

EDF R&D



FLUID DYNAMICS, POWER GENERATION AND ENVIRONMENT DEPARTMENT
SINGLE PHASE THERMAL-HYDRAULICS GROUP

6, QUAI WATIER
F-78401 CHATOU CEDEX

TEL: 33 1 30 87 75 40
FAX: 33 1 30 87 79 16

MAY 2013

documentation

version 3.0 tutorial - Shear driven cavity flow

contact: saturne-support@edf.fr



Code_Saturne Tutorial Series

TUTORIAL 1:
SHEAR DRIVEN CAVITY FLOW





Document Control

Version	1.0
Document reference	TutorialNumber1_ShearDrivenCavityFlow
Revised	27 October 2012



Outline

1	Components.....	4
2	Introduction.....	4
3	PART I – Setting up	5
3.1	What you will learn	5
3.2	Using <i>Code_Saturne</i> and SALOME together for End-to-End Simulations	5
3.3	Accessing and Installing <i>Code_Saturne</i> and SALOME.....	5
3.4	Setting Up End-to-End SALOME / <i>Code_Saturne</i> CFD Simulations.....	5
4	Part II - Shear Driven Cavity Flow CFD Study	8
4.1	What You Will Learn.....	8
4.2	Case Description.....	8
4.3	Creating the <i>Code_Saturne</i> case	9
4.4	Creating the Computational Domain.....	9
4.5	Setting up the CFD Simulation	18
4.6	Running and Analysing the Simulation	25
4.7	Post-processing the Results.....	26
5	References	32
	Appendix A – Reference Data from [4]	34



1 Components

This tutorial makes use of:

- The SALOME [1] platform for geometry generation, meshing, and post-processing
- *Code_Saturne* [2,3] for CFD calculations
- Reference [4] for comparison with published results

To work through this tutorial you will need a computer on which these two software applications are already available or on which you have permission to install them.

2 Introduction

This tutorial is built in two complementary parts.

The first part describes and runs the user through all the procedures required to get going with setting up CFD simulations using SALOME and *Code_Saturne*, from code download to case creation.

The second part illustrates setting up, running, and analysing a CFD simulation entirely with SALOME and *Code_Saturne*, using the laminar, Shear Driven Cavity as an example of a simple case, yet with enough physics to make it interesting and relevant to practical problems.

If you are already familiar with setting up CFD simulations with *Code_Saturne* and SALOME, you may go directly to Part II.



3 PART I – Setting up

The first part of the tutorial is designed to explain the preliminary steps to setup an end-to-end CFD simulation using *Code_Saturne* and SALOME and provide an introduction to these software applications.

3.1 What you will learn

In the first part of the tutorial, you will learn:

- How *Code_Saturne* and SALOME may be combined for end-to-end CFD analyses, from CAD generation to post-processing of CFD results
- How to access and download *Code_Saturne* and SALOME
- How to install *Code_Saturne* and SALOME
- How to set up the control variables for both codes
- How to set up a case with *Code_Saturne*

3.2 Using *Code_Saturne* and SALOME together for End-to-End Simulations

End-to-end simulations are described as the multi-stage process of starting from a geometric description of the object of interest which will form the computational domain all the way to completing and analysing CFD simulations for this object and the physical phenomena (flow, energy) relevant to the operating conditions. This process can be divided in three main steps.

In the pre-processing step, SALOME makes it possible to build the virtual representation of the geometry and to generate a computational mesh which is compatible with *Code_Saturne*.

In the processing step, the *Code_Saturne* Graphical User Interface (GUI) makes it possible to generate the CFD, physical and numerical model, specify the calculations, possibly on multiple processors, and to execute the *Code_Saturne* solver for this model on the mesh created in SALOME. On output, *Code_Saturne* returns data files which are compatible with SALOME.

In the post-processing step, SALOME makes it possible to visualise and analyse the *Code_Saturne* results.

3.3 Accessing and Installing *Code_Saturne* and SALOME

Access to both applications is provided through their respective internet sites. Install utilities, installation instructions, environment variables setup for Linux installations, and code documentation including user manuals can also be downloaded from the sites.

- ***Code_Saturne***

Intermediate and Stable *Code_Saturne* releases can be downloaded from the website:

www.code-saturne.org

From version 2.1.0 onwards, the code is packaged with all its libraries, and the installer which can also be downloaded from the same site makes the installation process fully automatic. However, external, required packages, such as Python, are not part of the *Code_Saturne* package and must be acquired from their respective publishers and installed separately. As part of the installation process, the *Code_Saturne* installer will check for their availability and issue notifications if they are not found.

- **SALOME**

Different versions of SALOME can be downloaded from the website: www.salome-platform.org/. Complete versions come with the source and binary files and interactive installation procedures.

3.4 Setting Up End-to-End SALOME / *Code_Saturne* CFD Simulations

The directory and file structures required and generated by *Code_Saturne* and SALOME are described first and then illustrated by example through the creation of all the pre-requisites for the Shear Driven Cavity Flow end-to-end simulations.

3.4.1 File structure

- **Code_Saturne**

Code_Saturne makes use of a pre-set directory structure to access and save input and output files. Simulations are organised in terms of studies and cases. Conceptually, a study contains a series of cases which all rely on the same geometry. The cases represent different instances of simulations for this common geometry, for example for different operating conditions.

The command 'code_saturne' is used to set up the study and case structure automatically by executing:

```
code_saturne create -s NameOfStudy -c NameOfCase
```

When executing the command, *Code_Saturne* not only creates the directory structure for the specified study and case but also all their sub-directories and required files. The automatic procedure ensures that all the files required by *Code_Saturne* are available and in the expected directories.

If a study already exists, a new case may be created automatically simply by executing:

```
code_saturne create -c NameOfCase
```

within the study directory.

An example of study and case creation is given in the next Section for the Driven Cavity case which is detailed in Part II.

- **SALOME**

SALOME does not require a specific directory structure or input and output files. Whilst a Python Textual Interface (TUI) may be used to record macros of instructions which can then be read in by SALOME and executed automatically, creating the geometry of a computational domain and meshing of the domain can be conducted entirely from the Graphical User Interface (GUI). The GUI can be started from any directory as per the instructions from [1].

3.4.2 Shear Driven Cavity Flow

For this example, in the tutorial directory we create the 'DrivenCavity' study and the first case which we name 'Reynolds1000' after its flow Reynolds number:

```
code_saturne create -s Driven Cavity -c Reynolds1000
```

The 'DrivenCavity' study directory is created, containing the sub-directories (underlined):

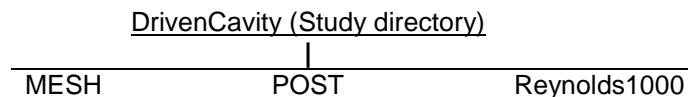


Figure 3.1: Study directory structure.

Initially, the MESH and POST sub-directories are both empty. By default when run from a case directory, in this example 'Reynolds1000', *Code_Saturne* is set up to look for the mesh file for the case in the MESH directory. The POST directory could contain post-processing routines if required.

Upon creation, the case 'Reynolds1000' contains by default the sub-directories (underlined) and files:

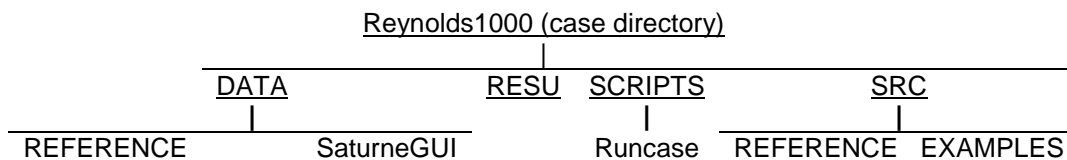


Figure 3.2: Case directory and file structure.

The meaning of the different directories and files is recalled here briefly. For detailed information, please refer to the *Code_Saturne* user manual [3].

The 'DATA/REFERENCE' directory contains data files (tabulated thermophysical properties for chemical species and atmospheric properties) and the Python file 'cs_user_scripts.py' in which users can modify *Code_Saturne* (cs) parameters and settings.

'SaturneGUI' is a shell file pointing to the *Code_Saturne* GUI executable.

Initially, the 'RESU' directory is empty. It will contain the results files and *Code_Saturne* outputs of each *Code_Saturne* run, organised in chronological order.

The sub-directory 'SCRIPTS' contains the bash file 'runcase', in which the PATH to the *Code_Saturne* executable and the run command are automatically setup for the case name.

The sub-directory 'SRC' is used to store all the user source files. The directory 'SRC/REFERENCE/' contains templates for all the available files, whilst examples of implementation are stored in 'SRC/EXAMPLES'.

4 Part II - Shear Driven Cavity Flow CFD Study

In the second part of the tutorial, the preparation, simulation and analysis of the 'Reynolds1000' case of the 'DrivenCavity' study is described, from the construction of the computational domain and mesh to the preparation, running and post-processing of the CFD simulation.

4.1 What You Will Learn

Through this tutorial, you will learn how to perform an end-to-end CFD simulation using *Code_Saturne* [2, 3] together with the SALOME [1] platform, from creating the computational domain to analysing the end results and comparing them with available data. Specifically, you will:

- Create a computational domain with SALOME using available shapes and groups
- Create an hexahedral mesh with different mesh refinement in the X, Y and Z directions with SALOME
- Setup a steady-state, viscous, laminar, isothermal, constant properties fluid CFD simulation with non-slip walls, a moving wall, and symmetry planes with the *Code_Saturne* GUI and using the SALOME mesh
- Control and run the *Code_Saturne* simulation from the *Code_Saturne* GUI
- Examine the *Code_Saturne* output and results files, including data along specified profiles and at monitoring points
- Analyse and visualise the results with the SALOME GUI

4.2 Case Description

A viscous fluid is contained in a box, or cavity. All the walls of the cavity are stationary, but for one wall which is sliding in its plane and sets the fluid in motion inside the box through entrainment.

For the purpose of this example, the cavity is rectangular, two-dimensional, and the top wall of the cavity slides from left to right, as illustrated in Fig. 4.1 below.

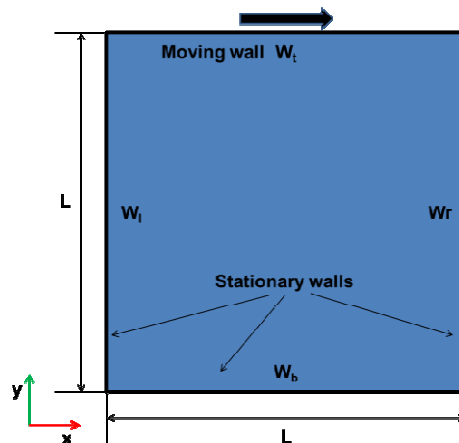


Figure 4.1: Schematic of the 2D Lid Driven Cavity.

Variants of this case and numerical results are also described in [4].

4.2.1 Geometry

The cavity is a square of length $L=1.0\text{m}$.

4.2.2 Fluid Properties

The fluid is given the properties listed in Table 4.1 below:

Property	Value	Units
Density (ρ)	1.0	kg/m ³
Viscosity (μ)	10 ⁻³	Pa-s

Table 4.1: Fluid Properties.

4.2.3 Boundary Conditions

The problem is considered to be isothermal and the domain is fully enclosed by impermeable, non-slip walls. This means that, exactly at the surface of the walls, the fluid inside the box attaches to the walls and has exactly the same velocity as the walls. Therefore, the problem is fully defined by specifying the velocity of the walls (Table 4.2). The side and bottom walls are stationary. The top wall slides in the positive X direction at a speed of 1.0m/s.

Surface	Velocity component (m/s)	
	v_x	v_y
W_l	0.0	0.0
W_b	0.0	0.0
W_r	0.0	0.0
W_t	1.0	0.0

Table 4.2: Wall velocities.

4.2.4 Flow Regime

The flow Reynolds number is evaluated to determine whether the flow is in the laminar or turbulent regime. The Reynolds number is calculated based on the lid velocity, the box dimensions, and the fluid properties.

For this case, the fluid properties and lid velocity have been chosen so that $R_e = \frac{\rho UL}{\mu} = 1000$, indicating that the problem is in the laminar flow regime.

4.3 Creating the *Code_Saturne* case

The 'DrivenCavity' study and 'Reynolds1000' case are created by following the instructions in Part I of this tutorial.

The next sections describe how to setup and run the lid driven cavity simulations for a flow Reynolds number of 1000.

4.4 Creating the Computational Domain

The SALOME platform is used first to create the geometry of the cavity which defines the bounds of the computational domain and then mesh this domain. Installation of this tool is described in Tutorial I/Part I.

Code_Saturne is a 3D code and, therefore, the computational domain will be created as a 3D box, aligned with the X, Y, and Z directions. However, as we are only interested in 2D modelling and the flow motion in the X-Y plane, the box will be designed to be thin and contain only one computational cell in the Z direction.

Start SALOME from within the study directory: here, directory 'DrivenCavity'.

4.4.1 Geometry

Upon starting SALOME, you are presented with a blank screen topped by the main menu bar. The menu bar contains drop-down menu topics and short-cut icons for direct access to some of the menu selections.

