex Dynamics in the cylinder wake, Annual Review of Fluid Mechanics.

[LIMA ET AL., 2003] A.L.F. Lima E Silva a, A. Silveira-Neto a,*, J.J.R. Damasceno, 2033, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, Journal of Computational Physics, 189, 351-370

CFD WORKSHOP

UPM 41 ATHENS COURSE

STAR CCM+ TUTORIAL Number 4

DAM BREAK

CONTENTS

1 NOTATION

VOF Volume of Fluid

2 RELATED DOCUMENTS

Tutorial 1 (Poiseuille flow)
Tutorial 3 (Cylinder)

3 DESCRIPTION OF THE PHYSICAL PROBLEM

A schematic drawing of the dam break model is presented in figure 1. In this problem, a rectangular column of water, in hydrostatic equilibrium, is confined between two walls. Gravity is acting downwards with a magnitude of 9.81 m/s^2. At the beginning of the calculation, the right wall is removed and the water is allowed to flow out along the horizontal wall.

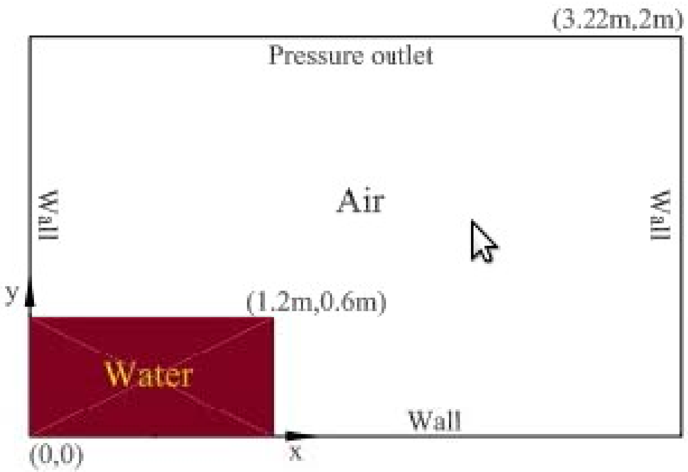

Figure 1: General layout of the dam break problem and boundary conditions. [Abdolmaleki et al, 2004]

In this model, the tank size is L = 3.22 m and H = 2.0 m. A column of water (***l*** = 1.2 m and ***h*** = 0.6 m) is located on the left side of the tank. For impact pressure measurements on the downstream wall, as in the experiments, a sensor is placed on the wall(3.22 m, 0.16 m). Free-surface elevations are recorded at stations h1 and h2 at distances x1 = 2.725 m and x2 = 2.228 m from the origin (bottom of left side wall).

4 TOPICS COVERED

This tutorial examines the dam-break problem using the Volume of Fluid (VOF) multiphase model. The new topics covered in the present tutorial are:

- Set up an incompressible multiphase problem.
- Solve the interface dynamics using the VOF technique (free surface problem).
- Group (combine) boundaries.
- Generate animation movies.

5 SETUP AND SOLUTION

5.1 Preparing Surfaces

File → New Simulation → OK
File → Import Surface→ …. /Dam_Break.igs → Open *(Tessellation Density: Fine)* **→ OK**

Regions → Shell 1, F2 → DAM_BREAK

Let us rename the surfaces that appear in our geometry

Regions → DAM_BREAK → Boundaries → change default name, F2 → WALLS

Regions → DAM_BREAK → Boundaries → WALLS → RC > Split by angle

Rename top, back and front boundaries. The other boundaries will be grouped in order to simplify the set up. This is only possible because the same BC will be applied to all of them

Regions→ DAM_BREAK→ Boundaries→ Select, pressing Control key, the unnamed boundaries→ RC→ Combine

Rename the new boundary as WALLS

5.2 Physics Models

5.2.1 General

Continua→ RC→ New Physics Continuum→

Now we have to define our physical model and the initial conditions. If our reference system is fixed the Stationary option should be selected.

