I declare that,

- The work in this Degree Thesis is completely my own work,
- No part of this Degree Thesis is taken from other people's work without giving them credit, all references have been clearly cited,
- I'm authorised to make use of the research group related information I'm providing in this document.
- I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by *The Universitat Politècnica de Catalunya - BarcelonaTECH*.

Santiago Guarner Escribano 30/06/2020

Student Name Signature Date

Title of the Thesis : Study of fluid-dynamics applications by using on-web-browser software.

BACHELOR'S THESIS

AEROSPACE TECHNOLOGY ENGINEERING
ACADEMIC YEAR: 2019-2020 Q2

Study of fluid-dynamics applications by using on-web-browser software

Guarner Escribano, Santiago

Director: Prof. Pedro Javier Gamez Montero
Co-director: Prof. Roberto Castilla

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

July 2020

Abstract

This thesis provides an introduction to computational fluid dynamic using on-web-browser software. Therefore, it presented two classical cases so that the students apply the theory learned in the lessons and, also, an investigation has been done in order to optimize the meshing procedure with SimScale. In addition to this, in the last part of the project, the learner is introduced to the compressible flow and it is described the problems found with the software and some solutions. The two cases are the Pipe Flow and the Flat Plate Flow. Every time, it has been imperative to follow the typical procedure; pre-processing, simulation and post-processing in order to underline to the learner the importance of the interpretation of the results and the sensitivity of the simulations.

Contents

List of Figures

List of Tables

1 Introduction

In this technological world with limited resources, simulation has become one of the key points to predict future situations and prevent bad designs, waste of resources and even catastrophic situations. In particular, fluid dynamics is a very complicated subject and CFD is an excellent tool to help understand problems that were out of our reach. These simulations require a certain degree of knowledge in the area to prepare the study to be done and to come up with some conclusion from the results. The mesh is one of the most important parts (if not the most) in CFD simulation. Creating the right mesh for the right situation is imperative to use this tool. Therefore, it is very important that the students learn how to create and optimize simulations in order to improve the results, adjust them as much as possible to reality and save calculation time. Taking into account this, it is added that the on-web-browser software is growing so that the user does not need very powerful computers to do big calculations and to store the heavy results.

1.1 Aim of the Project

The main objective of this project is to develop an introductory and complete user guide for SimScale [1] including explanations and detailed considerations for new learners, specially bachelor students. In addition to this, a meticulous study of the different types of meshing for different flows (internal, external and compressible) will be performed.

1.2 Project requirements

The requirement bulleted for this project are the following:

- Deliver a free introductory and complete user guide to SimScale.
- Study with SimScale the meshing procedures and methods needed for a particular case flow. These cases will involve internal and external.
- Compilation of problems and possible solutions for a compressible flow through a nozzle.
- Deadline: June 30th

1.3 Scope of the project

The project is divided in four parts, each of the first three is focused in one type of flow. The fourth will a finale case study for the learner to assimilate all the concepts. This bachelor's thesis have the following structure.

1. **Introduction to Simscale** In this introduction, the online software will be presented to the user. In order to do that, it has been adapted the following case; "Lid-Driven Cavity problem". It is a simple problem, great to understand how to use the basic functions of this tool.

2. **Internal Flow** The second section is related to the internal flow inside a pipe. The article [2] will be taken as a reference to perform the different kind of meshing a simple geometry as it is a circular straight pipe. In this study the laminar and turbulent Flow are considered.

    - **Laminar Flow**: this is the simplest case of all, where with a homogeneous mesh it will be enough to describe accurately the effects related to this regime.
    - **Turbulent Flow:** for this regime it will be necessary to mesh the pipe with two approaches, one will be homogeneous and the other one, exponential.

    Once the simulations are performed, the analytical and numerical results will be compared to validate the results with each method used. This validation will include the graphical representation of the variation of pressure along the pipe, the variation of u^+ over y^+, the shear strength along the outlet section over the y and y^+ and the velocity profile analysis. Due to the basic post-processor of SimScale, the post-process analysis will be done with ParaView, a very powerful tool designed to help us with this part.

3. **External Flow**: it will be simulated a flow through a thin plate. An exponential method will be used for the mesh, very important to see the effect of the boundary layer. The validation process will involve the analysis of the variation of the boundary layer along the plate, the variation of the velocity profile and the shear strength and friction coefficient along the plate. All the results obtained form the simulations will be compared with the ones known form [3] and [4].

4. **Compressible Flow:** In this section, a simple nozzle will be used. This kind of flow is much more sensible to the boundary conditions and the solver selection than the others, for this reasons is harder to obtain good results. The aim of this part is to describe some problems found simulating a compressible flow through a nozzle and some possible solutions.

The main focus of the report is the meshing process and post processing. The fluid dynamic theory has been over and over explained by different authors as we can observe in many books and there is no need to go deeper. What is needed is to understand better the CFD and improve our simulations to be more efficient with this new technology.

1.4 Justification and Utility

This project tries to give a solution to two problems by using SimScale as the software; studying the best mesh configuration to any particular case and helping the student to acquire some competences that will be essential for his future. It has been chosen this tool because is a computer-aided engineering (CAE) software product based on cloud computing and allows Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEM). With this product it is possible to do CFD, FEA and Thermal simulation from an online platform. The main advantages are no downloads, installs, license keys, service packs or compatibility issues. This means that it is available for everyone and this kind of software will be very common in a near future.

So, introducing the student to a, potentially, very powerful tool and with a lot of future is excellent for his academic development. Understanding better the CFD and improving our simulations to be more efficient with this new technology is key. SimScale comes directly from the well-known Open Foam, this is the reason why there is a lot of information on the meshing modelling as it is possible to find in [5] and how is adapted in different situations. Nevertheless, it has not previously done with a software as SimScale or a similar one and, furthermore, it has not been done in a compressible flow as it is intended to do in this project. Taking this investigation and adapt it to give the student the opportunity to find by himself and learn the complexity of the simulations is an excellent way to put at the service of the others our small knowledge in this area.

1.5 State of the Art

There are many software on the market based on the Finite Element Method and focused in Fluid Dynamics, Structural Mechanics and Thermal Analysis. The main products are the following:

SimScale: the cloud-based platform was released in 2013. As explained in the next subsection, is a partially gratuitous computer-aided engineering (CAE) software product based on cloud computing and allows Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEM).

Onshape is a modern computer-aided design (CAD) software system, also cloud computing, primarily focused on mechanical CAD (MCAD) and is used for product and machinery design. It allows teams to collaborate on a single shared design, the same way multiple writers can work together editing a shared document via cloud services. It has three main plans available for the general public; Enterprise, Professional and Standard. All of them with an anual price.

OpenFoam: Open-source Field Operation And Manipulation is a C++ toolbox created for the solution of continuum mechanics problems, including computational fluid dynamics developed since 2004. It is also supplied with pre- and post-processing environments. It does not have an graphical user interface which makes this software to be not friendly to the new users. The operating systems where is available are Windows, Linux and Unix. OpenFoam releases are scheduled every 6 months.

Ansys is an American publicly traded company that develops and markets finite element analysis software used to simulate engineering problems. Particularly, **Ansys Fluent** is the very powerful CFD software package. It contains the broad, physical modeling capabilities needed to model flow, turbulence, heat transfer and reactions for industrial applications.

SolidWorks is a full CAD package with different modules. FEA is applied through the SolidWorks Simulation module. The module comes in three levels: Standard, Professional and Premium. You can perform Static studies, Fatigue studies, Motion analysis, Thermal analysis, Frequency studies, Buckling studies, Pressure vessel studies, Topology studies, Linear dynamic studies, Non-linear analysis and also CFD tests. The main problem is that the system requirements are quite high.

Adina is a commercial engineering simulation software program that is developed and distributed worldwide by ADINA R & D, Inc. ADINA is used in industry and academia to solve structural, fluid, heat transfer, and electromagnetic problems. ADINA can also be used to solve multi-physics problems, including fluid-structure interactions and thermo-mechanical problems. ADINA CFD is capable of modeling a wide array of fluid flows, including those in the laminar and turbulent regimes, thin-film Reynolds flow with smooth or rough boundaries, two-phase flow, non-isothermal flow and conjugate heat transfer, porous-media flow, flows with mass transfer, low- and high-speed compressible conditions, and comes equipped with material models for handling non-Newtonian fluids and real gases. General flow conditions in arbitrary geometries can be solved.

Siemens Solid Edge is a 3D CAD, parametric feature (history based) and synchronous technology solid modeling software. FEA capabilities in Solid Edge include simulation of individual parts, assembly analysis and computational fluid dynamics (CFD). With this CAD and Finite Element Analysis software it is possible to perform Stress analysis and simulations, Vibration simulations, Full motion simulations, Buckling simulations, Thermal simulations and also CDF tests.

PTC Creo: Creo is another well-known company in the design and engineering community. Creo runs in Windows and offers scalable 3D CAD product development packages and tools. Those tools feature modelling and design, simulation and analysis, among others. With Creo 6.0 it is possible to run Structural analysis, Thermal tests, Motion analysis, Fatigue simulation, Mould fill analysis and CDF analysis.

Code_Saturne is a free, open-source software developed and released by EDF to solve computational fluid dynamics (CFD) applications. It accepts and solves Navier-Stokes equations for 2D, 2D-axisymmetric and 3D flows, steady or unsteady, laminar or turbulent, incompressible or weakly dilatable, isothermal or not, with scalars transport if required. It is based in a finite volume approach that handles meshes with any type of cell and any type of grid structure. Several turbulence models are available, from Reynolds-Averaged models to Large-Eddy Simulation models.

SU2 is a suite of open-source software tools written in C++ for the numerical solution of partial differential equations (PDE) and performing PDE-constrained optimization. The SU2 team is making multi-physics analysis and design optimization software freely available and involving everyone in its creation and development.

Basilisk is a Free Software program for the solution of partial differential equations on adaptive Cartesian meshes describing fluid flow. It is destined to be the successor of **Gerris** and is developed by the same authors.