For detailed information about SALOME, please refer to SALOME's manuals and documentation [1].

First, the current SALOME file is saved by accessing **File→Save**. For this tutorial, we choose the file name 'Cavity' (Fig. 4.1). The file 'Cavity.hdf' will be created in directory 'DrivenCavity'.

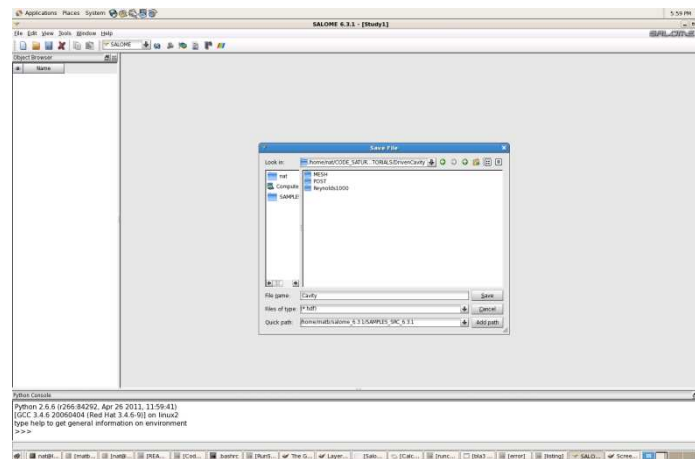


Figure 4.1: Saving the SALOME file.

It is advisable to save the file at regular intervals during this tutorial and, preferably, before and following all significant modifications. That way your latest work is preserved and, should mistakes happen, it is always possible to exit SALOME and restart it from the latest saved file.

As SALOME is an integrated interface to different capabilities: CAD, meshing, visualisation, the next step is to access the 'Geometry' module of the program in order to start building the shape of the computational domain.

In the main menu bar, either click on the 'Geometry' icon or choose 'Geometry' from the drop-down module menu as shown in Fig. 4.2 below.



Figure 4.2: Selecting 'Geometry' in the application chooser.

The 'Geometry' menus then become available in the top bar and the 'OCC scene viewer' opens in the right hand side, where the geometry objects can be visualised and manipulated. The scene viewer comes with its own set of 3D-viewing menus to control the display and perform operations such as panning in and out, and rotating and moving the object in the scene.

The 'Geometry' module of SALOME makes it possible to build bespoke objects from scratch from constituting elements such as vertices, edges and faces, or from pre-set generic shapes.

In this instance, we use the latter, more direct method and create the cubic cavity shape by selecting: **New Entity→Primitives → Box**. Alternatively, use the 'Box' icon short-cut on the menu bar.

A 'Box Construction' pop-up menu is then activated, as shown in Fig. 4.3 below.

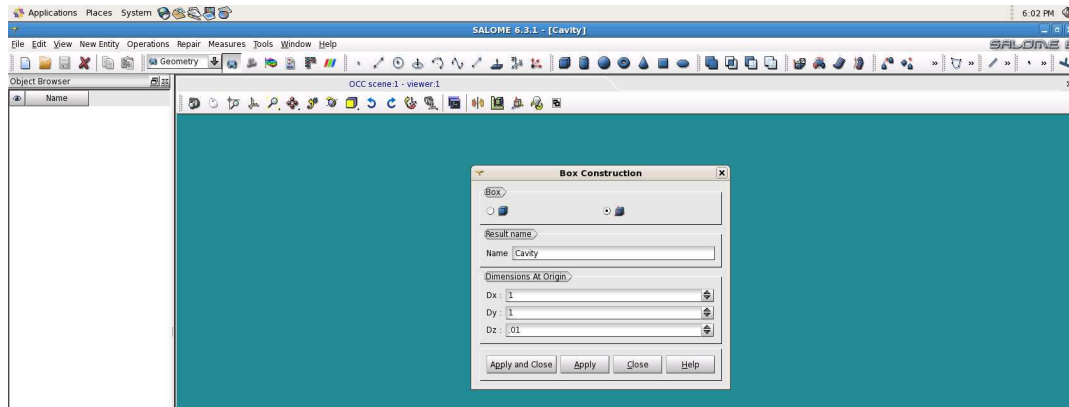


Figure 4.3: Box construction menu.

In this menu, choose the second method of Box construction, which makes it possible to build a prismatic box attached to the origin by defining its three dimensions. The default name is changed to 'Cavity' and the dimensions in the X, Y, and Z directions are specified according to the problem definition in **Section 4.2**.

The resulting Cavity object is shown in Figures 4.4a-b below, as wireframe in Fig. 4.4a and after applying shading in Fig. 4.4b.

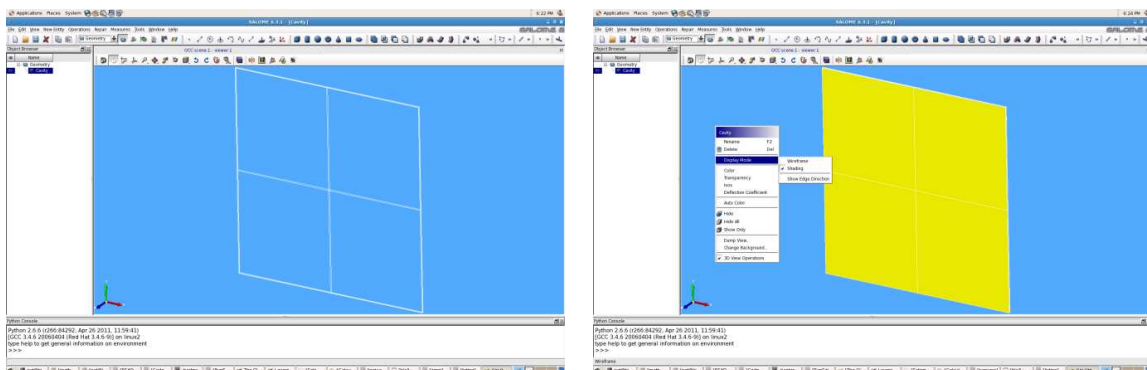


Figure 4.4a-b: Wireframe (left) and Shading (right) display modes.

As shown in Fig. 4.4b, to change the Display Mode to 'Shading', right click in the scene viewer to open the drop down menu, then select 'Display Mode' and 'Shading'. Likewise, 'Change background...' may also be accessed from the same drop-down menu to change the background colour of the 'OCC viewer' to light blue as in the figures.

At this point, the cavity object is complete and could be meshed straight away. However, since the box has been created as a whole entity it would then neither be possible in the mesher to specify different meshing characteristics for the X, Y and Z directions nor to identify the 6 external surfaces as boundaries for the *Code_Saturne* calculations. Before meshing, it is then necessary to create and identify these elements. In the next step, you will automatically subdivide the Cavity object in its constituting 'Faces' and 'Edges' using the powerful SALOME Geometry 'Explode' utility.

With the 'Cavity' object selected in the Object Browser in the left-hand column, in the top menu bar, click on **New Entity → Explode** to access the 'sub-shape' menu. By using the 'Apply' action at the bottom at the menu, as opposed to 'Apply and Close', you will be able to first create the faces and then the edges without leaving the 'sub-shape' pop-up. The faces are created first, as shown in Fig. 4.5 below.

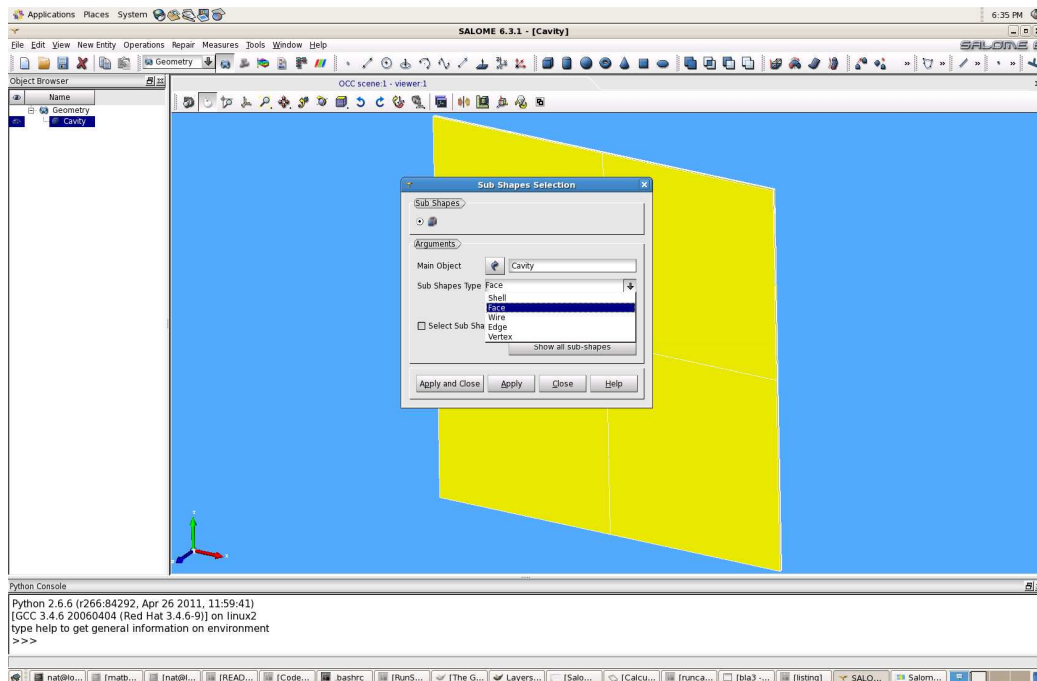


Figure 4.5: Selecting 'face' sub-shapes in to 'Explode' the Cavity object.

Check that the Main Object is the 'Cavity' and select 'Type: Face' in the drop-down selector. Press 'Apply' to 'explode' the box in its constituting faces. Do not close the 'sub-shape' menu. Next, select 'Edge' in the drop-down selector and press 'Apply and Close' to create the edges and leave the 'sub-shape' menu. The 'Cavity' object in the object browser should now list all the faces and edges, as shown in Fig. 4.6 below.

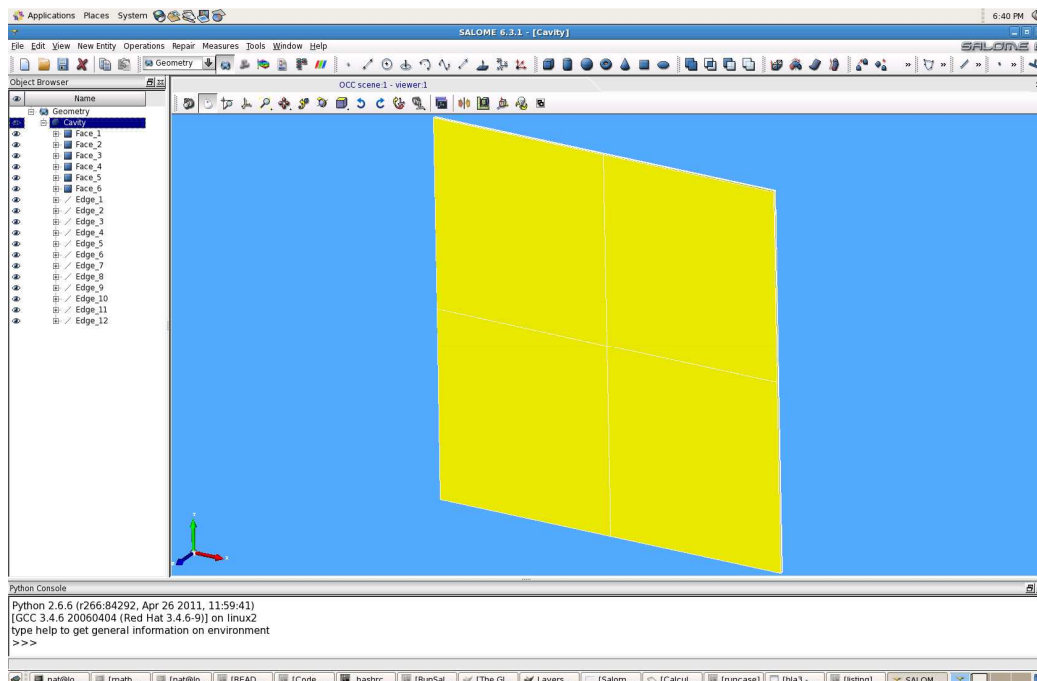


Figure 4.6: Cavity sub-divided in all its faces and edges.

Having clearly identified parts of the Cavity on which meshing and boundary characteristics can be attached, the Cavity could be meshed now. However, for convenience, the edges and faces will first be assembled in a smaller number of easily identified groups. This will make it easier to specify geometry parts later in the mesher.

First, the faces are grouped according to the boundary types listed in Table 4.2 above. Right-click on 'Cavity' in the object browser and select 'Create Group' to activate the pop-up menu, as shown in Fig. 4.7 below.