Continua → Physics 1 → Models, DC→ Three Dimensional, Implicit Unsteady, Multiphase Mixture, Volume of Fluid (VOF), Laminar and Gravity → Close

Continua → Physics 1 → Models → Eulerian Multiphase → Eulerian Phases, RC → new

Continua → Physics 1→ Models → Eulerian Multiphase → Eulerian Phases → Phase 1, F2 → Water

Continua → Physics 1 → Models→ Eulerian Multiphase → Eulerian Phases → Water → Models, DC→ Enable Liquid and Constant Density boxes.

Continua → Physics 1→ Models→ Eulerian Multiphase → Eulerian Phases, RC → new

Continua → Physics 1→ Models→ Eulerian Multiphase → Eulerian Phases→ Phase 1, F2→ Air

Continua → Physics 1→ Models→ Eulerian Multiphase → Eulerian Phases→ Air→ Models, DC→ Gas and Constant Density

5.2.2 Boundary conditions

As aforementioned, on the bottom and side walls a free-slip BC will be imposed.

Regions → DAM_BREAK → Boundaries → WALLS → Physics Conditions → Shear Stress Specification → Slip

A Pressure Outlet condition is considered for the top wall.

Regions→ DAM_BREAK→ Boundaries→ TOP→ Type→ Pressure Outlet

5.2.3 Initial conditions for the VOF function

In order to define which part of the fluid domain corresponds to Air and which part to water in the initial configuration, a Field Function is going to be created.

Tools → Field Functions, RC → New
Tools → Field Functions→ User Field Function 1, F2→ Water_at_time0
Type: Scalar
Function Name: Water_at_time0
Dimensions: Dimensionless
Definition: ($$Centroid[0]<=1.2 && $$Centroid[1]<=0.6)?1.0 : 0.0

Tools→ Field Functions→ Water_at_time0 > RC > Copy
Tools→ Field Functions→ RC > Paste
Tools→ Field Functions→ Copy of Water_at_time0 > F2 > Air_at_time0
Tools→ Field Functions→ Air_at_time0 > Function Name > Air_at_time0
Tools→ Field Function→ Air_at_time0 >Definition > Change the existing by swapping the final 1.0:0.0 to 0.0:1.0

File > Save

With these field functions, we can set up the material composition (air or water) of the physical zones in the geometrical domain at time zero. This is actually the initialization of the VOF function.

Continua → Physics 1 → Initial Conditions → Volume Fraction → Method → Composite
Continua → Physics 1 → Initial Conditions → Volume Fraction → Water → Method → Field Function
Continua → Physics 1 → Initial Conditions → Volume Fraction → Water → Field Function → Water_at_time0
Continua → Physics 1 → Initial Conditions → Volume Fraction → Air → Method → Field Function
Continua → Physics 1 → Initial Conditions → Volume Fraction → Air → Field Function → Air_at_time0

Now, an initial condition in the pressure (hydrostatic pressure in the water zone and 0 for the air zone) has to be defined. For this purpose a field function is specifically created.

Tools → Field Functions, RC → New
Tools → Field Functions → User Field Function 1, F2 → P_at_time0
Type: Scalar
Function Name: P_at_time0
Dimensions: Pressure
Definition: 1000*9.81*(0.6 - $$Centroid[1])*$Water_at_time0

Now, an initial condition for the pressure at time zero for the water phase is set.

Continua → Physics 1 → Initial Conditions → Pressure → Field Function
Continua → Physics 1 → Initial Conditions → Pressure → Field Function → P_at_time0

5.3 Mesh

5.3.1 General

The characteristics of the mesh have to be specified. First, if some regions are going to have a mesh refinement, the option *Surface Remesher* must be selected.Furthermore, since we are going to convert the 3D geometry into a 2D geometry, the *Trimmer* option should be chosen.