ParaView is an open-source, multi-platform data analysis and visualization application. The users can quickly build visualizations to analyze their data using qualitative and quantitative techniques. The data exploration can be done interactively in 3D or programmatically using ParaView's batch processing capabilities. This platform can handle extremely large data-sets using distributed memory computing resources. Is the perfect post-processor to use with SimScale.

1.6 About SimScale

This software is a computer-aided engineering (CAE) software product based on cloud computing. SimScale was developed by SimScale GmbH and allows Computational Fluid Dynamics, Finite Element Analysis and Thermal simulations. The backend of the platform uses open source codes as Code_Aster, CalculiX and OpenFOAM. The cloud-based platform of SimScale allows users to run more simulations, and in turn iterate more design changes, compared to traditional local computer-based systems [1]. It is an all-in-one simulation platform across CFD, FEA, and Thermal Analysis. It is very simple because there is no need to install updates, have maintenance or introduce any kind of licence keys. The basic plan is free for everyone and it is perfect for the beginners. It has even more advantages among which it is remarkable; it is an Easy-to-Learn interface, it has a live costumer support, with a big community. The online forum is very useful to share projects and to work together to find solution to new problems. Thanks to the online source, you can design your simulations and run them form anywhere. The software supports all standard 3D files to help the user to work with the CAD he chooses. Finally, it has his own post-processing tool, still in development, that can help the user to avoid any other programs. Nevertheless, in this thesis, it will be used ParaView as the post-processor, because is much more complete.

2 Introduction to SimScale

During this section, the student will be introduced to SimScale with a simple step by step tutorial. In every step, this tutorial will try to explain the reason for every choice in order to ensure an understanding and future autonomy of the user.

2.1 Turbulence Models

Before introducing the student into the this tool, it will be presented the most common used turbulence models available in SimScale. It is very important to select the right model for each simulation because the results depends on them and each one behaves differently in a specific situation. None of the developed models is universally applicable to all flow conditions. Though each group has certain advantages and strengths [6].

2.1.1 K-Epsilon Standard Turbulence Model

This model is the most common to simulate the mean flow characteristics for fully turbulent flow conditions [7]. It is based on the Boussinesq approximation [6] and it gives a general description by means of two transport equations. It has the kinetic energy (k_t) equation and another equation based on the dissipation of the turbulent kinetic energy (ε). The parameters are formulated as:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon}$$

$$k_t = \frac{c_p \mu_t}{Pr_t}$$

Where μ_t is the turbulent viscosity. The k-ε model is shown to be applicable for free-shear flows, such as the ones with relatively small pressure gradients. The advantages of this model, among others, are robustness, easy implementation and low computational cost. It does not perform well with rotational flows, inlets or compressors [8]. Modeling flows close to solid walls requires integration of the two equations over a fine grid in order to correctly capture the turbulent quantities inside the boundary layer as well as the corrections for low Reynolds number effects [6] .

2.1.2 K-Omega Standard Turbulence Model

This model uses, as well, two equations. The first equation is the kinetic energy nevertheless, in this case, the second equation is the transport of turbulence and the parameter is the specific ratio of dissipation of turbulent kinetic energy (ω). The equation for ε is replaced by an equation for ω. The new parameter is defined as [6]:

$$\omega = \frac{\varepsilon}{C_\mu k}$$

The ω equation is easier to integrate. This model performs better near the wall than the k-ε model and handles laminar-turbulent transitions. It is very sensible to turbulence on the inlet and the free stream. Requires a refined mesh near the wall in order to resolve the viscous sub-layer.

2.1.3 K-Omega SST Turbulence Model

K-omega Shear Stress Transport model is one of the most commonly used models. It combines the advantages of the previous from k-ε and k-ω models. The transport variable k determines the energy in turbulence and ω determines the scale of turbulence [9]. It can be used for boundary layer problems because the formulation works from the inner part through the viscous sub-layer, till the walls [7]. The SST model has the ability to account for the transport of the principal shear stress in adverse pressure gradient boundary-layers. To resolve the sub-layer, a high resolution mesh is required. It has shown a good behaviour in adverse pressure gradients and separating flow. Nevertheless, it produces large turbulence levels in regions with large normal strain, like stagnation regions and regions with strong acceleration.

2.2 The Lid-Driven Cavity Case

To introduce the student into the work tool that it will be used during the cases presented, it has been adapted the following case from OpenFoam.
Before the case is developed, it is necessary to create a SimScale® account and take the first tour following these instructions:

SIMSCALE

Creating a SimScale Account

Unlike most FEA & CFD software, you don't have to download or install anything to use SimScale. But, you do have to create an account. Once you have an account, you can log into SimScale from any computer as long as you have an internet connection. Follow the steps below to create your account:

1. Go to https://www.simscale.com/
2. Click on the 'CREATE FREE ACCOUNT' button at the top of the page.
3. Answer a few questions and click 'Create Account'.
4. Check your email for a confirmation email and follow the link to activate your account.

Signing In

1. Navigate to https://www.simscale.com/ and click on the 'LOGIN' button at the top of the page.
2. Enter your login information.
3. Click 'Login'
4. Choose a username and click 'SAVE'. Your unique username will be associated with your profile and activity such as public projects, comments, or posts in the SimScale CAE forum.
5. Help us get you started easily by filling in the interest fields. Then click 'Get Started'
6. Select the SimScale Community Plan and click 'Start For Free'
7. Start the 2-minute tour to see the most important features of the SimScale Community.

Fluid Flow Tutorial for CFD

8. At the end of the tour, you have the option to start one of our 3 Getting Started tutorials. Select the **Fluid Flow Simulation (CFD)** tutorial and click 'Start Tutorial'
9. The project will automatically open in your SimScale Workbench. The tutorial instructions will open in a new pop-up window.
10. Follow the tutorial to complete your first pipe flow simulation in SimScale. Happy Simulating!

NOTE: As a cloud-based tool, all of your work is automatically saved. If you need to close the project and resume the tutorial at any point, you can re-open the project by going to your SimScale Dashboard and clicking on the project. To open the project in the Workbench, click the blue **OPEN** button in the upper right hand corner of the screen.

Important: SimScale® website is not properly working with Microsoft Explorer or Apple Safari. We strongly recommend using Mozilla Firefox or Google Chrome.

The first case will be the well known **lid-driven cavity** problem.

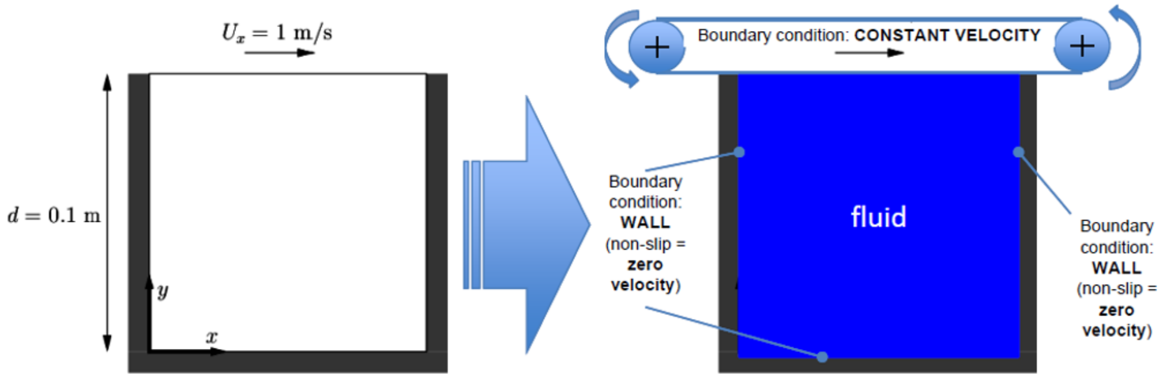

Figure 2.1: Scheme of the case.

A two dimensional case in the x-y plane of 0.1 ´ 0.1 m2.

2.2.1 The mesh of the case

Create the project. You can give any name, for instance "*Lid − driven − cavity*" (you could use a more original name). Then, in the window asking for dropping or uploading a file, drop the mesh file *'cavity-mesh-2D-coarse.zip'* and choose the 'OpenFOAM' format. Press 'Upload' and you will get a simple hexahedral mesh:

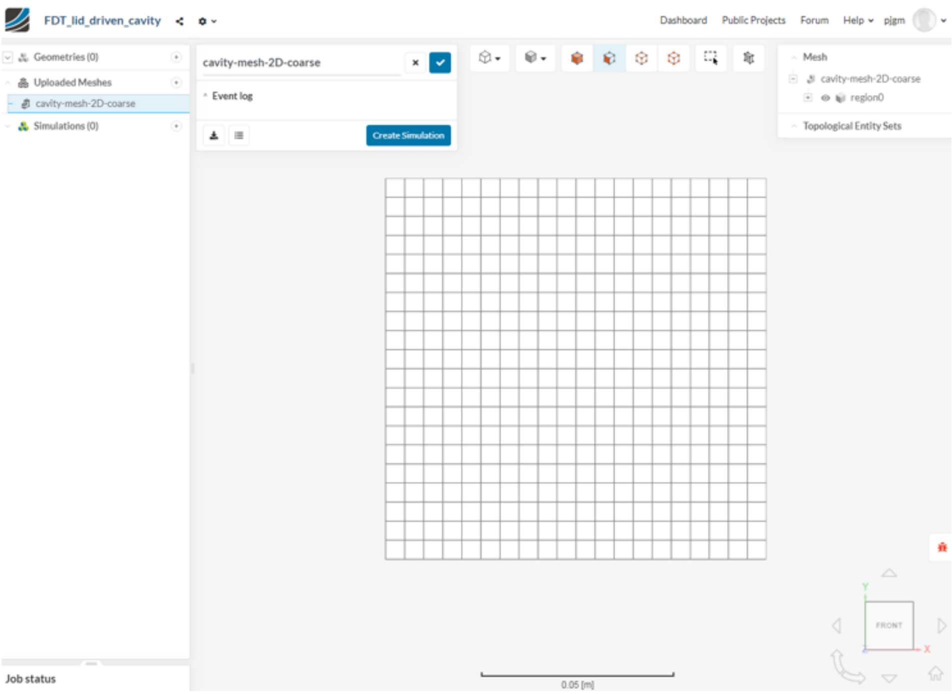

Figure 2.2: Basic hexahedral mesh

The hexahedral mesh is a uniform mesh of 20 by 20 cells (20 x 20 x 1 cells). This case is two dimensions (2D), and by default, 1 cell is assigned normal to the (3rd) dimension for which no solution is required. The cell size is $\delta_x = 0.1/(20 = 0.005)$ m in x-direction and $\delta_y = 0.1/(20 = 0.005)$ m in y-direction.