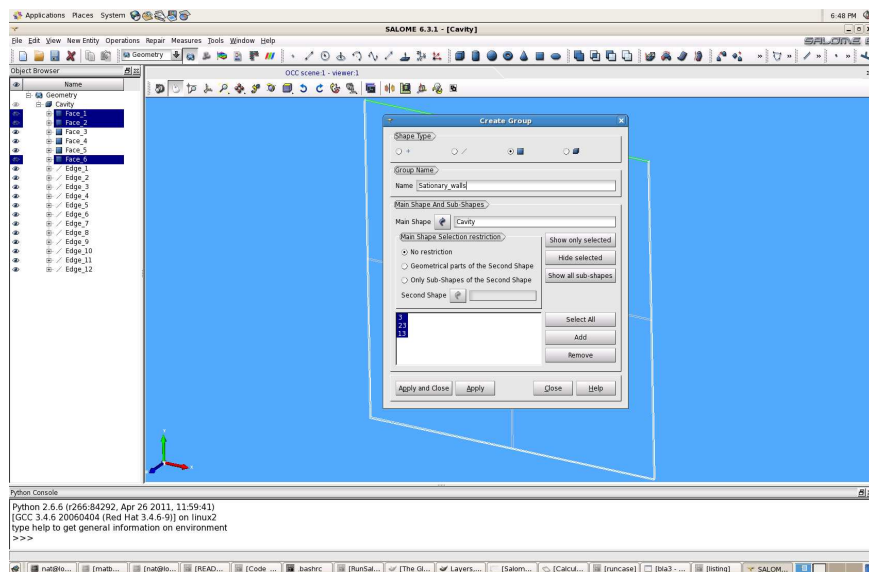


Figure 4.7: Creating the 'Stationary_wall' face group.

Alternatively, access the group creation via 'New Entity' or its short-cut icon on the main menu bar.

Select the 'Face' icon as 'Shape type', name the group, and ensure that the 'Main Object' is 'Cavity'. To be easily identifiable, the groups are named after their boundary types. The first group is created for the external, stationary walls and named 'Stationary_wall'. With the default 'no restriction' choice already active in the 'Main shape restriction selection' menu, start adding the required faces by left-clicking on them in the Object Browser. The faces may be added one by one, or together by multiple-selection through left-clicking and holding the 'Ctrl' key. Press 'Apply' and repeat these steps for the next two groups: 'Symmetry_plane' and 'Sliding_wall', pressing 'Apply' each time.

Do not close the 'Create Group' pop-up menu and move to creating groups for the edges. Here, the edges are grouped per direction, in order to later be able to designate different discretisations along the X, Y, and Z directions in the mesher. Switch the 'Shape type' to the symbol for edges, name the edge groups according to their direction, and add edges picked from the Object Browser, as seen in Fig. 4.8 below.

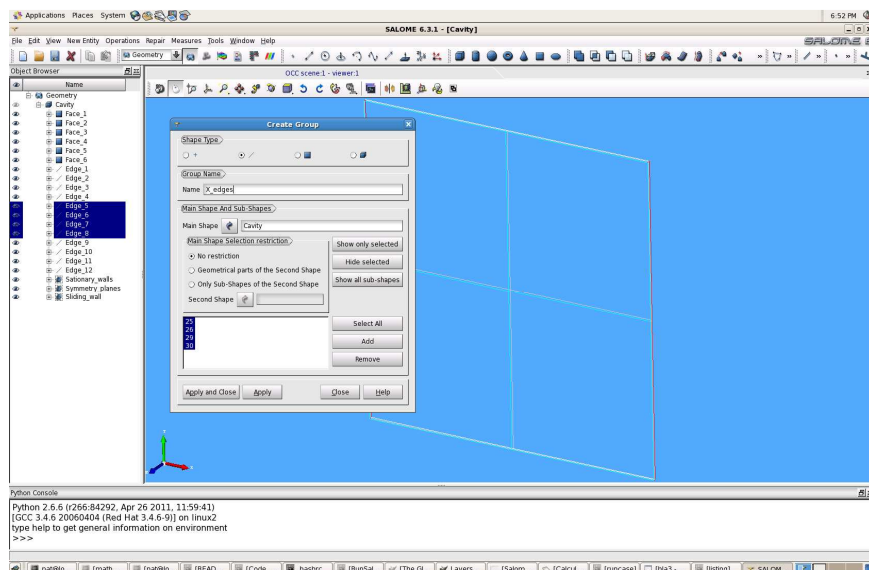


Figure 4.8: Creating the 'X_edges' edge group.

In succession, create the edge group 'X_edges', press 'Apply', the edge group 'Y_edges', press 'Apply', the edge group 'Z_edges' and conclude the creation of groups by pressing 'Apply and Close'.

The face and edges groups are now listed under the Cavity object in the Object Browser (Fig. 4.9).

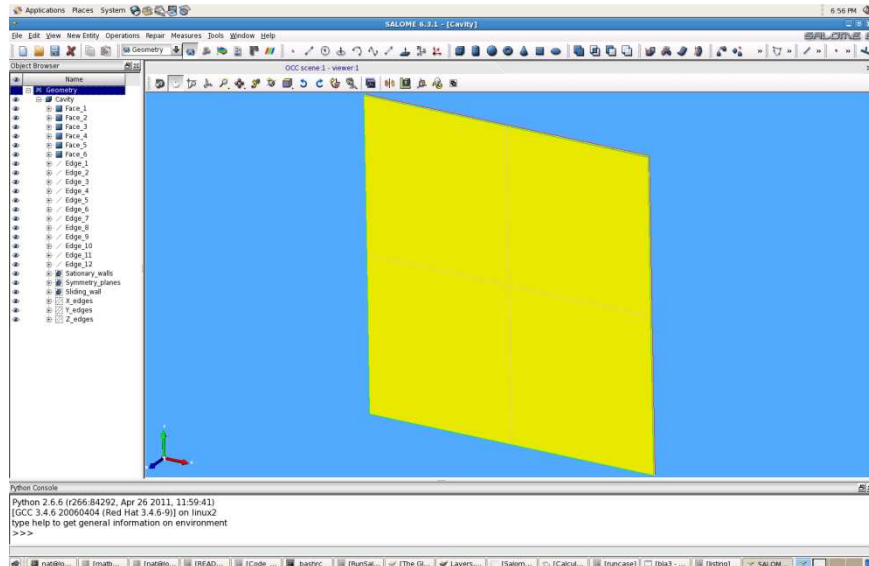


Figure 4.9: Cavity with the edges, faces, and groups.

'Cavity' is now ready to be meshed. Save the file and proceed to Meshing.

4.4.2 Meshing

Switch to the meshing module, which opens the 'VTK scene viewer' as shown in Fig. 4.10 below. In this view, the background colour has been changed to white similarly to the 'OCC scene viewer' for the 'Geometry', by right clicking in the viewer and accessing 'Change background...'

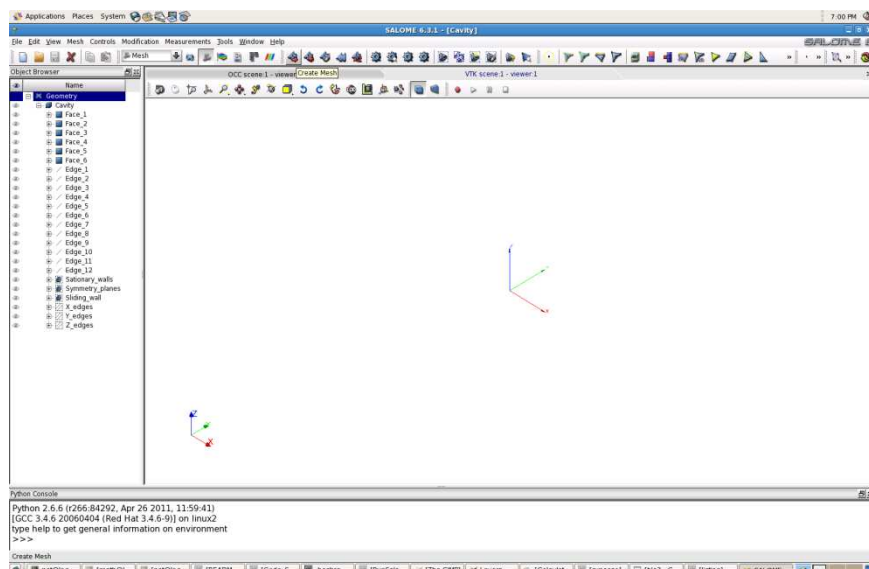


Figure 4.10: Switching to the 'Mesh' module.

Similarly to the earlier creation of the geometry object 'Cavity' in the 'Geometry' module, a 'Cavity_mesh' mesh object is created first by selecting **Mesh** → **Create mesh** in the main menu bar.

As shown in Fig. 4.11 below, in the 'Create mesh' pop-up menu, type 'Cavity_mesh' for the 'Name' and select 'Cavity' as the 'Geometry' object by left clicking on it in the Object Browser.

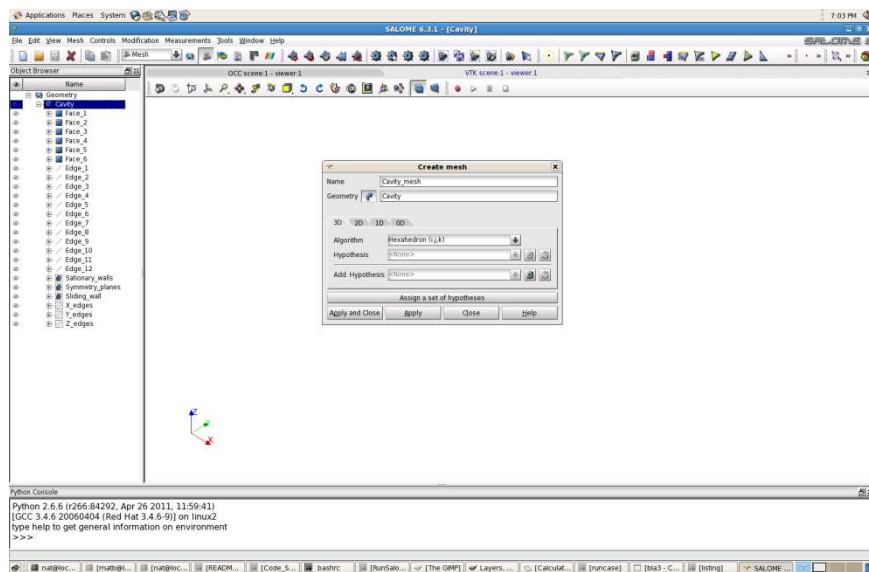


Figure 4.11: Creation of the main mesh object.

The main mesh object holds the global characteristics of the Cavity mesh.

For the volume, in the '3D' tab, choose 'Hexahedron (i,j,k)'. All the (external) surfaces of the Cavity will be meshed similarly, therefore a global setup is applied for the entire mesh via this pop-up menu by clicking on the '2D' tab and selecting 'Quadrangle (Mapping)'. These two selections will result in a block-structured mesh, with an (i,j,k) matrix structure. This requires identical discretisations along all the edges aligned along the same directions. If we wanted the discretisation to be the same along all the directions, for example 15 sub-divisions along the X, Y, and Z edges, the '1D' tab could be selected and this unique discretisation specified. However, here we need to be able, at least, to specify the Z direction independently of the two others to create a 3D domain which is one cell deep. Therefore, the edge discretisations will be specified separately using 'sub-meshes'.

The sub-mesh defining the meshing characteristics in the X direction is specified first.

Right-click on 'Cavity_mesh' in the Object Selector and select 'Create sub-mesh'. Alternatively, with 'Cavity_mesh' selected (highlighted) in the Object Browser, the sub-meshes can be created from 'Mesh' or the short-cut icon in the main menu bar. This activates the 'Create sub-mesh' pop-up menu in the 'VTK scene viewer' as shown in Fig. 4.12 below.

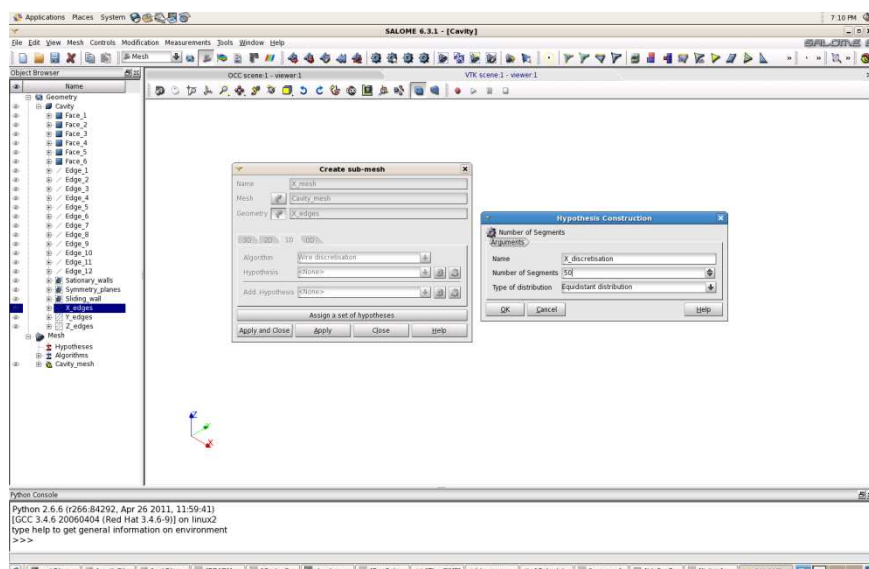


Figure 4.12: Creation of the X_mesh sub-mesh.