Continua → Mesh 1 → Models, DC → Surface Remesher, Trimmer → Close

Let us specify a base size for the mesh; this means that all the relative distances will be referred to this dimensional base size. It is important to check the characteristic size of the body that we are studying and choose reference values of the same order. Use the rule icon if necessary to get an approximate idea of the characteristic size of the imported object.

Continua → Mesh 1 → Reference Values → Base Size = 1.0 m
Continua → Mesh 1 → Reference Values → Maximum cell size → Relative Size = 10
Continua → Mesh 1 → Reference Values → Surface Size → Relative Minimum Size = 100
Continua → Mesh 1 → Reference Values → Surface Size → Relative Target Size = 100

5.3.2 Mesh refinement

Two volume blocks are going to be created adjacent to the bottom and right wall of the domain. The bricks will be meshed with 2.5% of the base size.

Tools → Volume Shapes, RC → New Shape → Block →
Corner 1= [0.0, 0.0, 0.0] and
Corner 2= [3.22, 0.6, 0.1] → Create → Close

Tools → Volume Shapes, RC → New Shape → Block →
Corner1= [2.82, 0.6, 0.0] and
Corner 2= [3.22, 2.0, 0.1] → Create → Close

Select both pressing the Ctrl key in order to watch them both in Geometry Scene 1. We can include these blocks in our mesh definition.

Continua → Mesh → Volumetric Controls, RC → New
Continua → Mesh → Volumetric Controls → Volumetric Control 1 → Shapes→ Block 1 & Block 2 →OK

Continua → Mesh → Volumetric Controls → Volumetric Control 1 → Mesh Conditions → Trimmer → Customize isotropic size → click

Continua → Mesh → Volumetric Controls → Volumetric Control 1 → Mesh Values → Custom Size → Relative size = 2.5

File → Save

Generate mesh (Surface mesh + Volumetric mesh) there is no need to split because the back part of the domain lies on the plane Z=0**. Check this by activating Tools → Coordinate Systems → Laboratory**

Mesh → Convert to 2D

Since we want to work with 2D physics, we can remove the physics from the 3D Region. This way, this region will be ommited in the simulation along with its physics.

Regions > DAM_BREAK > (Below) Physics Continuum > None

File→ Save

Scenes, RC → New Scene → Mesh

You should see something like the figure 2

Check gravity [0, -9.81, 0]

Continua → Physics 1 2D → Reference Values → Gravity

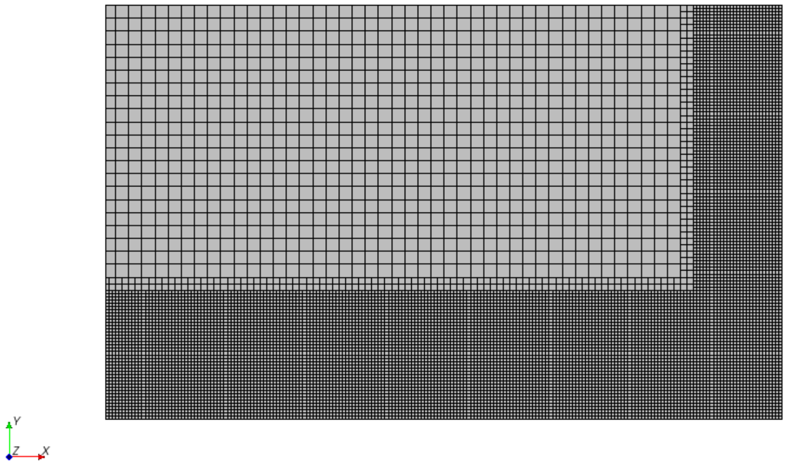

Figure 2: Dam Break Mesh

A scalar scene can now be created in which the VOF function at time 0 can be shown.

Scenes, RC → New Scene → Scalar

On the Scene, RC over the blue horizontal bar → Volume Fraction → Water

Click in the top menu to initialize the VOF and the scene (figure 3).