2.2.2 The Selection of the Solver

Click on 'Create Simulation', and select 'Incompressible'. Click 'Ok' to apply. Select also 'Laminar' for the turbulence model, 'Transient' for Time dependency and 'ICO' for the Algorithm. Save the configuration.
ICO is the abbreviation for the solver icoFoam. icoFoam solves the **incompressible laminar** Navier-Stokes equations using the PISO algorithm. The code is inherently **transient**, requiring an initial condition (such as zero velocity) and boundary conditions.

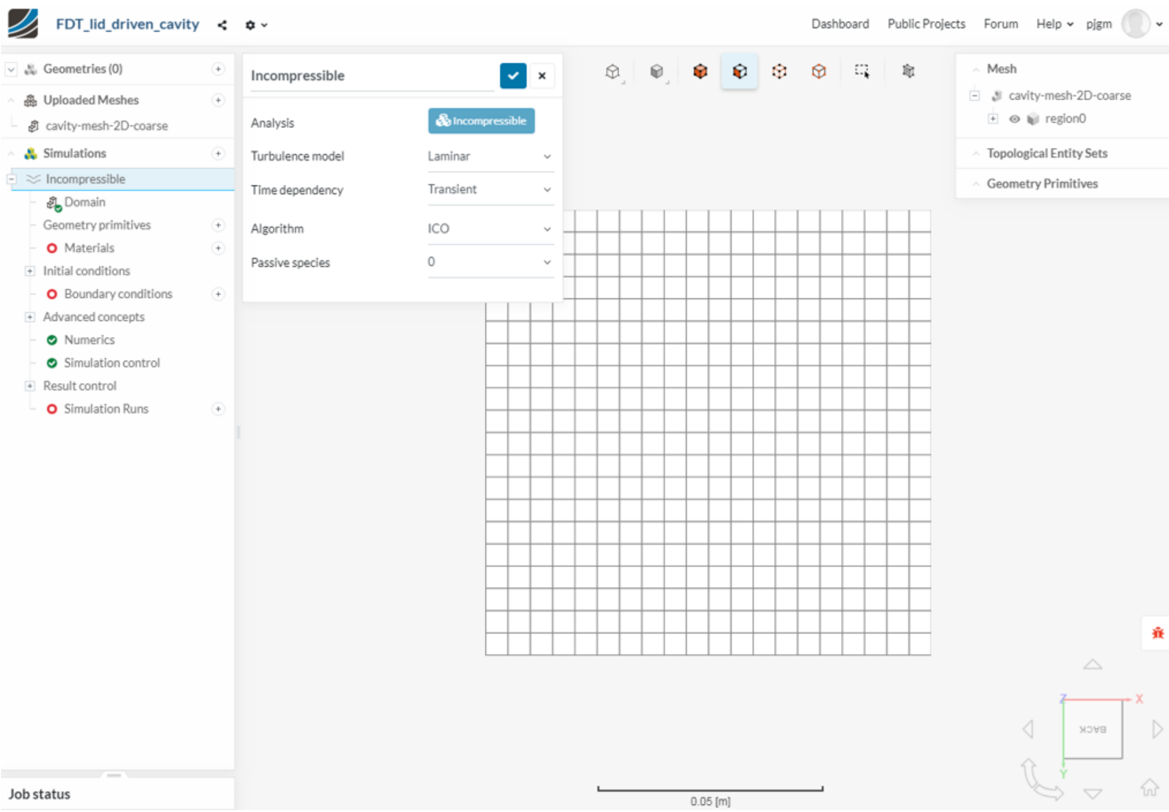

Figure 2.3: Solver selection.

Note that in the left tree branch there are some items marked with a red circle meaning that the simulation is still uncompleted.

2.2.3 2.3 The Topological Entities and Regions

Up to now, we have the mesh assigned to the simulation. But we have not defined any regions in this mesh. We will define 3 regions on the faces. The lateral and bottom faces will be called "walls". The top face will be called "MovingWall" and the front and back faces will be called "FrontAndBack".
In the mesh, select the two thin laterals and bottom faces in the viewer to create the region, click the **"+"** symbol next to **"Topological Entity Sets"** in the top-right side in the viewer, and create a new set, named "Walls".

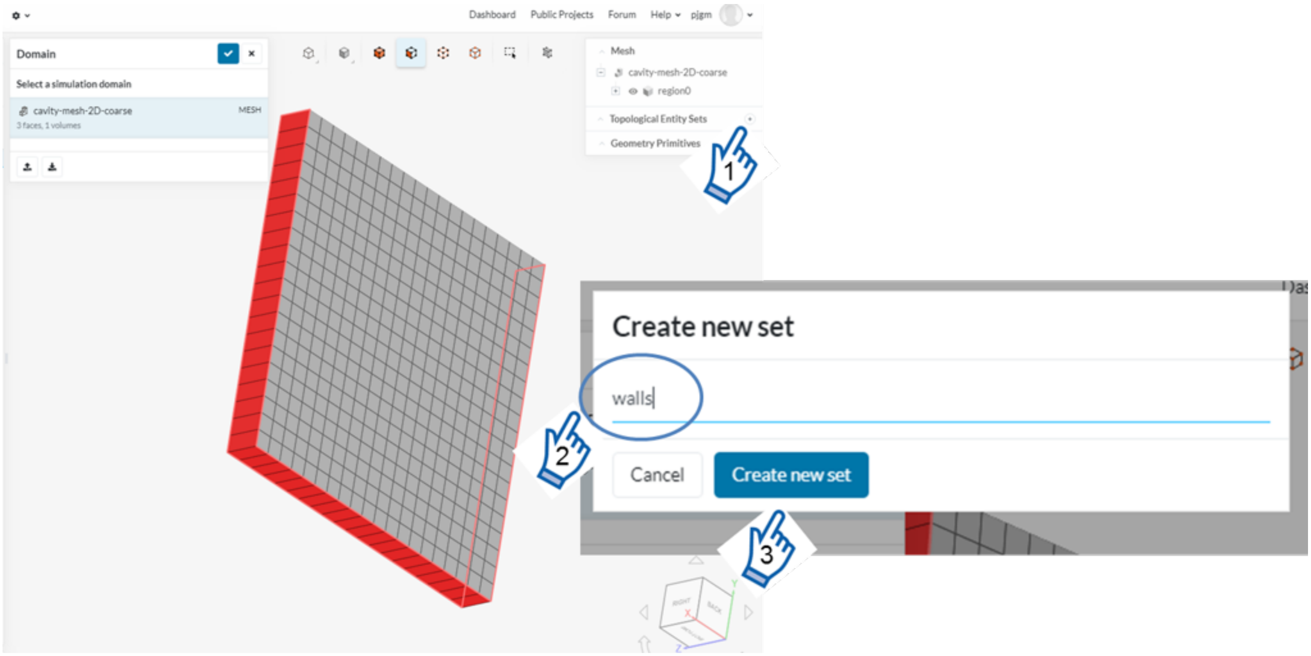

Figure 2.4: Walls selection.

Proceed in the same way for "MovingWall" and "FrontAndBack".

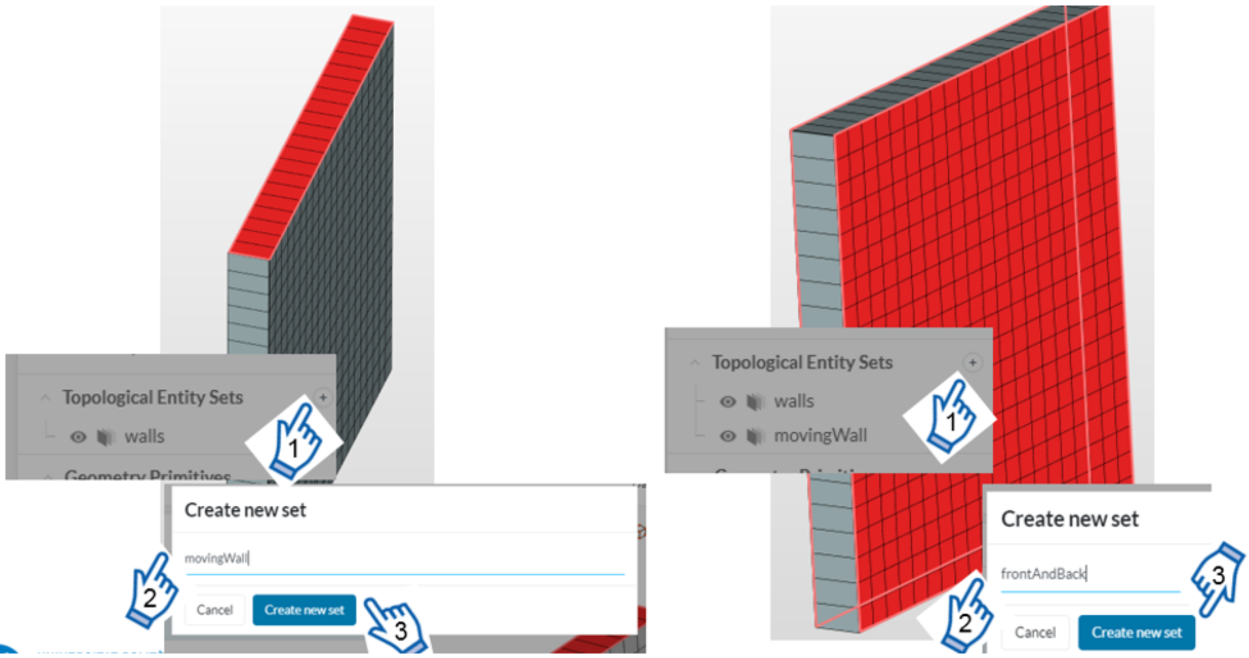

Figure 2.5: Topological Entity Sets.