In the name field, enter 'X_mesh'. The 'Cavity_mesh' should already be selected in the Mesh field. If not, select it by clicking on 'Cavity_mesh' in the Object Browser. Next, pick 'X_edges' for the 'Geometry' field which stores the Geometry Objects the sub-mesh will apply to. Selecting an edge group for the Geometry automatically deactivates the 3D and 2D tabs in the Algorithm menu. Select 'Wire discretisation' in the drop-down 'Algorithm' menu and press on the gear button next to 'Hypothesis' to specify the discretisation in its own pop-up menu. The 'Number of Segment' object is renamed 'X_discretisation' for future reference. Choose 50 segments with an 'Equidistant distribution' and press 'OK'. Returning to the 'Create sub-mesh' menu, press 'Apply' to instantiate the 'X_mesh' sub-mesh. The sub-mesh objects for the other two directions are defined similarly.

Starting with the Y direction, the sub-mesh is renamed 'Y_mesh'. The same number of segments is applied in the Y and X directions. Therefore, for this sub-mesh the previously created 'X_discretisation' 'Number of Segment' Object is reused in the 'Hypothesis'. Complete the specification of 'Y_mesh' and press 'Apply'.

Lastly, the sub-mesh for the Z direction is defined. Named 'Z_mesh', it is similar to the 'X_mesh' and 'Y_mesh', but defines the discretisation as 1 segment only. For 'Z_mesh', define a new 'Number of Segment' object named 'Z_discretisation' and specifying 1 segment. Complete the specification of 'Z_mesh' and press 'Apply and Close' to instantiate 'Z_mesh' and leave the 'Create sub-mesh' menu.

The mesh characteristics of the Cavity are now fully specified, as shown in Fig. 4.13 below where all the characteristics of 'Cavity_mesh' and its sub-meshes have been listed.

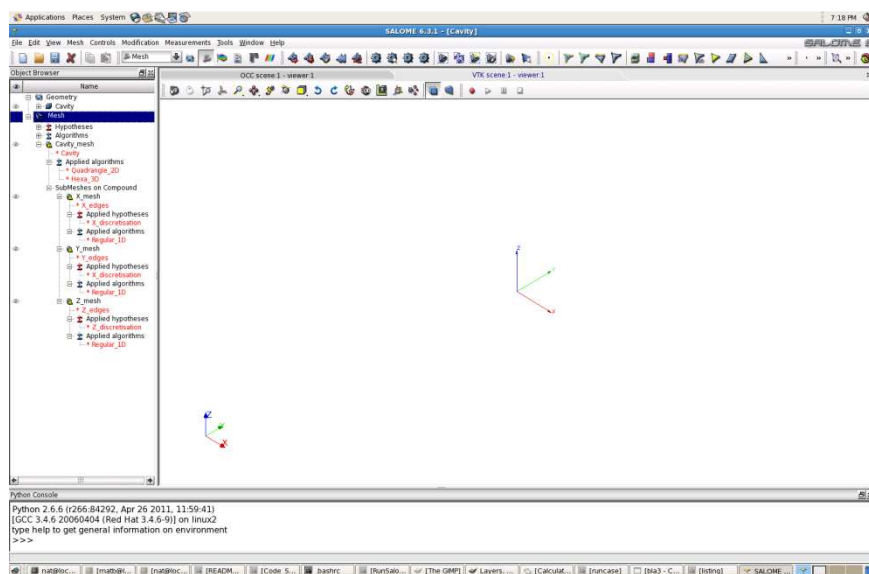


Figure 4.13: Complete specification of 'Cavity_mesh' and sub-meshes in the three directions.

To mesh the computational domain, right-click on 'Cavity_mesh' in the Object Browser and select 'Compute', as shown in Fig. 4.14a below. Again, with 'Cavity_mesh' selected, 'Compute' can also be accessed via the main menu bar and, either, the 'Mesh' category or the short-cut icon. The mesh is created and a pop-up window appears automatically summarising the mesh characteristics (Fig. 4.14b).

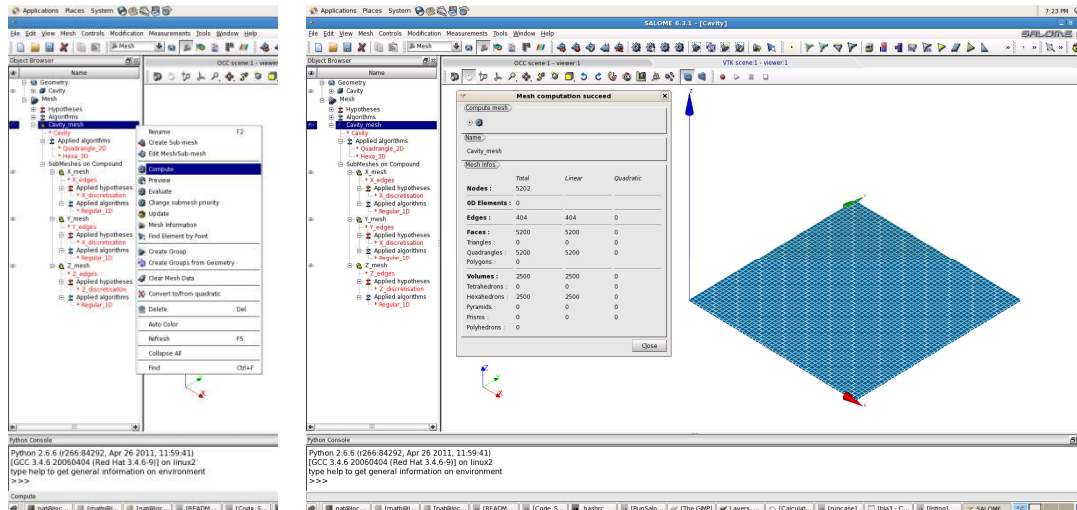


Figure 4.14a-b: Meshing activation (left) and resulting mesh (right).

Before the mesh can be used in *Code_Saturne*, the boundary surfaces must be identified so that it will be possible to impose boundary conditions in *Code_Saturne*. To designate these surfaces, we now can call on the groups of faces which were created previously in the 'Geometry'.

Right-click on 'Cavity_mesh' in the Object Selector and select 'Create Group' to create containers of mesh elements. The 'Create Group' pop-menu is activated. The group of mesh faces for the 'Stationary_walls' boundary is created first. As shown in Fig. 4.15 below, in the 'Create Group' pop-up menu choose 'Face' for the element type. Ensure that 'Cavity_mesh' has been selected for the mesh and do not modify the 'Name' for now.

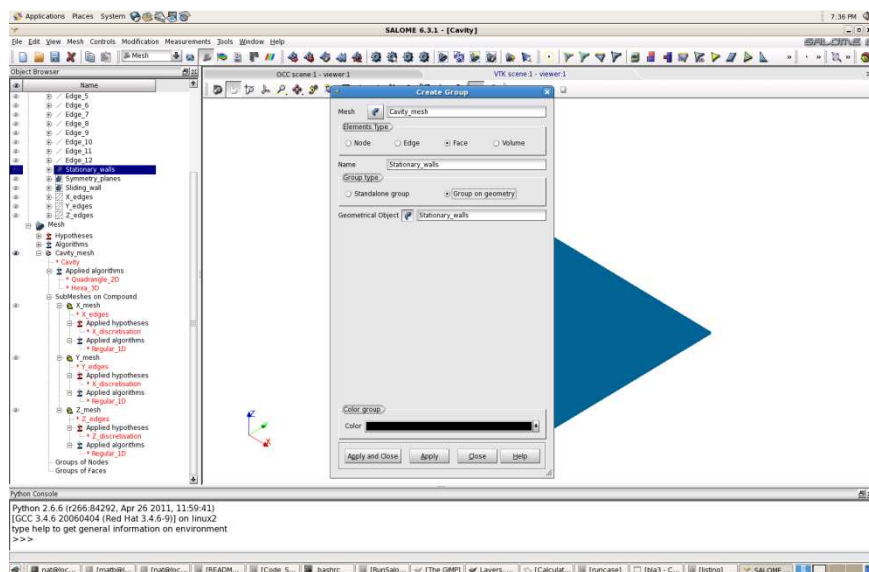


Figure 4.15: Creation of the mesh face group 'Stationary_walls'.

In 'Group Type', choose 'Group on geometry'. This will automatically restrict the choice to already available groups on the 'Cavity' geometry. For 'Geometry Object', left click on 'Stationary_walls' in the 'Geometry' part of the Object Chooser. By default, the mesh face group is automatically renamed with the name of the 'Geometry Object' selected. To help visualisation, the colour of this group of faces is switched to 'black'. Press 'Apply' and repeat these operations to create the 'Symmetry_plane' and 'Sliding_wall' groups of faces, choosing a different colour for each group. When the last group has been defined, press 'Apply and Close' to leave the pop-up menu.

The groups containing the boundary faces are now defined and may be visualised in the 'VTK scene viewer', together or individually, by selecting the groups of mesh faces in the Object Browser. For example, the 'Stationary_walls' group is shown in Fig. 4.16a, and all the groups are visualised in Fig.

4.16b. Note that the names of the groups which have been selected appear in the top right corner of the viewer, in the colour scheme chosen for the group.

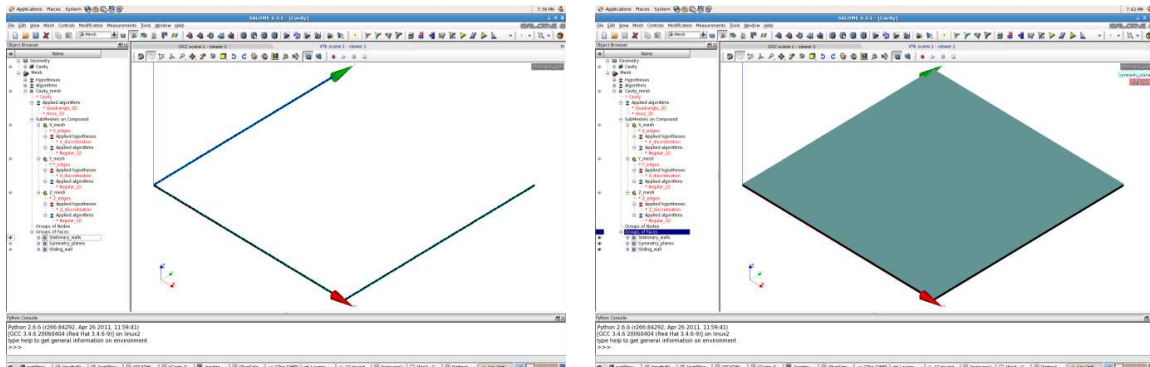


Figure 4.16a-b: Visualisation of the 'Stationary_wall' group (left) and all the groups (right).

The mesh is now ready to be exported in a separate file which can be read in by *Code_Saturne*.

Before exporting the file, it is important to highlight an important naming convention. As the names of the mesh and its elements will be used as their identifiers in *Code_Saturne*, they should not include any blank spaces or they will not be recognised by *Code_Saturne*.

Save the SALOME file and export the mesh file in '.med' format by selecting from the main menu: **File→Export→MED file**, as shown in Fig. 4.17 below. The file should be placed in the 'MESH' directory of the DrivenCavity study, where *Code_Saturne* will expect the file to be situated by default.

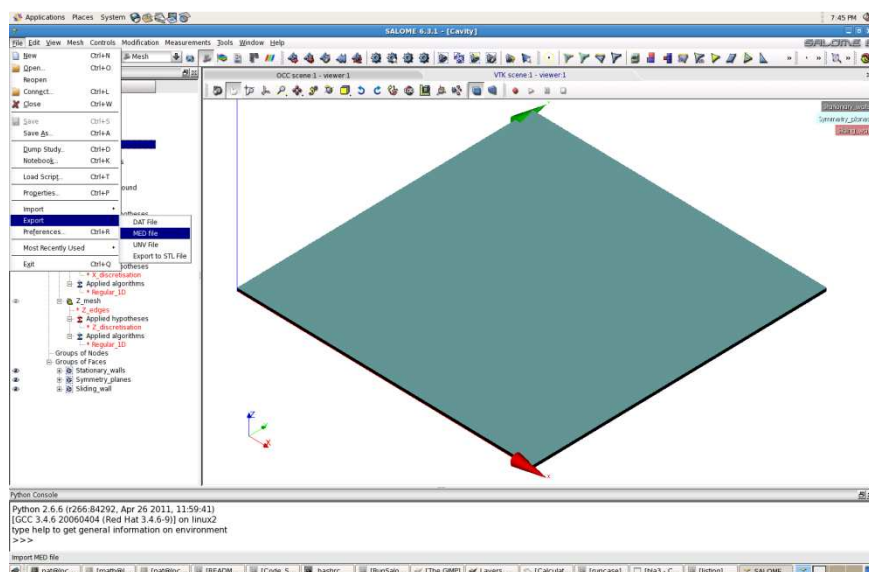


Figure 4.17: Exporting the mesh file in '.med' format.