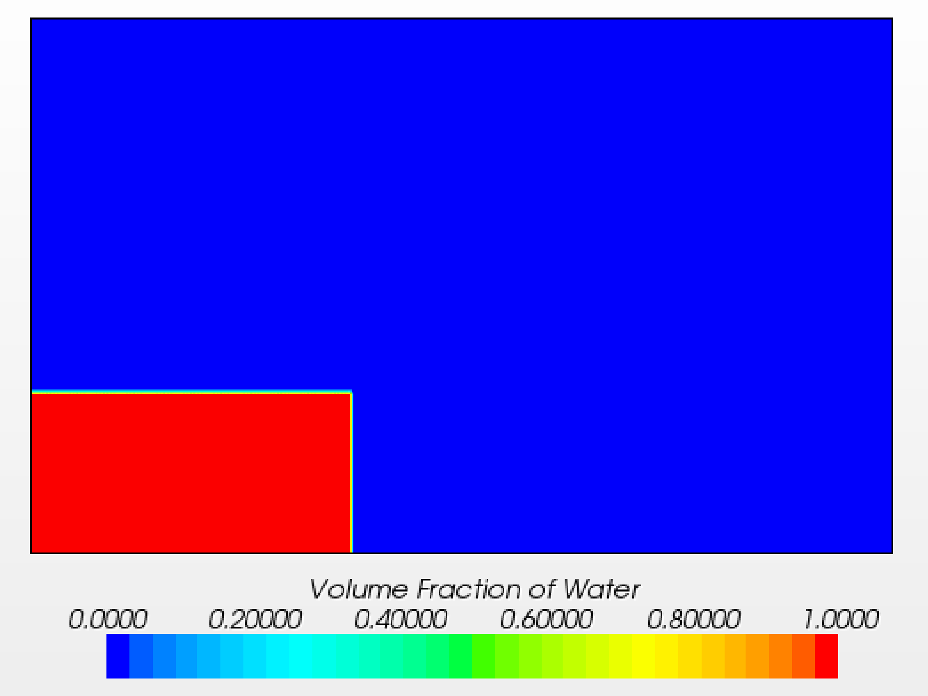

Figure 3: VOF function initial condition

5.4 Solver

Solvers → Implicit Unsteady → 0.002
Stopping Criteria → Maximum Inner Iterations → 3
Stopping Criteria → Maximum Physical Time → 3
Stopping Criteria → Maximum Steps → Enable → unclick

5.5 Run

File → Save
Solution → Run

6 POST-PROCESSING EXTRAS

A pressure sensor is placed in the fluid close to the bottom left corner of the tank. It will basically register the pressure drop due to the reduction of water height at that point (see figure 4). The data for that sensor are:

Derived Parts, RC → New Part → Section → Cylinder
Input Parts → DAM_BREAK 2D
Origin → [0.1, 0.1, 0.0]

Orientation → [0.0, 0.0, 1.0]
Radius →0.02 m → Create → Close

and are introduced using the menu of figure 5.