2.2.4 The Fluid

Add "Water" as the material by pressing the **"+"** symbol in **Materials** item in the tree branch of the left side of the viewer.

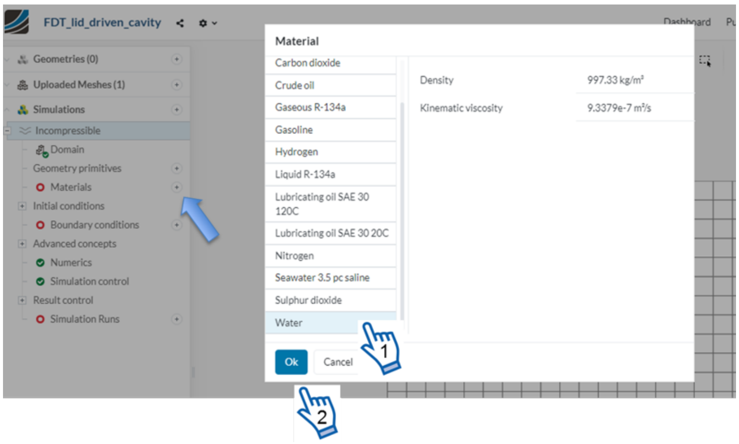

Figure 2.6: Material selection.

However, we will modify its viscosity, introducing a value of 0.01 m2/s. We intend to keep a low value of Reynolds number, and the flow will be laminar, as the icoFoam solver requires. Please, do not forget to assign it to "region0". Save it.

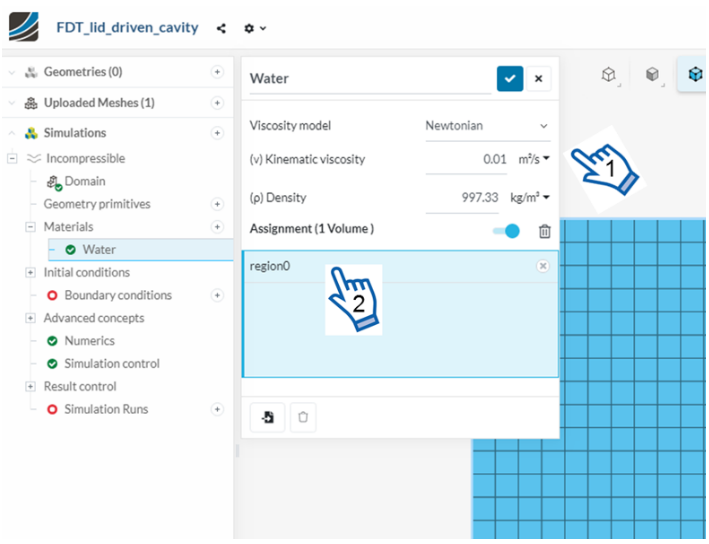

Figure 2.7: Material region selection.

Before stepping to configure the boundary conditions, please make sure that all the previous steps are done properly and if we have set to the faces the right names.

2.2.5 The Boundary Conditions

Keep the default pressure and velocity values as "Initial Conditions" in the tree branch of the left. One of the most important steps is the definition of the boundary conditions. Here we have three:

- Fixed walls
- Moving Walls

- "Front and Back" faces

Add a new boundary condition by clicking in the **"+"** of **Boundary conditions** in the tree branch, its type is "Wall".

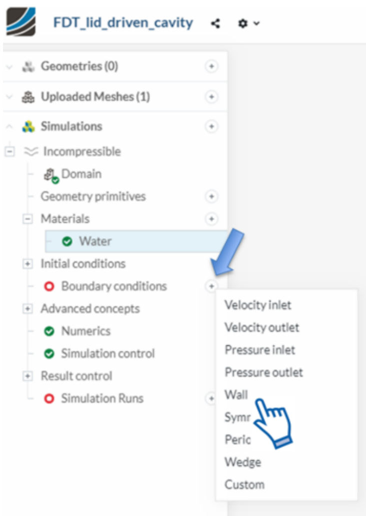

Figure 2.8: Boundary condition options.

Add a "Wall" boundary condition assigned to the "FixedWall" face. It can be renamed as "walls". Keep the "No-slip" condition for velocity and save it.

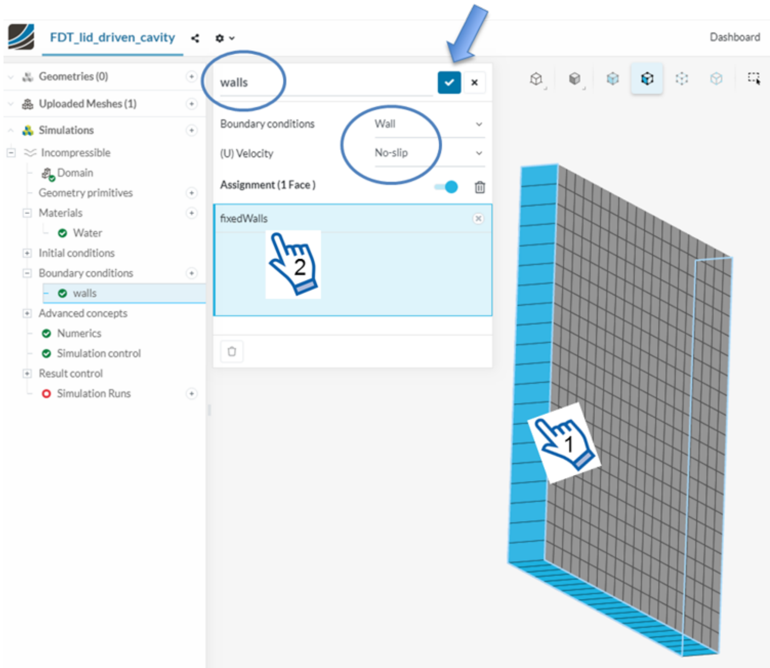

Figure 2.9: No-slip boundary conditon.

Create a new boundary condition by clicking in the **"+"** of **Boundary conditions** in the tree branch, its type is "Wall". We name it "MovingWall", and assign it to the

"MovingWall" face and the "Movingwall" in the (U) Velocity. The value is 1 m/s in the Ux. Save it.

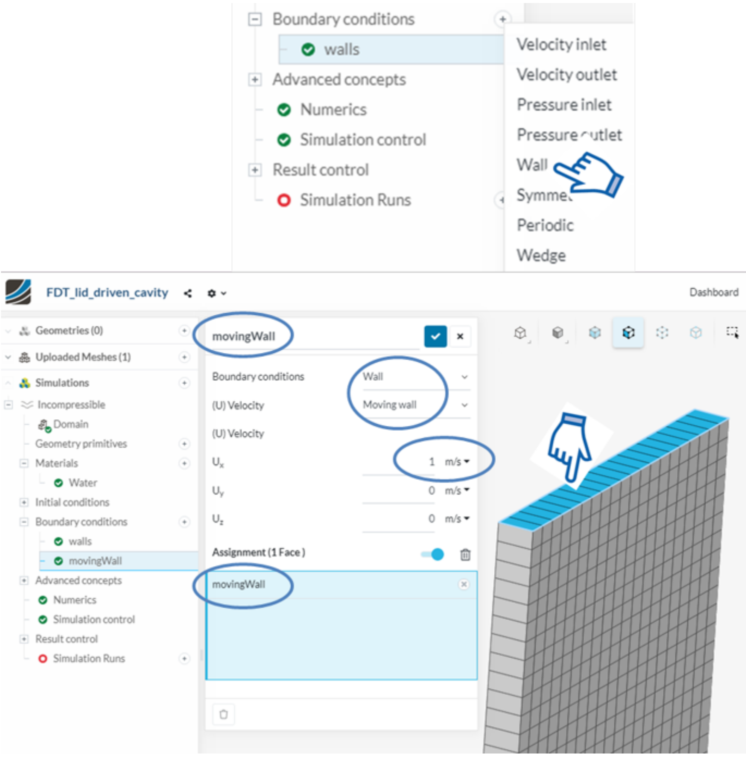

Figure 2.10: Moving wall BC.

And, finally, create a new boundary condition by clicking in the **"+"** of **Boundary conditions** in the tree branch, its type is "2D Empty". We name it "FrontAndBack ", and assign it to the to the "FrontAndBack" face Save it.

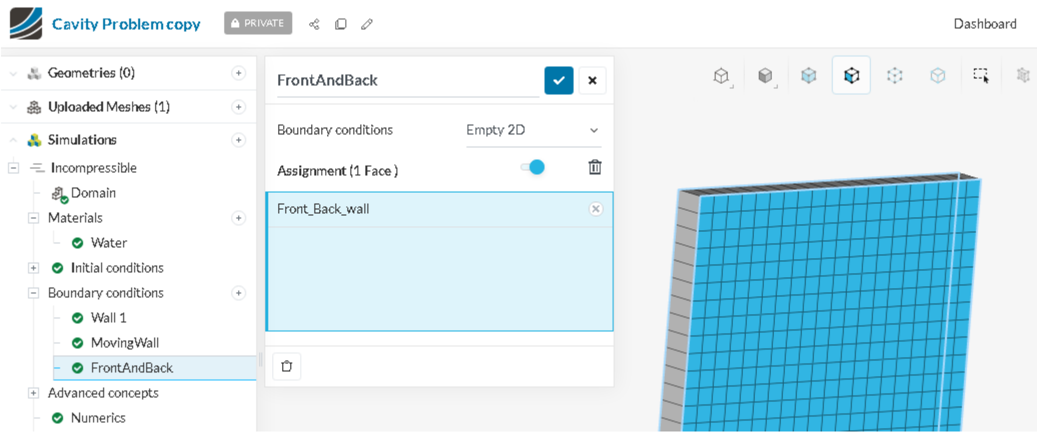

Figure 2.11: On the new version we can find the option "2D Empty".

From the OpenFOAM documentation this is the definition for the "empty" boundary type: "empty: for solutions in 2 (or 1) dimensions (2D/1D), the type used on each patch whose plane is normal to the 3rd (and 2nd) dimension for which no solution is required."