For the file name, choose 'Cavity_mesh'; the '.med' extension is automatically added. You are now ready to set up the CFD simulation with the *Code_Saturne* Graphical User Interface (GUI).

4.5 Setting up the CFD Simulation

The CFD case is setup and run from the *Code_Saturne* GUI. Go to the case directory, 'Reynolds1000', and start the GUI by typing 'code_saturne gui'.

A startup screen with blank 'Study', 'Case' and 'XML file' categories will appear as shown in Fig. 4.18a below. See Tutorial I/Part I to recall how these directories relate to the *Code_Saturne* file structure, or the *Code_Saturne* manuals and documentation [3] for a more in-depth description.

Select **File → New file** in the top left drop down menu bar.

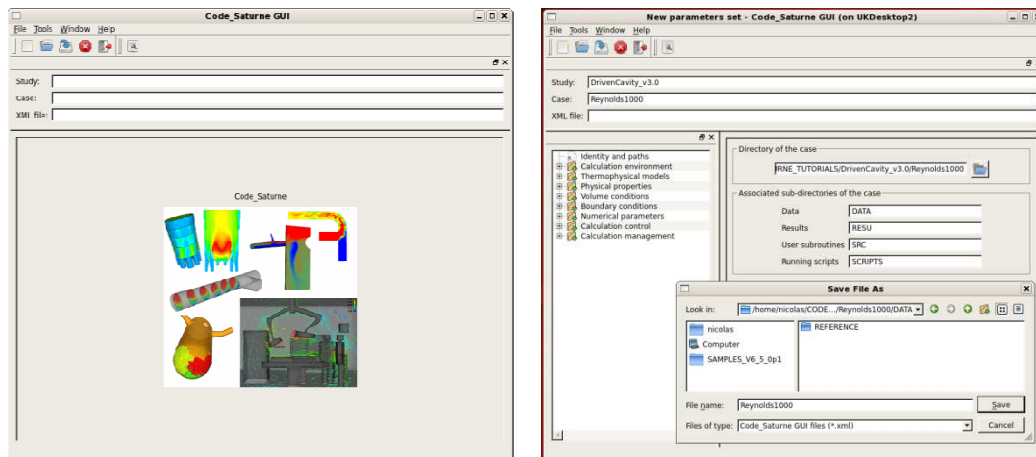


Figure 4.18a-b: Code_Saturne GUI startup (left) and initial screen for a new case (right).

The GUI automatically recognises the directory structure and, by default, fills in the 'Study' and 'Case' categories with the names of the study and the case the GUI is running from. If not, redirect the GUI to the desired 'Directory of the case' (Fig. 4.18b). If the file structure under the case directory is correct (**Section 3.4**), the GUI will then correctly recognise and identify the different directories.

The GUI file should now be saved. Select **File → Save**, and input 'Reynolds1000' in the empty 'Name' box of the 'Save File As' popup (Fig. 4.18b). By default, the file will be saved in the 'DATA' directory. Do not change this setting. Press Save.

You can now proceed with setting up the case, in the top down order of the folders in the left-hand column of the GUI.

Press on 'Calculation Environment' to open the folder and select 'Meshes Selection' in order to specify the mesh file which will be used for these calculations. In the 'Meshes' tab, for 'Mesh import', 'Import meshes' is already selected by default. The 'Local mesh directory' should already be pointing to the 'MESH' directory for the study and where the mesh file had been exported to from SALOME.

Initially, the 'List of meshes' table is empty. Click on the '+' sign button below the table and a popup menu will show the content of the local mesh directory (here './.MESH') from which the mesh file can be selected. Select the 'Cavity_mesh.med' file previously exported from SALOME at the end of the meshing process, and press 'Open'. The file name will then be added to the list of meshes (Fig. 4.19).

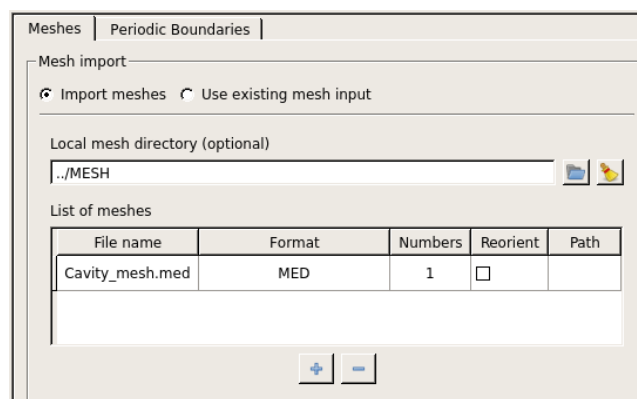


Figure 4.19: Selecting the mesh file for the calculations.

No further input is necessary as the faces do not require reorientation, joining, or sub-dividing, and there are no periodic boundaries to be concerned with.



Now, select 'Mesh quality criteria' in the left hand column. In this menu, the 'Check mesh' facility can be used to analyse the mesh and, for visualisation of the mesh quality criteria, export files in the specified output format. As shown in Fig. 4.20 below, the 'EnSight Gold' format is preselected by default.

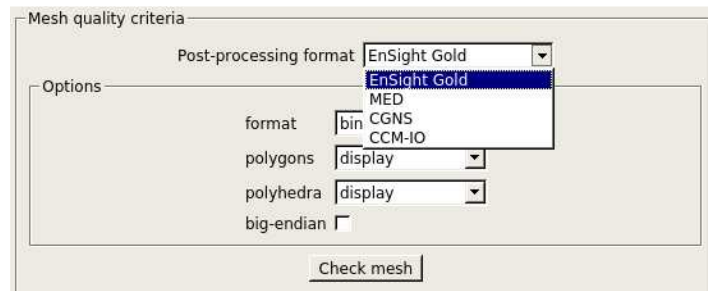


Figure 4.20: Mesh quality criteria output file format selection.

Next, move to 'Thermophysical Models' to specify the flow physics for the calculations.

In 'Calculation features', change 'unsteady flow' to 'steady flow' in the choice of algorithms at the top. Leave all the other default values unchanged: single-phase flow is active and the multiphase, atmospheric, combustion, electrical, and compressible models are all inactive (Fig. 4.21).

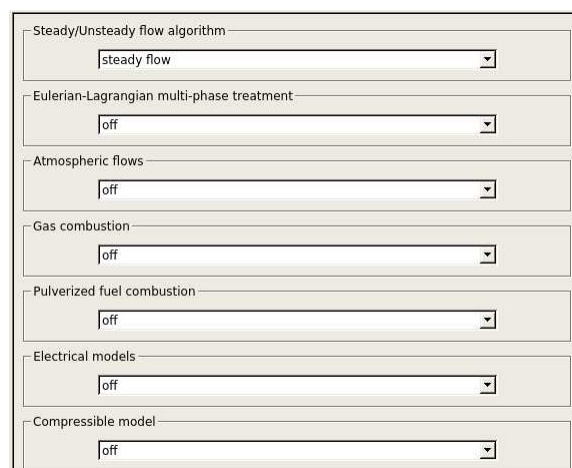


Figure 4.21: Selection of the flow physics.

Continue setting up the 'Thermophysical Models'.

By default 'Mobile mesh (ALE method)' is not selected. Therefore, as the mesh in our case is stationary, 'Deformable mesh' does not need visiting and changing. However, 'Turbulence models' needs to be changed to 'No model (i.e. laminar flow)', which is the correct flow regime for the flow Reynolds number that we have chosen (Fig. 4.22).

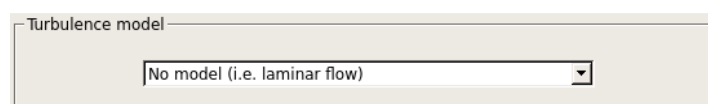


Figure 4.22: Selection of turbulence models.

'Thermal model' is inactive by default. Therefore, it does not need to be visited and changed.

Save the file and, as there are no additional scalars, skip the 'Species transport' category and move to the 'Physical Properties' folder where the thermal and physical characteristics of the fluid of interest for the calculations will be specified.

In 'Reference values', the pressure reference value of 101325.0 Pa is set by default and does not need to be changed. The other default values for velocity and length can also be left unchanged.

However, the 'Fluid Properties' must be modified for our imaginary fluid. Click on 'Fluid Properties' in the left hand panel and, as per Fig. 4.23 below, alter the values of 'Density' and 'Viscosity' to the chosen values which are listed in Table 4.1 above.

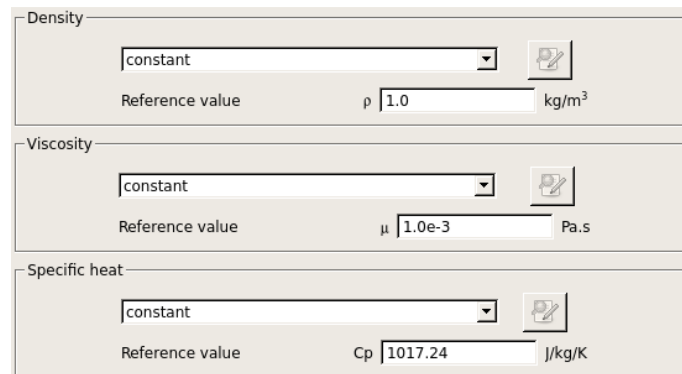


Figure 4.23: Selection of fluid properties.

The default value for 'Specific heat' does not need to be altered as it is not a parameter in these isothermal calculations. Likewise, gravity is not a parameter of this problem (constant density), therefore the 'Gravity' category does not need to be visited and changed.

Click on the 'Volume Conditions' folder. Sub-volumes of the computational domain ('Volume regions definition') have not been defined as separate regions for initialisation and 'Coriolis source terms' are not taken into account in this tutorial. Therefore, the 'Volume Conditions' category must be changed only for 'Initialization'.

In this sub-folder, press on the icon button next to 'Velocity' to bring up the 'Mathematical Expression Editor' and change the initial 'u' velocity to 0.1 m/s as shown in Fig. 4.24 below.

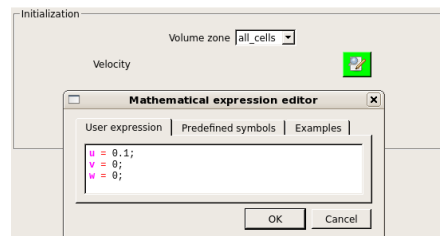


Figure 4.24: Selection of initial values.

To define both the boundaries and the boundary conditions, under the 'Boundary Conditions' folder, select 'Definition of Boundary Regions' and then press 'add' three times to add three boundary conditions. By default, the newly added boundary regions are given the 'Wall' type of boundary conditions.

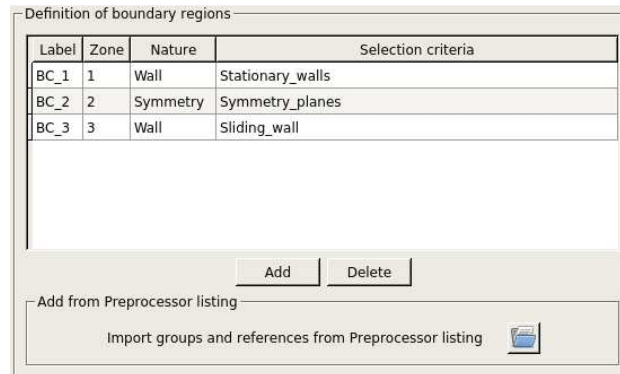
The first boundary in our problem is the side walls, which are considered stationary and non-slip. In the first row of the list of boundary regions, double click the cell in the 'Selection area' column in order to identify the boundary region in the mesh file. With the cursor in the cell, input the exact name of the boundary in the mesh file, i.e., 'Stationary_walls'.

The second set of boundaries is the symmetry planes which are used to enforce the 2D nature of the problem. In the second row, double click on 'Wall' in the 'Nature' column to activate the drop down menu of boundary types. Choose 'Symmetry' and release the menu. In the 'Selection criteria' cell, input the name of the boundary in the mesh file, i.e., 'Symmetry_planes'.

In the third row of the table, in the 'Selection criteria' column input the name of the sliding wall boundary in the mesh file, i.e., 'Sliding_wall'.

Note that it is best to press enter after typing in the name of the region to ensure that it is properly recorded by the GUI.

The boundaries and their global types are now defined, as shown in Fig. 4.25 below.



Label	Zone	Nature	Selection criteria
BC_1	1	Wall	Stationary_walls
BC_2	2	Symmetry	Symmetry_planes
BC_3	3	Wall	Sliding_wall

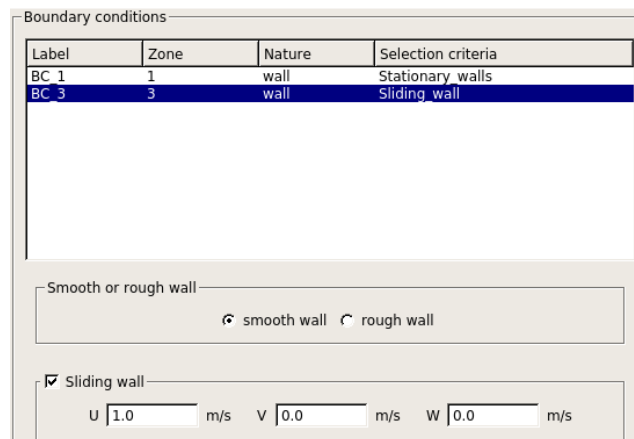
Add Delete

Add from Preprocessor listing

Import groups and references from Preprocessor listing

Figure 4.25: Boundary definition.