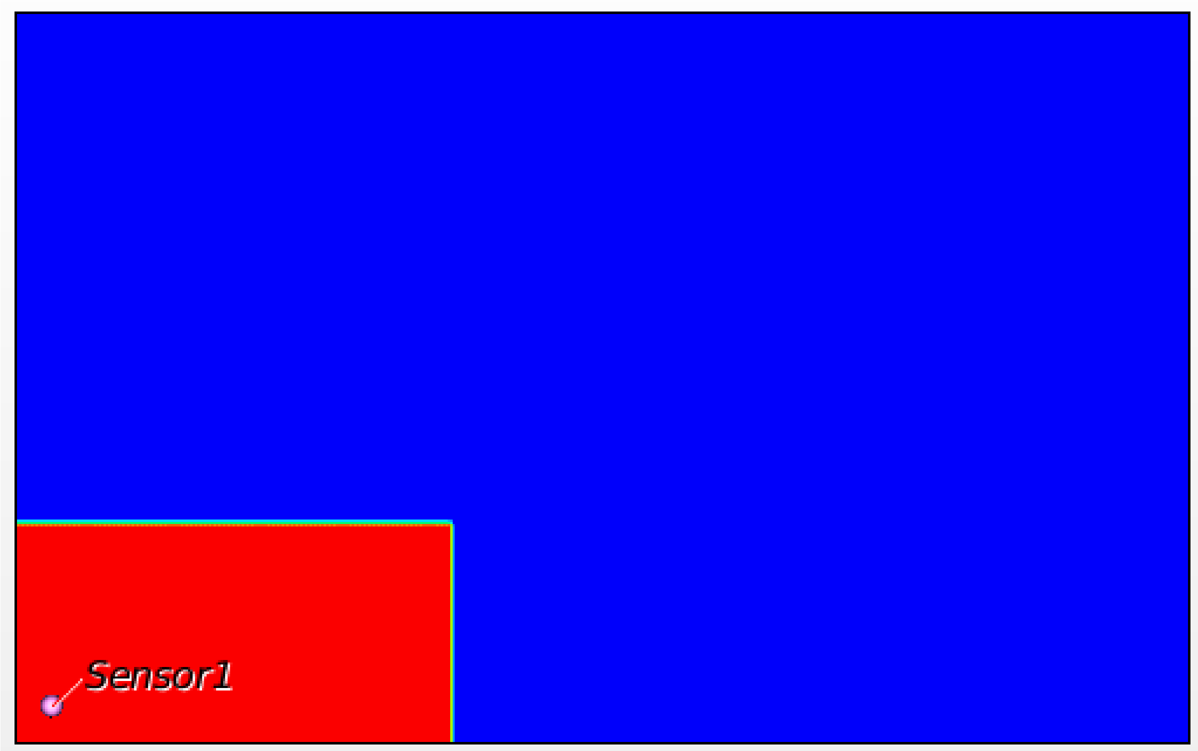

Figure 4: Cylinder Section Setup

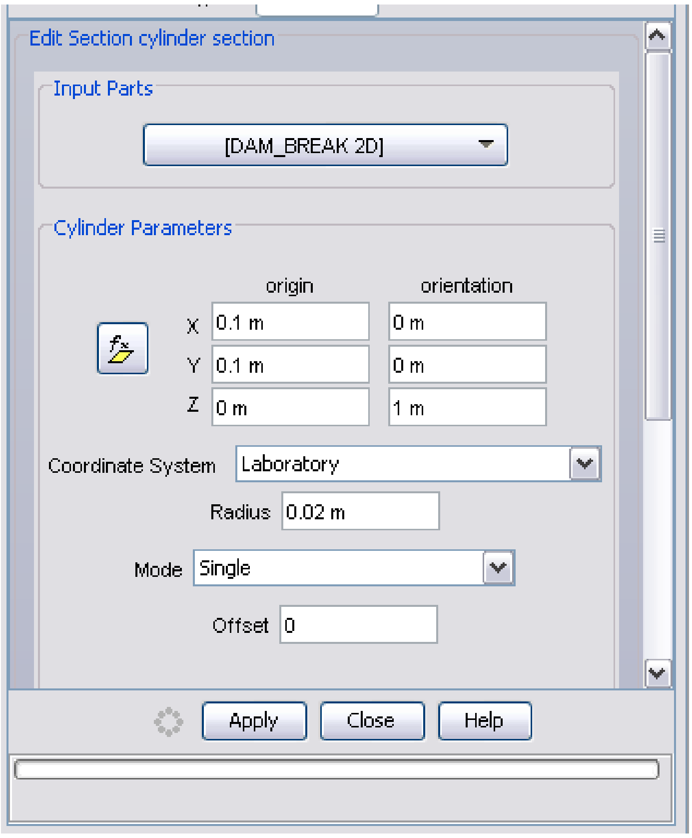

Figure 5: Cylinder Section Setup

Derived Parts → cylinder section, F2 → Sensor1

Reports, RC → New Report → Maximum
Reports → Maximum 1, F2 → PSensor1
Reports → Maximum 1, F2 → PSensor1 → Scalar Field Functions → Pressure
Reports → Maximum 1, F2 → PSensor1 → Scalar Field Functions → Pressure → Parts → Sensor1
Reports → Maximum 1, F2 → PSensor1 → Scalar Field Functions → Pressure → Units → Pa
Reports → PSensor1 > RC → Create Monitor and Plot from Report
Monitors → PSensor1 Monitor > Trigger → Time Step