2.2.6 Setting-up the Case

We can skip the **Advanced Concepts** and leave the default values in **Numerics**. In **Simulation control**, set the "End time" to 0.05 s. Set the "Delta t" to 0.0005 s in order to keep the Courant number below 1. In "Write control", choose "Adjustable runtime" and 0.005 s in "Write interval". Save it.

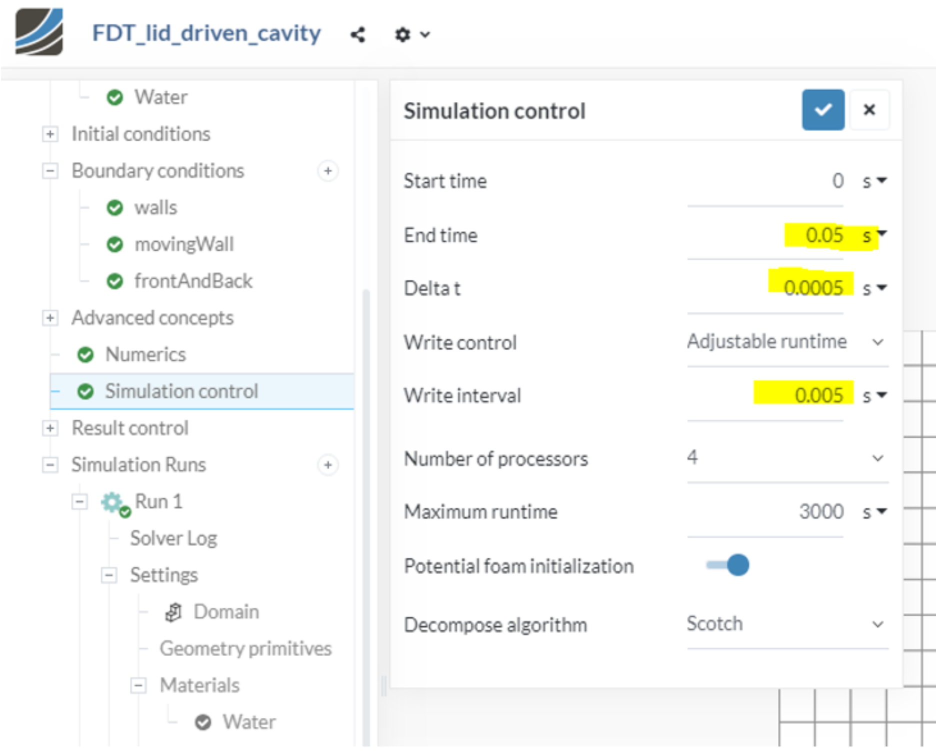

Figure 2.12: Simulation control confifuration.

It is worthwhile to comment about the time step "Delta t". The time step is related to the numerical stability. To achieve numerical stability, we need a low Courant number (Co). The Courant number is defined as:

$$C_o = \frac{|U|\delta_t}{\delta_x} \tag{1}$$

where $|U|$ is the velocity magnitude, δ_t is the time step and δ_x is the cell size.
In order to have a numerically stable simulation, we need that Co < 1 everywhere in the domain. Since the spatial resolution of the mesh is 20´20, $\delta_x = 0.1/20 = 0.005$ m, and since the maximum value of U will be 1 m/s near the upper wall, the value of the time step has to be:

$$\delta_t = \frac{C_o\delta_x}{|U|} = \frac{1 \cdot 0.005}{1} = 0.005s$$

However, in order to ensure even more the convergence of this case, we will set up thee time step ten times smaller.

$$\delta_t = 0.0005s$$

We can also skip the **Result control** this time.

2.2.7 Running the Simulation

We go directly to **Simulation Runs** and by clicking in the "+" **create** a new one. We can leave the "Run 1" default name.

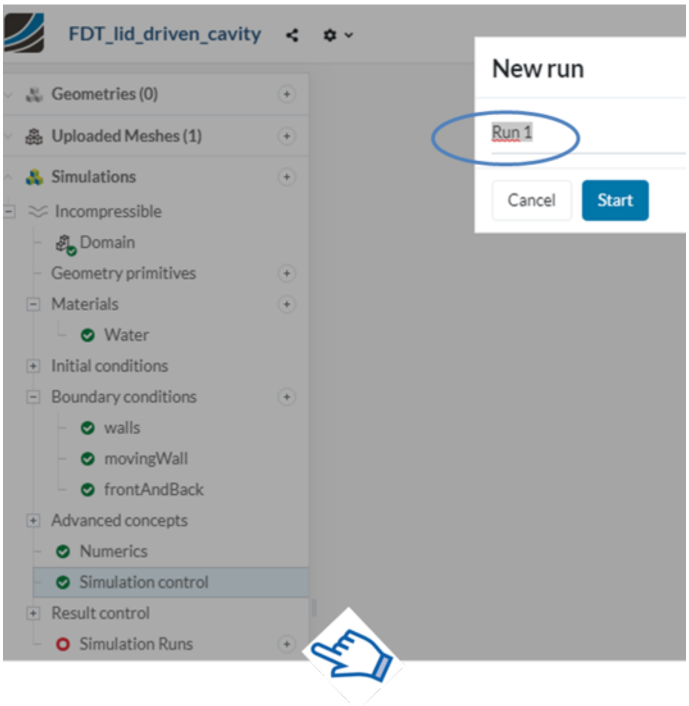

Figure 2.13: Create a new Run.

Start. The simulation will be queued and executed in the servers.

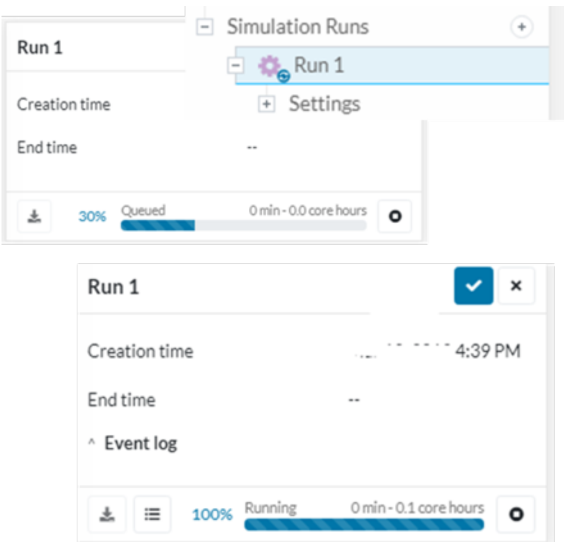

Figure 2.14: Run 1.

You will get the **Convergence plot**, in "Run 1 – Convergence plot" which indicates that the solution is correct with the default criteria.

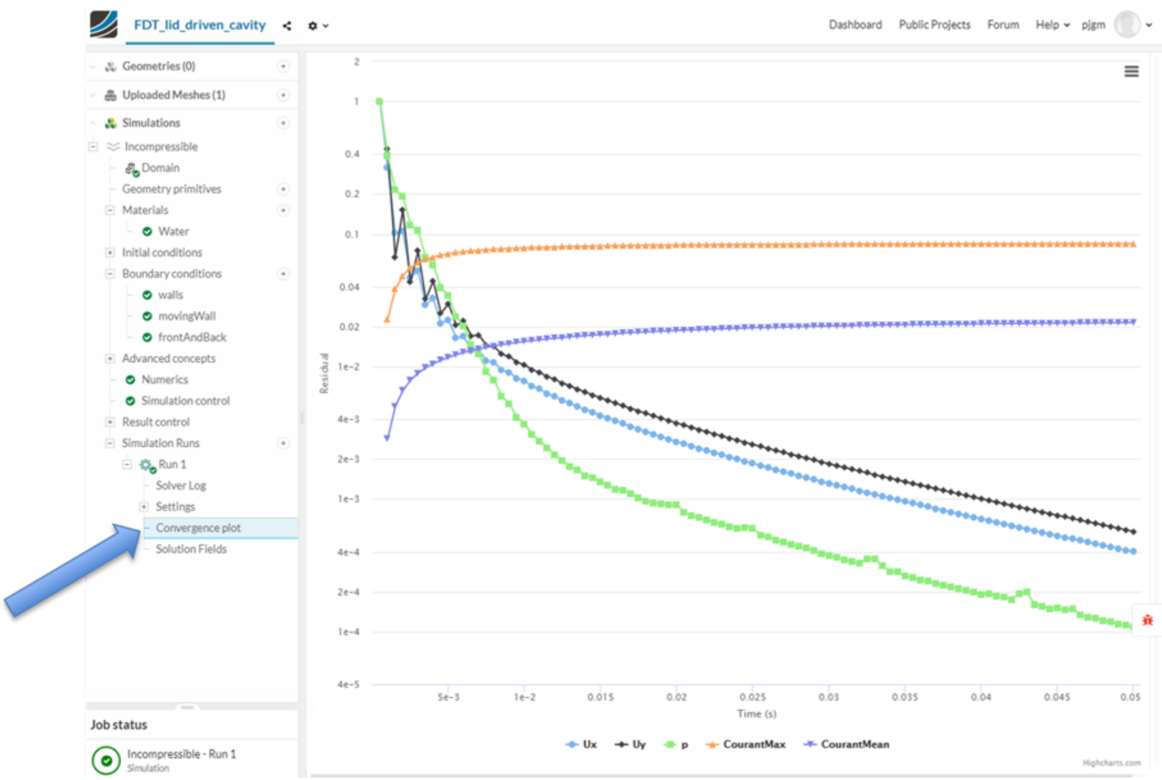

Figure 2.15: Simulation residuals.

Residuals should be below 10^{-3} and Courant number, maximum and minimum, should have reached a constant value. Residuals are a monitor of convergence of the linear solvers. A *low* value indicates that the linear algebraic system has been solved, but it does not mean that the problem has been solved.
Often other monitors are needed: CFL number (as in this case), forces, flow rates, etc.

2.2.8 Post-processing

Figure 2.16: Post-process results.

The post-process in initiated in each Run with the **Solution Fields** (1) option. Let us start by plotting the velocity contour, with **Results** (2) and the "Model" symbol next to **All Velocity** (3).
This shows the velocity magnitude as calculated, that is, in each mesh cell. Extrapolated field can be visualized with **All Velocity [node]**.