Having defined their type, the exact characteristics of the boundaries must be further specified. Click on the 'Boundary conditions' sub-folder which presents the list of boundaries which can be further specified (Fig. 4.26).



Label	Zone	Nature	Selection criteria
BC_1	1	wall	Stationary_walls
BC_3	3	wall	Sliding_wall

Smooth or rough wall

☒ smooth wall ☐ rough wall

☒ Sliding wall

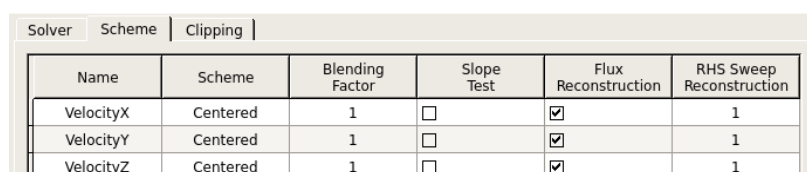
U 1.0 m/s V 0.0 m/s W 0.0 m/s

Figure 4.26: Specification of sliding wall.

The symmetry boundaries are fully defined and do not need further specification. Therefore, they do not appear in the list. However, 'Wall' boundaries can each be further defined as 'smooth' or 'rough', and as 'Sliding wall'. By default, the wall surface is 'smooth' and this parameter does not need to be changed. The 'Stationary_walls' boundary is fixed and fully specified. However, click on the 'Sliding_wall' boundary and activate the 'Sliding wall' selection. Fields then appear for the 'U', 'V', and 'W' velocity components of the wall. By default, these velocities are null. Click in the 'U' field and enter '1.0', as shown in Fig. 4.26.

The mesh and physics of the problem have now been set up. Now, parameters related to the calculation can be specified.

In 'Numerical Parameters', leave the settings unchanged in the 'Global parameters' sub-folder and move to the 'Equation parameters'. In the 'Scheme' tab, apply the 'Centered' scheme and deactivate the 'Slope test' for all three components of velocity (Fig. 4.27). The former will provide second-order accuracy in space and the latter is unnecessary for a smooth flow.



Name	Scheme	Blending Factor	Slope Test	Flux Reconstruction	RHS Sweep Reconstruction
VelocityX	Centered	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1
VelocityY	Centered	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1
VelocityZ	Centered	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1

Figure 4.27: Specification of the flux schemes.

To set the time step, move down the selection tree to the 'Pseudo-Time step' sub-folder. Change the 'Number of iterations' to 400 and increase the 'Maximal CFL number' to 8.0, leaving the other parameters unchanged (Fig. 4.28).

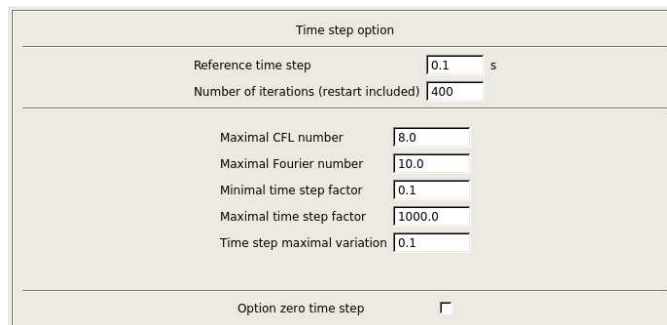
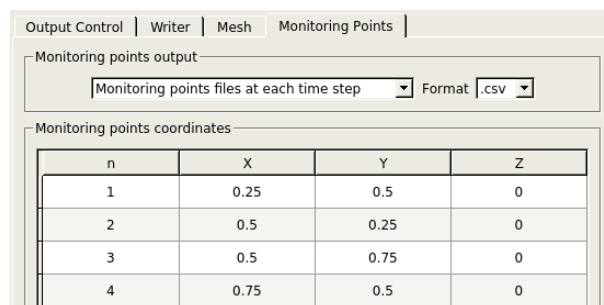


Figure 4.28: Selection of the number of iterations and CFL number.

Move to the 'Calculation control' folder. 'Time averages' are not of interest for these simulations and the sub-folder does not need to be visited. However, we want to keep track of the solution at different monitoring points to see how it evolves during the calculations. Aside from the solution residuals and the minimum and maximum values of the flow variables which *Code_Saturne* outputs during the calculations, tracking the solution at significant monitoring points is a very important means of gaining confidence in the convergence of the calculations and judging whether calculations have been run for a sufficiently large number of iterations. This is explained further in **Section 4.6** below.

Click on the 'Output control' sub-folder panel. The first three tabs, 'Output Control', 'Writer', and 'Mesh' are set by default to the correct values for this case and do not need to be changed. The 'Log frequency' in the 'Output Control' tab is set to print the calculations diagnostics such as the residuals at each time step. In the 'Writer' tab, the format of the 'results' file which will be used for post-processing is already set to 'Enight', which is compatible with the SALOME module 'ParaVis' which will be used to post-process the results after the run. The file will be located in the 'postprocessing' sub-directory of 'Reynolds1000/RESU/runDateAndTime/', where *runDateAndTime* corresponds to the time at which the run was started. Clicking on the 'results' row brings up the additional information about the 'Frequency', 'Time-dependency', and 'Options'. As already set, the 'results' file will only be written at the end of the run and on the 'Fixed mesh' used for this case. The 'Options' relate to the specific details of the file format. The 'Mesh' tab is already set to output the calculations data in all the fluid cells and at all boundary faces to the 'results' file.

In the 'Monitoring Points' tab, change the 'Monitoring points output' file format to '.csv' so that the files may be read in 'ParaView'. Using the '+' button, add four monitoring points situated at {0.25,0.5,0.0}, {0.5,0.25,0.0}, {0.5,0.75,0.0} and {0.75,0.5,0.0} (Fig. 4.29).



n	X	Y	Z
1	0.25	0.5	0
2	0.5	0.25	0
3	0.5	0.75	0
4	0.75	0.5	0

Figure 4.29: Definition of monitoring points.

In the 'Volume solution control', 'Calculation control' sub-folder, deactivate the 'Print in listing' for 'VelocityZ' but keep it active for 'Post-processing' (Fig. 4.30). Whilst listing the Z velocity in the calculations' 'listing' output file is not of interest, all three velocity vector components must be present in the 'results' file in order to visualise streamlines (and velocity vectors) with ParaView in the post-processing stage.

Solution control			
Name	Print in listing	Post-processing	Probes
Pressure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4
VelocityX	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4
VelocityY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4
VelocityZ	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4
total_pressure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4
LocalTime	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4
CourantNb	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4
FourierNb	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1 2 3 4

Figure 4.30: Selection of output variables.

Click on the 'Surface solution control' sub-folder and deactivate 'Post-processing' for 'Yplus' and 'Efforts' as they are not relevant to these simulations (Fig. 4.31).

Solution control	
Name	Post-processing
Yplus	<input type="checkbox"/>
Efforts	<input type="checkbox"/>
Efforts, tangential	<input type="checkbox"/>
Efforts, normal	<input type="checkbox"/>

Figure 4.31: Selection of output variables on the surfaces of the domain.

Lastly, we want to output one-dimensional profiles of variables along straight lines at the end of the calculations. Click on 'Profiles' and add two profiles which go through the centre of the Cavity. The first one for the X velocity along the Y axis and the second one for the Y velocity along the X axis. In turn, specify all the fields listed below the table of profiles, starting with 'Filename' and finishing with the variables which are to be stored on output. When all these fields have been added, press 'Add' above 'Filename' to add the profile to the list. Do not move to another folder or sub-folder in the GUI without first pressing 'Add' or you will lose all your selections.

For the X velocity profile, choose 'XVel_YaxisCentreLine' for both 'Filename' and 'Title', and '.csv' for 'Format' so that the profiles may be read in Paravis. The 'Output frequency' is already set to write the file only at the end of the calculation. To define the line, press on the 'Mathematical expression editor' button adjacent to 'Line Definition'. The line is defined by the equation $\{x = 0.5, y = t, z = 0.0\}$, where, by definition, t varies between 0 and 1.0. For the 'Number of points', enter 50 to account for the 50 cells across the domain's height. Finally, click on 'VelocityX' and use the right-pointing arrow to select it in the list of variables, and press 'Add' to store the profile in the list (Fig. 4.32).

Filename	Variables
XVel_YaxisCentreLine	VelocityX

Filename:
 Format:

Title:

Output frequency:

Line Definition:

Number of points:

VelocityZ
 VelocityX
 VelocityY
 FourierNb
 Pressure
 CourantNb
 total_pressure

VelocityX

Figure 4.32: 1D output profiles specification.

Repeat the procedure for the Y-velocity profiles, this time entering 'YVel_XaxisCenterLine' for 'Filename' and 'Title', $\{x = t, y = 0.5, z = 0.0\}$ for the equation, and selecting 'VelocityY'. Press 'Add' to complete the definition of the profiles. The 'Reynolds1000' case CFD simulation is now ready to run.

In the folder 'Calculation management', go directly to the 'Prepare batch calculation' sub-folder. By default, the calculation-restart is deactivated in 'Start/Restart' and so this sub-folder does not need to be visited. In the 'Prepare batch calculation' sub-folder panel (Fig. 4.33), the default selections of 'runcase' for the 'Script file', 'standard' (straight calculations without mesh or partitions import or pre-processing) and '1' processor for the 'Calculation script parameters' are already set.

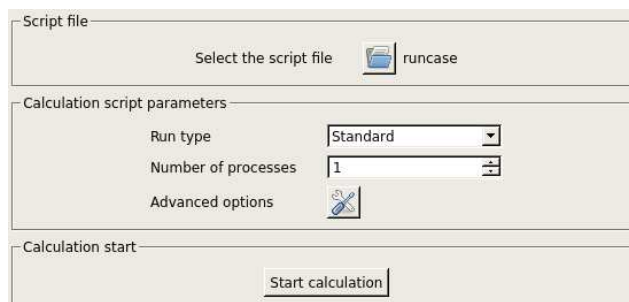


Figure 4.33: Batch calculation settings.

Save the case and press the 'Start calculation' button to run *Code_Saturne*.

4.6 Running and Analysing the Simulation

Upon firing the *Code_Saturne* run from the GUI, confirmation that *Code_Saturne* is running, "code_saturne is running", appears in a pop-up panel named after the time of the run. This is followed by further messages indicating what stage the calculation is in, from "Preparing calculation data" to "Saving calculation results".

Wait for the calculations to complete and enter the 'Reynolds1000/RESU' directory or open its contents via a browser to inspect its contents. Explanations of the meaning and purpose of the different output files and directories resulting from a run are available in the *Code_Saturne* Users Guide [3] and are not repeated here. Instead, we highlight individual items which relate to this specific run and how the output information should always be used in order to analyse a calculation.

The 'RESU' directory now contains a new directory named after the date and time at which the calculation was started expressed on a 24 hours clock, in the format "YearMonthDay-HourMinutes".

In this latter directory, note in particular:

- the profile files 'XVel_YaxisCenterLine' and 'YVel_XaxisCenterLine', written in '.dat' format,
- the 'listing' file,
- the 'monitoring' and 'postprocessing' directories

With your text editor, open one of the profile file to inspect its structure. The requested variables are listed in column format as a function of the {x,y,z} coordinates of the points along the profile line defined by in the GUI.

Open the 'listing' file to check that the calculation has converged towards stable values. The minimum and maximum values of the solution variables and the solution residuals for these variables are listed in summary tables at each iteration. By comparing the initial and final values in the 'derive' column for each variable, and their evolution throughout the calculation, check that the residuals have decreased by at least two orders of magnitude (Figs 4.34a-b).

** INFORMATION ON CONVERGENCE					** INFORMATION ON CONVERGENCE				
Variable	Rhs norm	N_iter	Norm. residual	derive	Variable	Rhs norm	N_iter	Norm. residual	derive
c Pressure	0.22286E-03	152	0.26549E-04	0.10000E+01	c Pressure	0.83182E-11	0	0.11848E-06	0.13010E-07
c VelocityX	0.24601E-03	24	0.52360E+00	0.10027E+00	c VelocityX	0.28547E-03	0	0.10042E-06	0.00000E+00
c VelocityY	0.24601E-03	24	0.52360E+00	0.92055E-03	c VelocityY	0.28547E-03	0	0.10042E-06	0.00000E+00

a : After 1 iteration

b : After 400 iterations

Figure 4.34a-b: Convergence history from the 'listing' file.

Inspect one of the monitoring files in the 'monitoring' directory. The value of each variable is recorded in separate files at all the monitoring points defined in the GUI and as a function of iteration.

Finally, verify that the 'RESULTS.case' file containing all the mesh and output variables information for post-processing has been output in the 'postprocessing' directory.

Having validated the calculation itself, you can now proceed to examining and post-processing the results by returning to the SALOME platform.