since STAR-CCM+ lets you output the image of the scene to a unique graphic file after each update of a scene during a simulation run, we can also create images for an animation of the simulation.

Scenes → Scalar Scene 1 → Attributes → Update (figure 6)

We can decide how frequently we want the scene to be updated by changing the **Update Frequency** property; the default setting of 1 means that the scene will update after each iteration or time-step. If you are working with a transient analysis, change the **Update Policy** from Iteration to Time-Step.

Usually, images are easier to work with if stored in a specific folder created with the file manager. For this, create a folder in your working folder and name it IMAGES. Select this folder in *Output Directory* box in the menu. Image file format and size can also be selected. The content of the Scalar Scene 1 will be saved in this directoryonce the computation starts running (see figure 6).

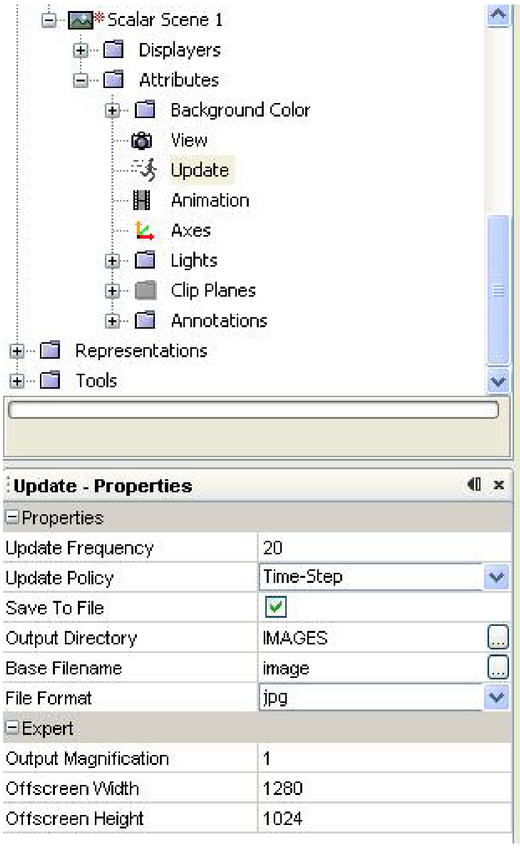

Figure 6: Create images for animation

It can be very interesting to generate a movie of certain simulations. Standard video formats frame per second rate is 25. If you want to show the animation in real time, you should take this into account for the frame rate and the time-step in order to adequately estimate the update frequency (in this case, our time step is 0.002seconds, and if the frame rate per second is 25, this means that the update frequency is 20, which means one image every 20 time steps). In order to create the video you can use the MATLAB script *im2avi.m*. Open it to see an example on how to run it.

7 EXERCISES

1 Measure of impact pressure against downstream wall at point *P* (3.22*m,* 0.16*m*).

Derive Parts, RC → New Part → Section → Cylinder
Input Parts → DAM_BREAK 2D
Origin → [3.21, 0.16, 0.0]
Orientation → [0.0, 0.0, 1.0]
Radius →0.02 m → Create → Close
Derive Parts → cylinder section, F2 → Sensor2

Reports, RC → New Report → Maximum
Reports → Maximum 1, F2 → PSensor2
Scalar Field Functions → Pressure
Parts → Sensor2
Units → Pa

Reports → PSensor2 > RC → Create Monitor and Plot from Report
Monitors → PSensor2 Monitor > Trigger → Time Step