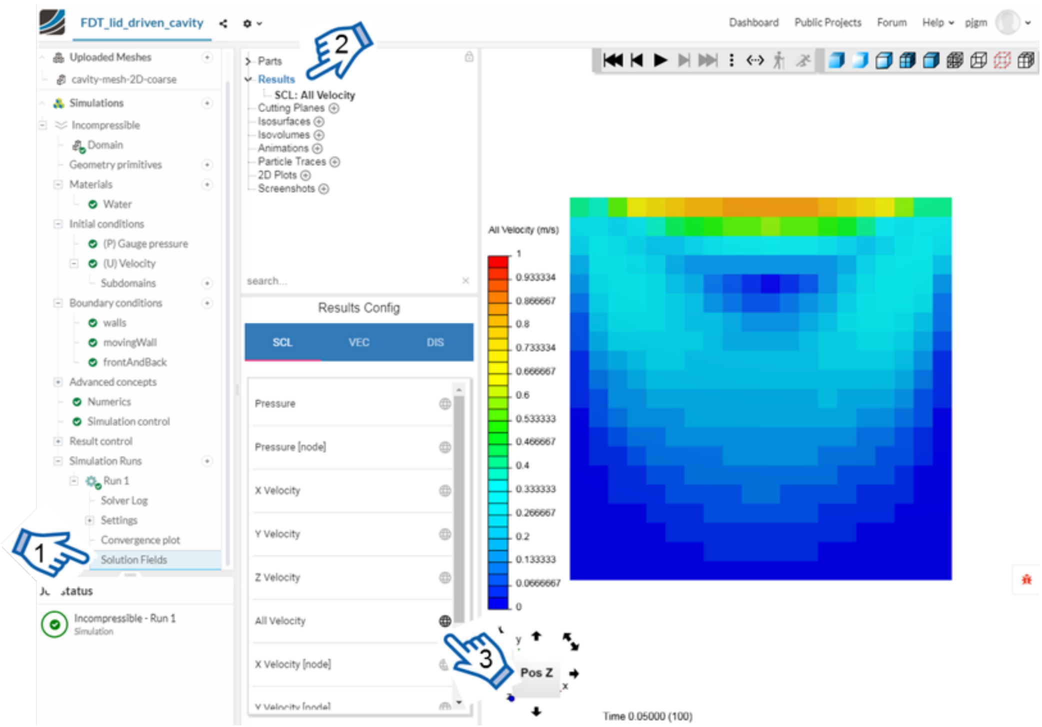

Figure 2.17: Velocity distribution.

Velocity vectors can be displayed with the tab **VEC** in "Results Cong". The setup is available clicking on the VEC: Velocity in the upper window and the color will be of the field in the "SCL" window.

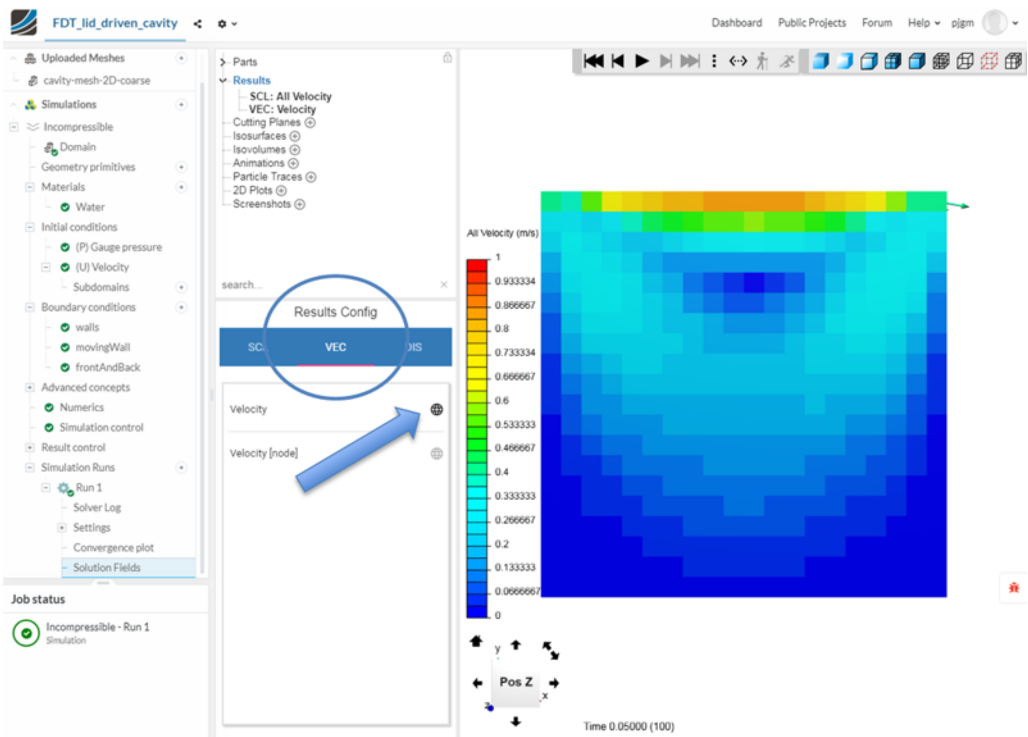

Figure 2.18: SimScale's post-processor.

In order to see the vectors, may be need to put the visualization mode in **Outline**. Otherwise, the velocity contour can cover the velocity vectors. Also change the "Color mode" to 'By fringes'.

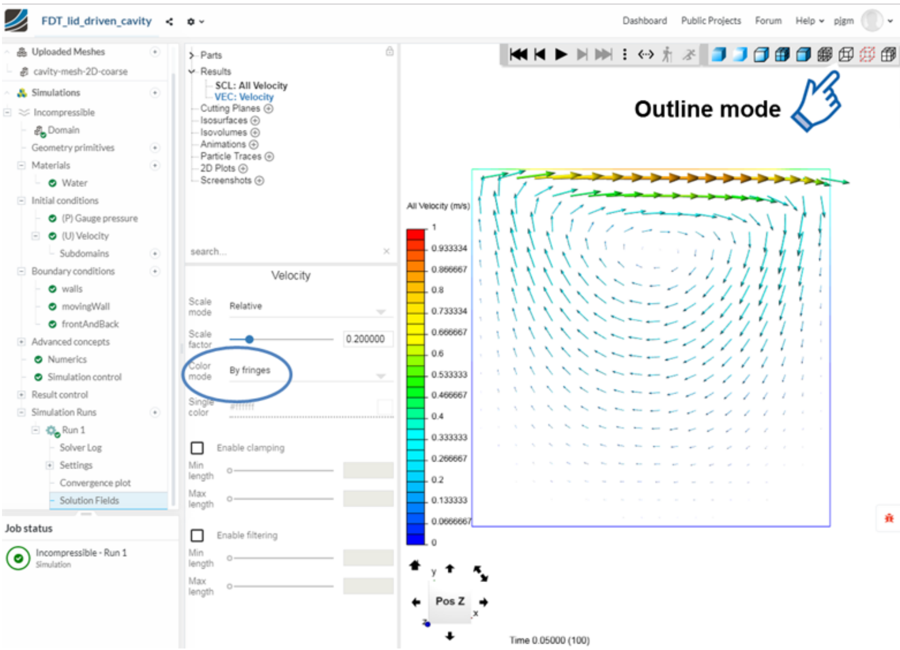

Figure 2.19: Vector velocity distribution.

Also streamlines can be depicted with **Particle Traces**. In this next figure, the configuration for the SETTINGS is shown

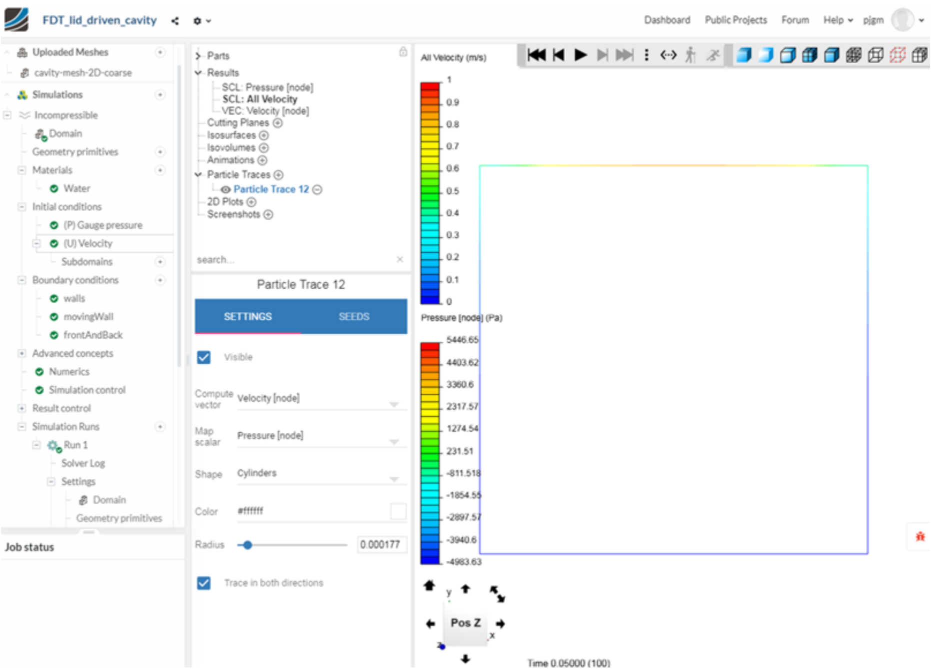

Figure 2.20: Configure the streamlines.

In this figure, the configuration for the SEED is shown. The seeds are located with the **PICK** button.