4.7 Post-processing the Results

In SALOME, select 'ParaViS' from the drop-down module selector in the top menu bar. The name of the module will add itself to the Object Browser list and the ParaView-specific panels and menus will be activated, including a new 'ParaView scene viewer' window.

Before loading the run data in ParaView, modify the default colour schemes. To visualise scenes on a white background, which is advantageous for printing, change the default settings for ParaView by clicking on **File → Preferences** in the top menu. For 'ParaViS', in the 'ParaView Settings' tab, change the colours to black for the foreground and text, white for the background, and grey for the surfaces, as shown in Fig. 4.35 below. Press 'Apply' to enforce the new settings and 'OK' to validate your selection when you are satisfied with the changes.



Figure 4.35: Setting the colour preferences for ParaView.

The data to post-process can now be imported in ParaView.

First, you are going to load the monitoring point data to validate that a stable, steady-state solution has been obtained. From the top menu bar, select **File → Open ParaView File**. In the 'Open File' pop-up panel, navigate to the 'monitoring' sub-directory and select three files. The multiple selection is performed by holding the Ctrl key down as you select the files. Select 'probes_Pressure.csv', 'probes_VelocityX.csv', and 'probes_VelocityY.csv'. Close the panel by clicking 'Ok'. The three sets of data are now displayed under their file name in the 'Pipeline Browser'. For each file, press 'Apply' to load the data. By default, the data is visualised in tabular form in the ParaView scene viewer. Close the View by clicking the cross button at the top, right hand corner of the view. The data view is now replaced by a 'Create View' menu at the centre of the view pane (Fig 4.36).

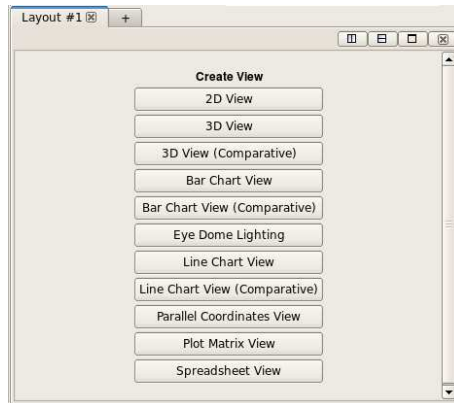
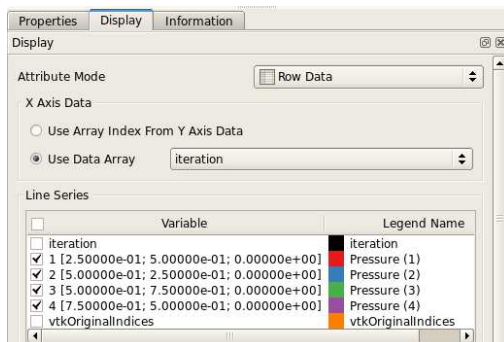
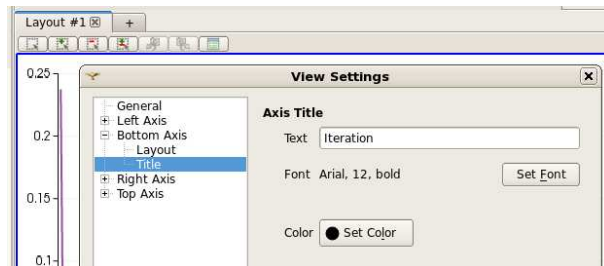


Figure 4.36: ParaView 'Create View' menu.

Click on the 'Line Chart View' button and prepare the line plots, starting with the pressure data. Make the data visible for that file by clicking on the eye symbol next to the file name in the 'Pipeline Browser'. Click on the 'Display' tab and set 'Attribute Mode' to 'Row Data'. In the 'X Axis Data' category, select 'Use Data Array' and choose 'iteration' as the abscissa. In the 'Line Series' category, unselect the 'iteration' and 'vtkOriginalIndices' variables. The variables '1', '2', '3', and '4', representing the pressure at each iteration at the four monitoring points you specified should be selected and the graphs of their evolution as a function of the number of iterations (time steps) should now be displayed in the view window. For clarity, change the legend for each variable. In the 'Line Series' category, left-click on the row for variable '1' to select it. Left-click again on the name of that variable, in the 'Legend Name' column to edit it. Change the name of each variable to 'Pressure (1)', 'Pressure (2)', 'Pressure (3)' and 'Pressure (4)' (Fig. 4.37a).



a : variables' legend



b : axis titles

Figure 4.37a-b: Specification of line plot's legend and axes.

In the view window's tool bar (underneath the window tab), click on the 'Edit View Options' button to bring up the 'View Settings' pop-up panel. Change the 'Bottom Axis' title to 'Iteration', and the 'Left Axis' title to 'Pressure (Pa)' (Fig. 4.37b). The graph shows that the pressure at the four monitoring points becomes constant after about 100 iterations (Fig. 4.38).

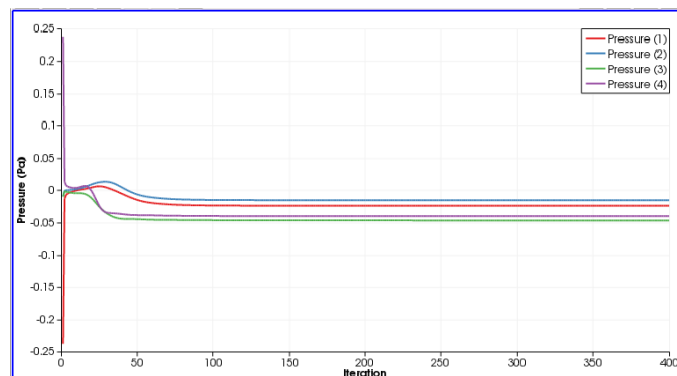


Figure 4.38: Pressure at the four monitoring points as a function of iteration.

Repeat the same operation for the remaining velocity files. For both files, change the left axis title to 'Velocity (m/s)'. For the X velocity, change the variable names to 'X Vel (i)', where $i = 1, 2, 3$, and 4. Keep the default colours for each line so that they are consistent for each monitoring point. For the Y velocity, change the variable names to 'Y Vel (i)', where $i = 1, 2, 3$, and 4. The graphs also confirm that the velocity components at the four monitoring points settle to a stable, constant value after about 100 iterations, indicating that the calculations are converged (Fig. 4.39a-b).

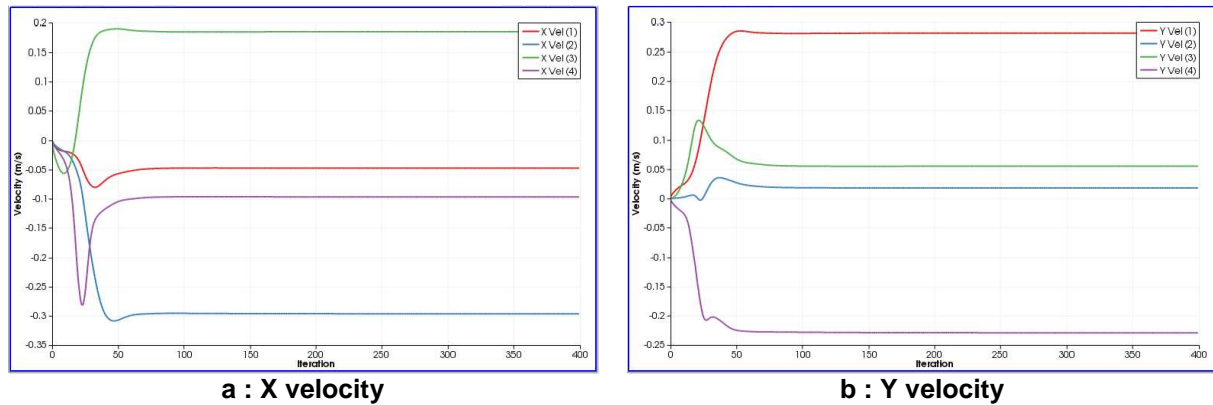


Figure 4.39a-b: Velocity components at the four monitoring points as a function of iteration.

You may now post-process the results from the calculations.

From the top menu by selecting **File** → **Open ParaView File**, or by right-clicking on the 'builtin' object in the Pipeline Browser panel and selecting 'Open', access the pop-up 'Open File' panel and choose the 'RESULTS.case' file from your run. The 'RESULTS.case' object will be added to the 'Pipeline Browser' and its contents displayed in the 'Object Inspector' panel underneath. Press 'Apply' to load the data in ParaVis.

Next, to extract the computed data in terms of fluid domain and boundary data, in the top menu bar click on **Filters** → **Alphabetical** → **Extract Block**. The new object 'ExtractBlock1' now appears in the 'Pipeline Browser'. With the object highlighted, move to the 'Block Indices' panel of the 'Properties' tab. In the data tree, under 'Root', select 'Fluid domain' to visualise the data calculated at the cell centres inside the cavity walls.

Left click on the 'Display' tab of the 'Object Inspector'. In the category 'Color', next to the 'Color by' heading two drop-down menus allow you to choose the variable to visualise. Choose 'Velocity' in the first one and 'Magnitude' in the second one. Contour plots of the velocity magnitude are then displayed in the 'ParaView scene viewer'. By default, the colour scale is set to the RGB colour scheme which is inadequate.

To modify the visualisation colour scheme, click on the button 'Edit Color Map...' in the 'Color' category of the 'Display' tab and bring up the 'Color Scale Editor' pop-up panel (Fig. 4.40). In the 'Color Scale' tab of the editor, click on the 'Choose Preset' button and select 'Blue to Red... HSV' in the list of 'Preset Color Scales'. Press 'OK'.

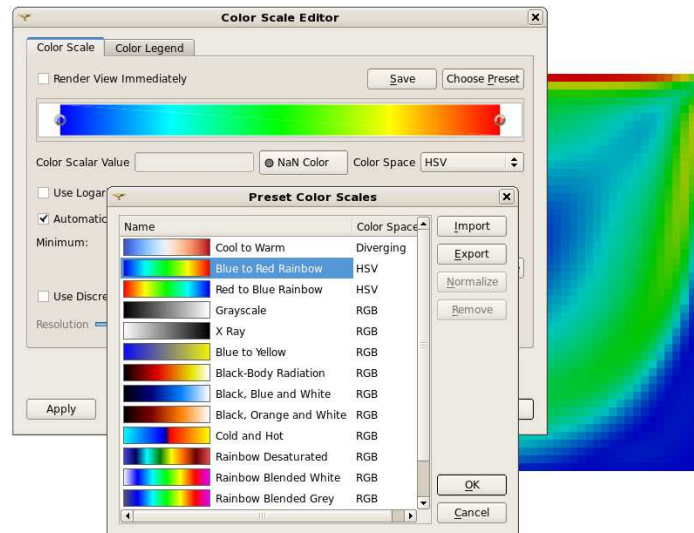


Figure 4.40: Selection of visualisation colour defaults.

Ensure that 'Automatically Rescale to Fit Data Range' is active and press 'Make Default' to save the changes and 'Close' to leave the 'Color Scale Editor'. The contour plot of velocity magnitude is now updated for the new colour scale (Fig. 4.41).

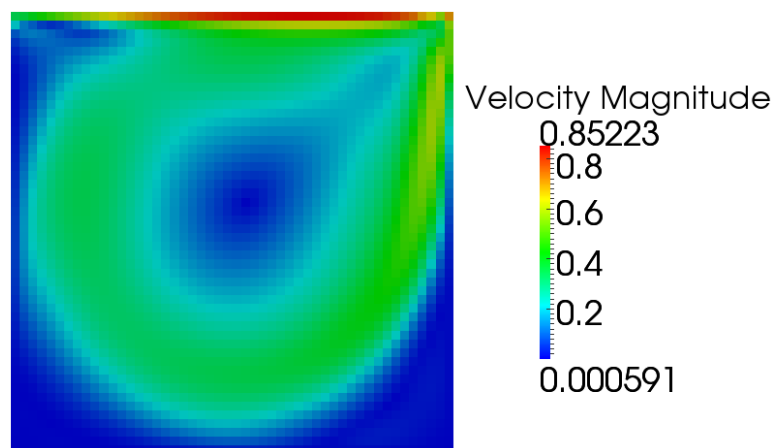


Figure 4.41: Contour plot of velocity magnitude.

The contour plot indicates that there is a zone of higher velocity flow defined by the green and red zones surrounded by lower and no-velocity regions in blue. Consistent with the chosen boundary conditions, the maximum velocity occurs next the sliding wall and the flow is almost stagnant near the other, non-slip walls.

With the HSV colour scale, the contour plot is now clear, with the different levels clearly differentiable, but the image looks tessellated. As *Code_Saturne* outputs data at cell centres in the 'results' file, in ParaView each mesh cell is painted with a pixel of colour corresponding to the exact value of the variable in the cell. Whilst this cell-data visualisation mode is correct to examine exact values at cell centres, it can yield ragged images unrepresentative of the solution's higher-order spatial accuracy and cannot be used in ParaView to generate vectors and streamline plots. Instead, to produce smoother images, and vector and streamline plots, the cell-data can be interpolated to cell vertices (corners) using the ParaView 'Cell Data to Point Data' filter.