2 Compare with figure 5 of Abdolmaleki et al, 2004 and [Lee et al., 2002] by exporting your time pressure curve as *tp.csv* and executing MATLAB script *dambreak.m*

3 Check mass conservation of the water phase by setting up the corresponding monitoring (use e.g. *Mass Averaged*).

4 Janosi et al. dam break over a wet bed [Jánosi et al, 2004]. Simulate the cases in sections 2.2 and 2.3 with d=18mm and 38mm. The file with the geometrical information is *Janosi_Geometry.igs.* Although the length of the tank (1335mm) is smaller than the real experiment (9.33m), it is enough to observe the main features of the simulation.

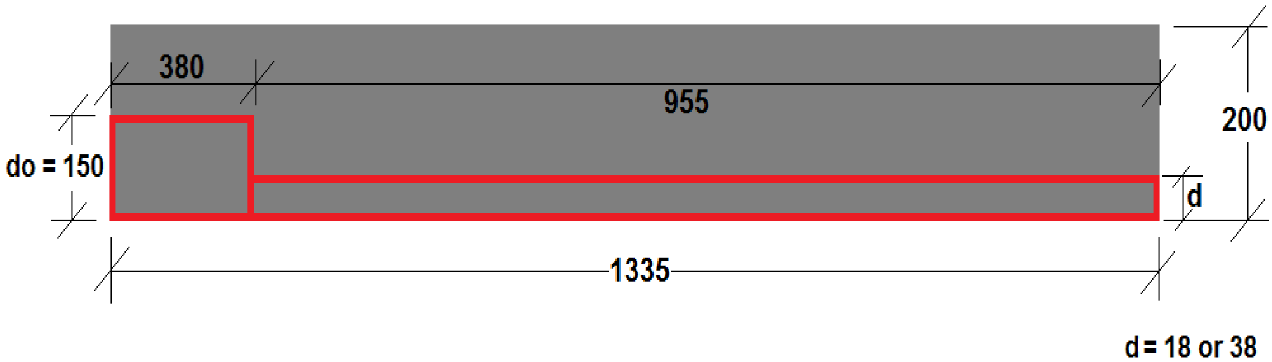

Figure 7: Geometry for the Dam Break over a wet bed (units in mm).

Tips:

Syntax for OR operator is the same as in C or JAVA with a double vertical bar. The field functions for the initial volume fraction for water (d0=0.038m in this one) is:

($$Centroid[0]<=0.38 && $$Centroid[1]<=0.15) || ($$Centroid[1]<=0.038)? 1.0 : 0.0

Air one can be defined by swaping 1.0 for 0.0.

Remember to be careful with the initial condition for the pressure, by defining the pressure at time 0 field function using a conditional and taking into account whether you are in the dam area or in the rest of the reservoir. Something like:

($$Centroid[0]<0.38) ? ------ : ------

Refine the mesh until you are able to capture the breaking events and back flows shown in figure 2 of the paper. Create the movies.

8 RELATED DOCUMENTS

[Abdolmaleki et al, 2004] Abdolmaleki, K., Thiagarajan, P., Morris-Thomas, M. Simulation of The Dam Break Problem and Impact Flows Using a Navier-Stokes Solver. 15th Australasian Fluid Mechanics Conference the University of Sydney, Sydney, Australia 13-17 December 2004

[Jánosi et al, 2004] Jánosi, I., Dominique, J., Gábor, S., Tamás, T. Turbulent drag reduction in dam-break flows. Experiments in fluids 37, 2004.

[Lee et al., 2002] Lee, T.-H., Zhou, Z., and Cao, Y. (2002). Numerical simulations of hydraulic jumps in water sloshing and water impacting. *Journal of Fluids Engineering*, 124(1):215-226.

FUTURE WORK

1. Tutorial 3D, with sphere. Exercises with e.g. Ahmed Body
2. Tutorial for running S60 model.
3. Updating tutorials to version 7.06