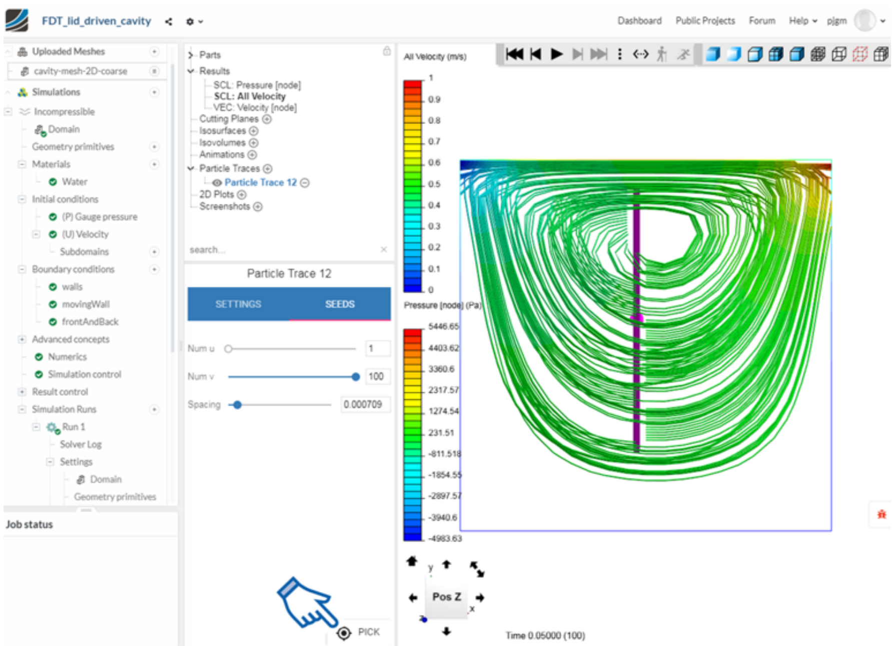

Figure 2.21: Streamlines.

Although this post-processing environment is useful to check the simulation results, it is quite limited. It is possible to download the simulation in order to locally process the results with the **ParaView** program (`https://www.ParaView.org`), which is much more complete. ParaView is an open-source, multi-platform data analysis and visualization application.

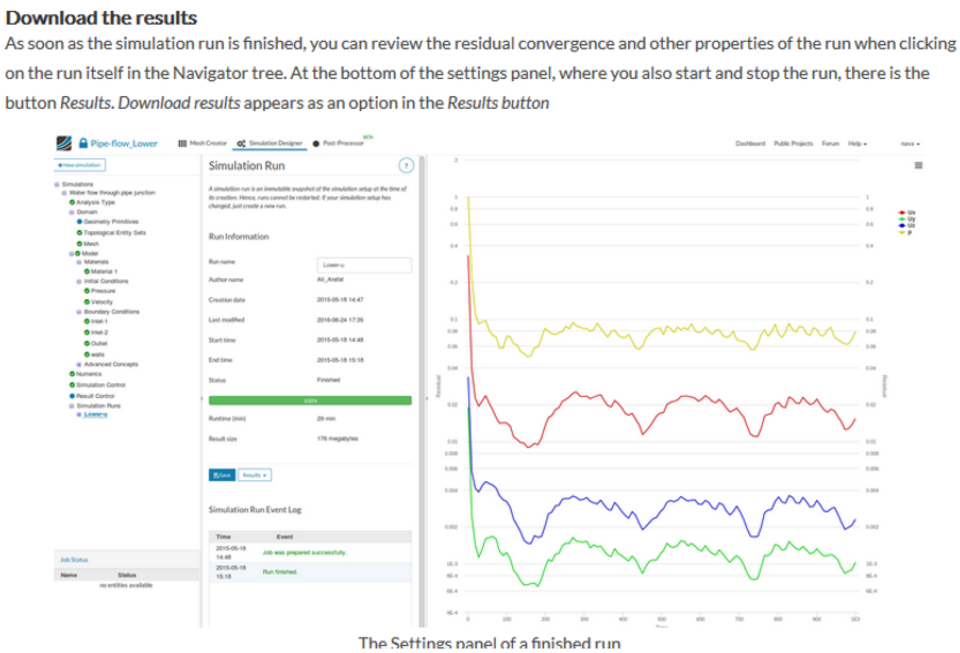

Figure 2.22: Download the results.

3 Internal Flow

Once the user is introduced to the software, it is the moment to start with the first case study. This case is the classical pipe flow. The objective is to study the meshing process to observe the boundary layer and the viscous effects. Before introducing the problem, some theory sum up used in this case that has been used and it will be key to check our results in the post-processing.

3.1 Theoretical Framework

As it is known and large explained over the years, it can be found two kind of flows: laminar and turbulent. Both cases will be considered in this study. When the flow inside the pipe is **turbulent** it is found two kind of shear strengths: one viscous and the other turbulent. The total shear strength is the sum of both [3].

$$\tau_{total} = \tau_{viscous} + \tau_{turbulent} \tag{2}$$

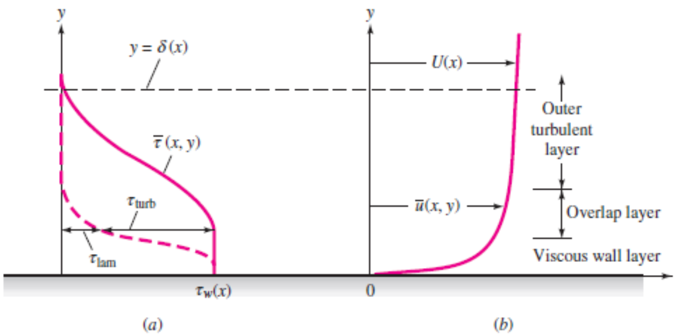

Figure 3.1: Typical velocity and shear τ distributions in turbulent flow near a wall: (a) shear; (b) velocity. [4]

As it is explained in [3], this two shear strengths can be calculated as:

$$\tau_{total} = \rho(\nu_t + \nu)\frac{d\bar{u}}{dy} \tag{3}$$

Being $\nu and \nu_t$ the laminar and turbulent viscosity, respectively. The turbulent component it is also called the kinematic eddy viscosity.
The boundary layer in this flow can be considered with three regions[4], which are characterized by the perpendicular distance to the wall:

- Wall layer: Viscous shear stress dominates
- Overlap layer: Both types of shear are important.
- Outer layer: Turbulent shear stress dominates.

Some authors consider another model, with four regions, an additional layer between the wall and overlap layer called the buffer layer. The velocity profile can be obtained by two dimensional variables 4 and 5 as it is explained in [3]:

$$y^+ = \frac{y u_\tau}{\nu} \tag{4}$$

Which is the dimensionless distance to the wall and

$$u^+ = \frac{u}{u_\tau} \tag{5}$$

is the dimensionless velocity.

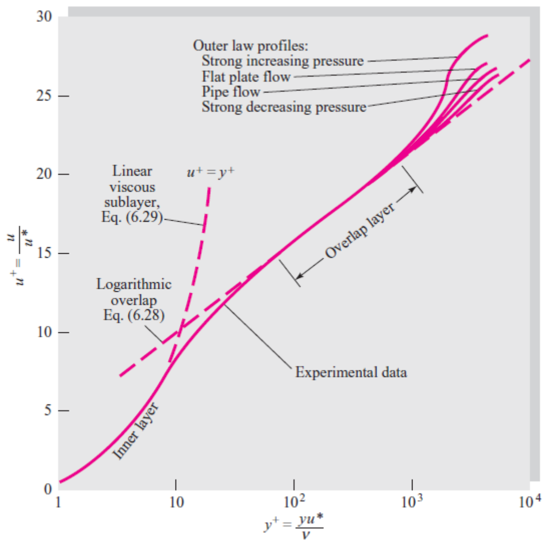

Figure 3.2: u+ over y+. [4]

After the simulations, the representation of u^+ over y^+ should mach with the plot showed in the figure 3.2. One of the characteristics of the turbulent flow is the instabilities produced near the wall. These appear due the viscous effects which induce to vorticity and difficult the task of modeling of the turbulent effects. It is wanted capture the consequences of the wall layer to have the most accurate result and help the student to understand the boundary layer.

In order to achieve this, it is needed to have the first node closer than $y^+ = 5$ to the wall, which is the point were this layer, approximately, ends. To obtain the value of $y(y^+ = 5)$, it has to be calculated first the friction velocity (u_τ) as:

$$u_\tau = u_\infty \left(\frac{C_f}{2}\right)^2 \tag{6}$$

Where C_f is the friction coefficient which is calculated with a common expression:

$$C_f = \frac{f}{4}$$

And f is the Darcy friction factor. This coefficient depends on the shear strength:

$$f = \frac{8\tau_w}{\rho V_{avg}^2}$$

But due to the complications to calculate τ_w, it will be used experimental data from [4]. Particularly for a laminar flow, can be approximated as [3]:

$$f = \frac{64}{Re} \tag{7}$$

After the friction coefficient is obtained, the velocity at the center of the pipe (u_∞) is needed. Nevertheless, lack of knowledge of the velocity profile makes impossible to find a average value of u_∞, thus, it must be approximated as:

$$\bar{u} = u_\infty$$

For a turbulent flow, the error assumed is little due to the smooth distribution of the velocity profile. Therefore, the maximum value is very close to the average value.

Finally, the last analytical calculation will be the pressure loss over the pipe, which is calculated thanks to the Darcy Weisbach expression for fully developed turbulent and laminar flows (Eq.8).

$$\Delta P = \rho f \frac{L}{D}\frac{u^2}{2} \tag{8}$$

This equation does not take into consideration the entrance pressure drop, characteristic of the internal flow in pipes. This is showed in the figure 3.3.