Having clicked on 'ExtractBlock1' in the Pipeline Browser to select it, in the top menu bar click on **Filters** → **Alphabetical** → **Cell Data to Point Data**. The new object, which now stores the fluid domain data interpolated to cell corners, is added in the Pipeline Browser (Fig. 4.42).

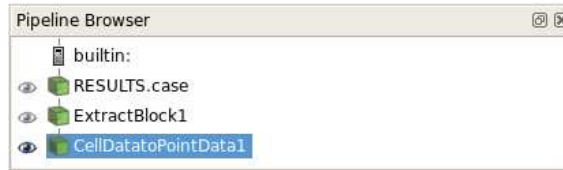


Figure 4.42: CellDatatoPointData filter added to the Pipeline Browser.

With 'CellDatatoPointData1' selected in the Pipeline Browser, click 'Apply' in the 'Properties' tab of the 'Object Inspector' to bring up the smoothed-out contour plot in the 'ParaView scene viewer'.

Picking the different variables to map, either from the 'Display' tab in the Object Inspector for the 'CellDatatoPointData1' object or directly from the drop-down list in the menu bar for contour plots at the top of the SALOME window, create contour plots of the X and Y velocity components, as shown in Figs. 4.43a-b below. The plots use a 'Surface' mode of 'Representation', which can be selected either from the 'Display' tab and 'Style' category, or directly from the drop-down list in the contour plot menu bar.

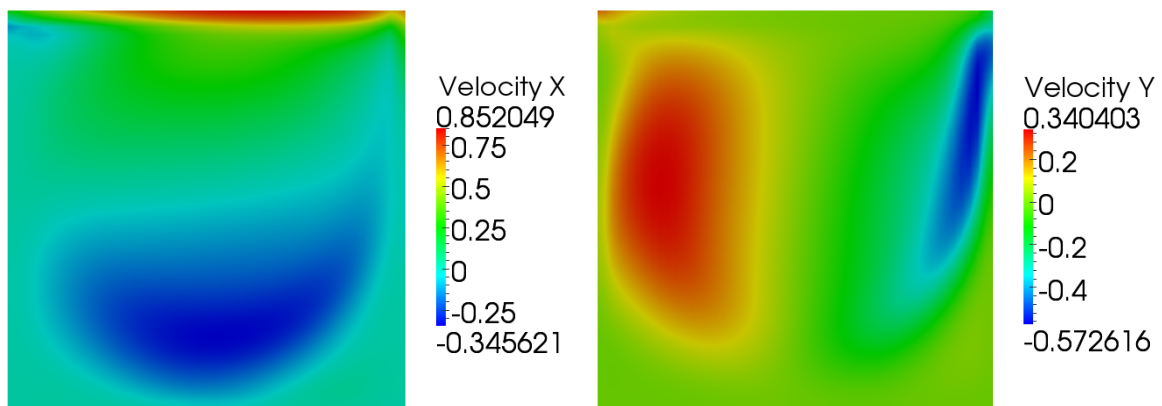


Figure 4.43a-b: Contour plots of the X (left) and Y (right) velocity components.

As expected, the maximum velocity in the X direction occurs at the top wall, where it nears 1.0 m/s (Fig. 4.43a). The locations of the blue (flow in the negative direction) and red (flow in the positive direction) areas in both plots (Figs 4.43a and 4.43b) indicate that the flow is entrained by the top wall in a clockwise circular motion inside the box. These plots are useful to verify overall velocity magnitudes and patterns but, to get a better visualisation of the flow, next create streamlines which will show fluid particles trajectories.

To make it possible to locate the streamlines with regard to the Cavity, you are going to create a combined image showing the streamlines superimposed on top of the mesh.

First, create the streamlines. With 'CellDatatoPointData1' selected in the Pipeline Browser, click on **Filters→Common→Stream Tracer** in the top menu. The 'StreamTracer' object is then added under 'CellDatatoPointData1' in the Pipeline Browser. Keep the 'CellDatatoPointData1' object selected and visible ('eye' icon in bold next to the object). In the Object Inspector, select the 'Display' tab. In the 'Color' category, choose 'Solid Color' in the 'Color by' drop-down list. In the 'Style' category, change the 'Representation' to 'Wireframe' and decrease the 'Opacity' to 0.1 (Fig. 4.44).

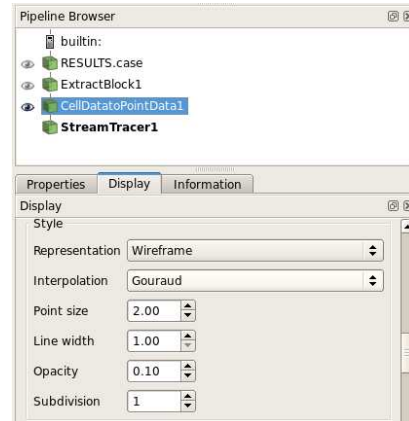


Figure 4.44: Selection of opacity.

The mesh lines displayed in the 'ParaView scene Viewer' should now display in faint, black colour.

Next, select the 'StreamTracer1' object in the Pipeline Browser. In the 'Object Inspector', select the 'Properties' tab and modify the default settings for 'Seeds'. For a 'Point Source' 'Seed Type', modify the X and Y coordinates of the seed point to 0.15 and 0.05, respectively. Request 60,000 points ('Number of Points') and a radius of 1.0 ('Radius'), as shown in Fig. 4.45 below. Press 'Apply' to validate your changes.

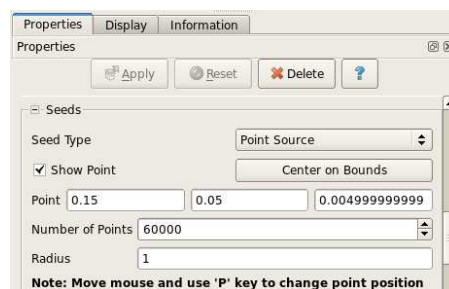


Figure 4.45: Selection of streamlines settings.

In the 'Display' tab, choose 'Velocity' and 'Magnitude' from the 'Color by' drop-down lists of category 'Color' and visualise the streamlines coloured by velocity magnitude, on top of the computational mesh (Fig. 4.46).

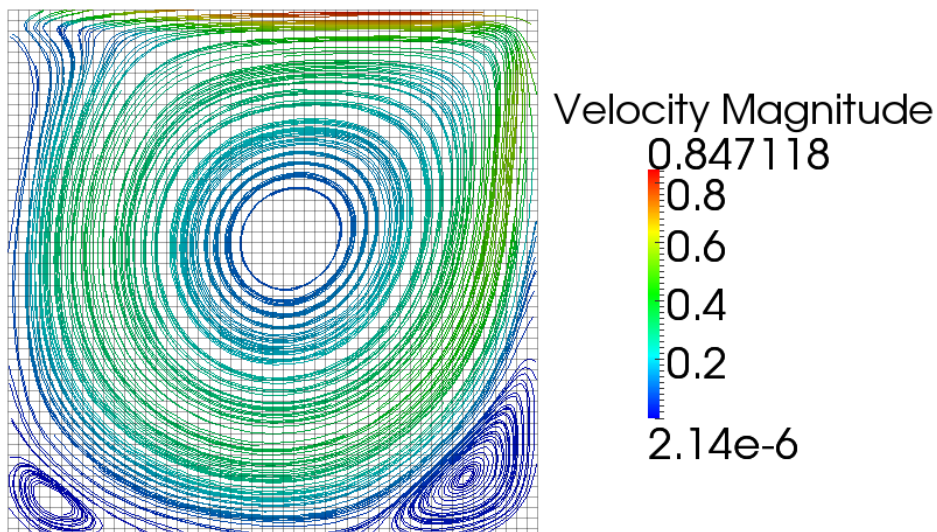


Figure 4.46: Streamlines coloured by velocity magnitude superimposed on the mesh.

In addition to the general circular motion which had been inferred from the contour plots of velocity components, the streamlines now reveal two recirculation pockets in the bottom corners of the Cavity.

To further compare the calculated data with the available data from [4], create line plots of the X and Y velocities along the Y-axis and X-axis profiles which were set in the *Code_Saturne* GUI. Using the menu **File→Open ParaView File**, import both the 'XVel_YaxisCentreLine.csv' and the 'YVelXaxisCentreLine.csv' files. To read in the data and ensure that the velocities are correctly interpreted as real numbers by ParaView, in the Properties tab, define the 'Field Delimiter Characters' as ' , ': a blank space followed by a comma. Also, select 'Merge Consecutive Delimiters' (Fig. 4.47).



Figure 4.47: Properties specification for the velocity profiles.

Go to the 'Information' tab and verify that all the data arrays are read in as 'double'. The steps to create and label the graphs are similar to those already detailed for the graphs of the monitoring points' data and they are not repeated here.

For direct comparisons with the plots from [4], for the graphs of X velocity, plot the 'Y' coordinate as a function of 'VelocityX'. For the graphs of Y velocity, plot 'VelocityY' as a function of the 'X' coordinate. The reference data provided in [4] for the fine mesh results (129x129) is listed in **Appendix A**. The data can be copied to a '.csv' file which is then imported in ParaView at the same time as the 'XVel_YaxisCentreLine.csv' and the 'YVelXaxisCentreLine.csv' files. Set the plots so that the *Code_Saturne* and the reference results are visualised in the same view, and display the reference points with a marker rather than a line. Each data set can then be compared directly with the reference results (Fig. 4.48a-b).

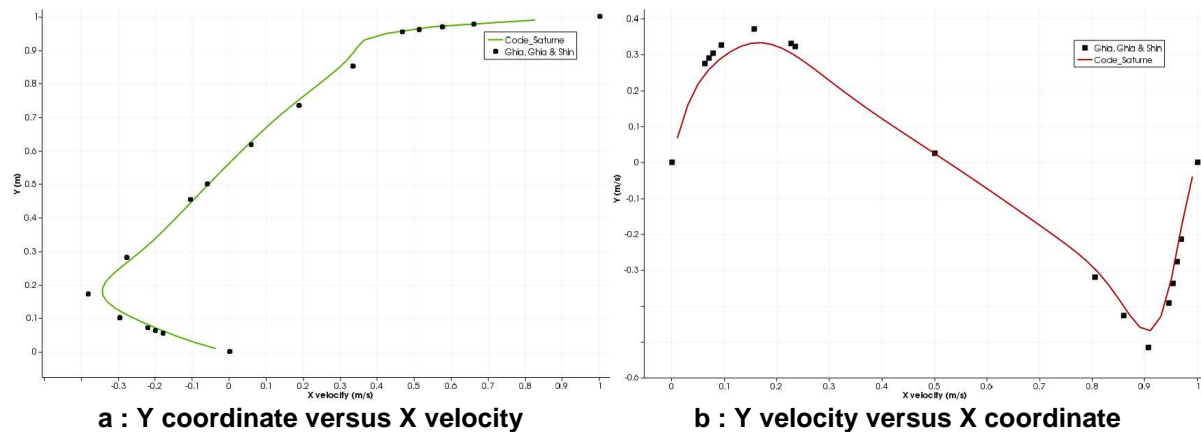


Figure 4.48a-b: Velocity profiles. Comparison between *Code_Saturne* (solid lines) and reference [4] results (circles).

Overall, good agreement is obtained with the reference results [4], even though the results with *Code_Saturne* were obtained on a coarser mesh. Running on a finer mesh would make it possible to capture the velocity extremas with increased accuracy.

5 References

- [1] www.salome-platform.org
- [2] *Code_Saturne: a Finite Volume Code for the Computation of Turbulent Incompressible Flows - Industrial Applications*, International Journal on Finite Volumes, Vol. 1, 2004.
- [3] www.code-saturne.org



[4] U. Ghia, K.N. Ghia, and C.T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, Journal of Comp. Phys., Vol. 48, pp. 387-411, 1982.

Appendix A – Reference Data from [4]

To make it easier to replicate the comparative plots of the *Code_Saturne* and reference results, the data from [4] (Tables I and II) at $Re = 1000$ is reproduced in Table A.1 below. The reference results listed in [4] were obtained on a fine, 129x129 mesh.

Y (m)	Ux (m/s)	X (m)	Uy (m/s)
1.00000	1.00000	1.00000	0.00000
0.97660	0.65928	0.96880	-0.21388
0.96880	0.57492	0.96090	-0.27669
0.96090	0.51117	0.95310	-0.33714
0.95310	0.46604	0.94530	-0.39188
0.85160	0.33304	0.90630	-0.51550
0.73440	0.18719	0.85940	-0.42665
0.61720	0.05702	0.80470	-0.31966
0.50000	-0.06080	0.50000	0.02526
0.45310	-0.10648	0.23440	0.32235
0.28130	-0.27805	0.22660	0.33075
0.17190	-0.38289	0.15630	0.37095
0.10160	-0.29730	0.09380	0.32627
0.07030	-0.22220	0.07810	0.30353
0.06250	-0.20196	0.07030	0.29012
0.05470	-0.18109	0.06250	0.27485
0.00000	0.00000	0.00000	0.00000

Table A.1: X velocity versus Y at the vertical mid-plane and Y velocity versus X at the horizontal mid-plane. Data reproduced from [4]. $Re = 1000$. 129x129 mesh.