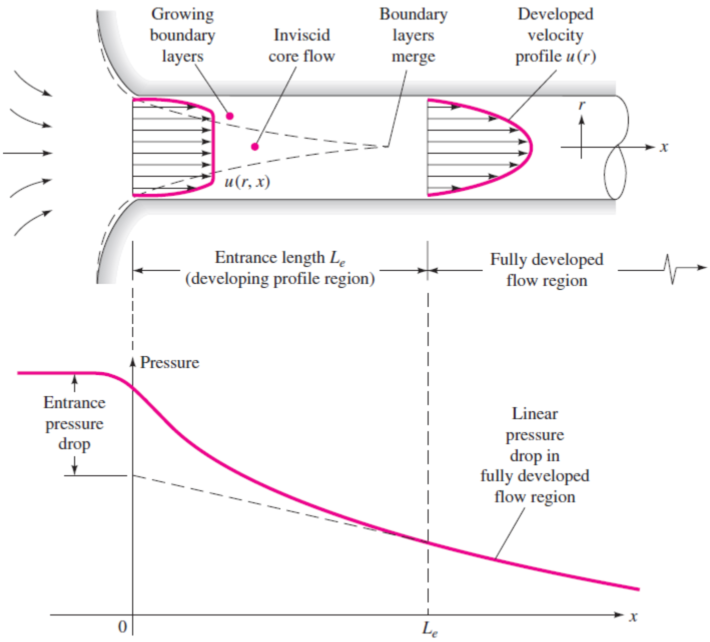

Figure 3.3: Developing velocity profiles and pressure changes in the entrance of a duct flow. [4]

Also it will be calculated the length to obtain a fully developed flow and with the expressions found at [3] for a laminar and turbulent flow:

$$L_{h,laminar} = 0.05DRe \tag{9}$$

$$L_{h,turbulent} = 1.359DRe^{1/4} \tag{10}$$

The entry length is much shorter in turbulent flow and its dependence on the Reynolds number is weaker.

3.2 Meshing Methods

As is it largely known, one of the most important tasks in CFD is to compute an efficient mesh able to capture all the effects and that not suppose a large amount of calculation time. This part is the most difficult.
During this project, a lot meshes has been tested in order to obtain the bests results but most of them were inefficient. One common meshing method is, for 2D cases (as the pipe flow), is to mesh just a wedge of the pipe taking advance of the axial symmetry of the the solid. One of the observations made is that SimScale algorithms do not compute 1D or 2D meshes, i.e., only computes 3D meshes and it has been argued in many forums ([10], [11]). Nevertheless, this tool can simulate 2D case but the attempt of this project is to introduce the student to the software, this is why it has been decided is better that the 2D mesh is created with the software and not imported from another place. Given the situation, to reduce computation time, it has been used the symmetry boundary condition, which "reduces the computational domain in size and enables the modeling of a sub-domain of the complete setup" [12]. For this reason, it has been used half pipe as a solid. It has been tried to reduce even more the problem by simulating not half of the section but a third and applying symmetry in the two external faces but with unsuccessful results.
With all this in mind, two meshing configuration will be used for the simulations.

1. Homogeneous mesh: used for the laminar case with no need to have node too close to the wall. Also this mesh will be used in the turbulent case it will be used "wall-functions to resolve the near-wall region" [7].

2. Refined mesh: most of the nodes will be placed near the wall to capture the wall layer. The first node will be placed at a distance smaller than $y^+ < 5$. To calculate this distance is used the equation 4.

To reduce computation time, it has been tried to reduce the number of cells in the x direction. Despite of this, it has been observed that with too slender cells the results are not satisfactory.

3.3 Pipe Flow Case Study

In this subsection, it will be detailed step by step the procedure to simulate with SimScale and use ParaView as a post-processor the pipe laminar and turbulent flow. The data of the problem are presented in the table 1. The main point is that we will create the mesh (or several) with SimScale, it will be provided only the 3D solid and it will be analysed the results for different meshes and flows.

Pipe's longitude	L	2	$[m]$
Pipe's diameter	D	0.025	$[m]$
Water's density	ρ	1000	$[kg/m^3]$
Water's dynamic viscosity	μ	0.001	$[kg/m.s]$
Inlet velocity*	u	4;0.01	$[m/s]$
Pressure outlet	P_s	101325	$[Pa]$

Table 1: Data of the problem. *The two values of the velocity are for the turbulent and laminar regime, respectively.

3.3.1 Analytical results

Before the simulation is performed, it is necessary to calculate some parameters to check in the post process the accuracy of the simulation. In the tables 2 and 3 for a laminar and turbulent flow, respectively, the analytical results are presented.

Reynolds	Re	250	-
Friction factor	f	0.256	-
Pressure drop	ΔP	1.024	$[Pa]$
Fully developed flow	$L_{h,laminar}$	0.3125	$[m]$

Table 2: Analytical results for a laminar flow.

Reynolds	Re	10^5	-
Friction factor	f	0.018	-
Pressure drop	ΔP	11520	$[Pa]$
Fully developed flow	$L_{h,turbulent}$	0.604	$[m]$
Friction velocity	u_τ	0.1897	$[m/s]$
First node wall distance	$y(y^+ = 5)$	$2.635 \cdot 10^{-5}$	$[m]$

Table 3: Analytical results for a turbulent flow.

Where ΔP has been calculated with the equation 9. The friction factor for the laminar case has been calculated with the equation 7 and for the turbulent case it has been used experimental data from [4].

3.3.2 Laminar Flow

Firstly, as it has been done in the *"The Lid-Driven Cavity Case"*, we have to create a new project: name it "Pipe Flow" and click create. Once the workbench is open we need to upload the geometry. To download the geometry click this **link**.

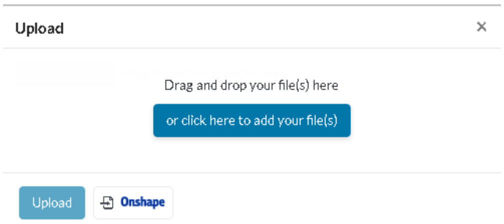

Figure 3.4: Add the geometry.

Before clicking *Upload* we have to say that our geometry is a Solid-Work geometry, right under *"Half_pipe_2m"*. Then click *Upload* and, once is uploaded, *"Create Simulation"*.

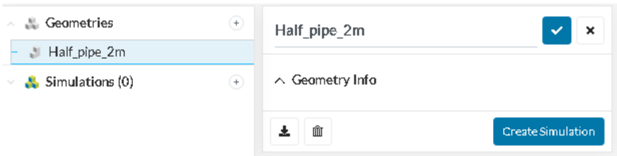

Figure 3.5: Create a simulation.

In this case, the fluid is flow is incompressible, so we will click *"incompressible"* and *"Create Simulation"*.

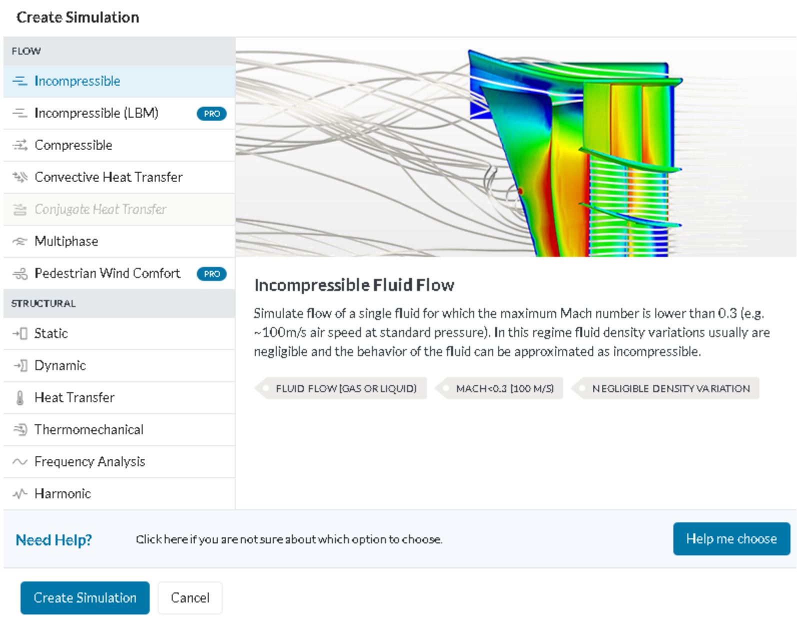

Figure 3.6: Choose the simulation type.

The first case to compute is the laminar flow. Thus, we will rename the simulation to *"Incompressible_Laminar"* and we need to change the turbulence model to *"Laminar"*. This is showed in the figure 3.6

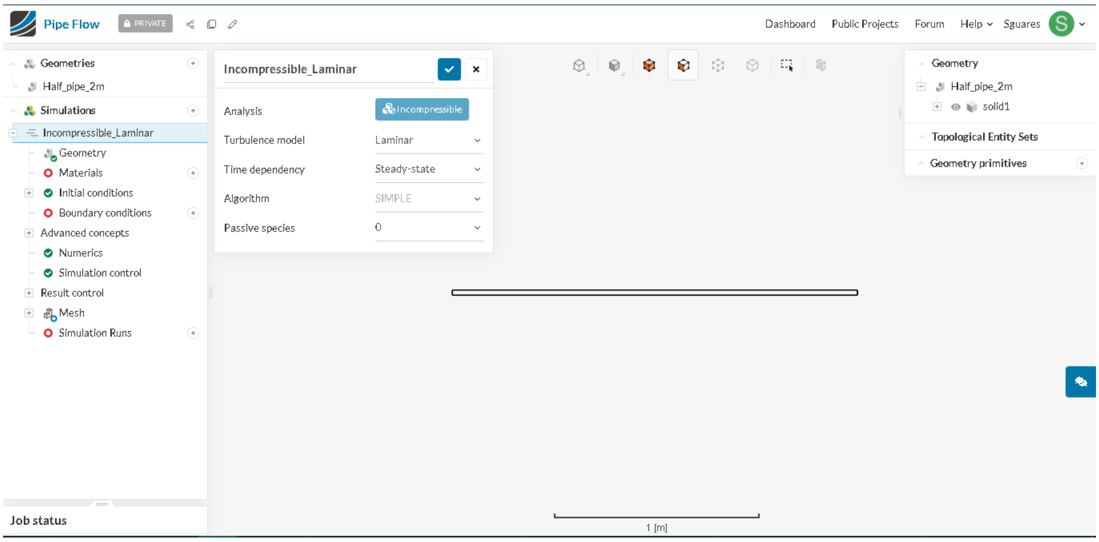

Figure 3.7: Your screen should be like this.

We have created a new simulations, now let's configure it. First of all, we will tell SimScale that the fluid is water. So, click *"Materials > Water > Apply"*. We will change the kinematic viscosity of the water and the density to integer numbers to make or hand calculations easier. The values are showed in this figure:

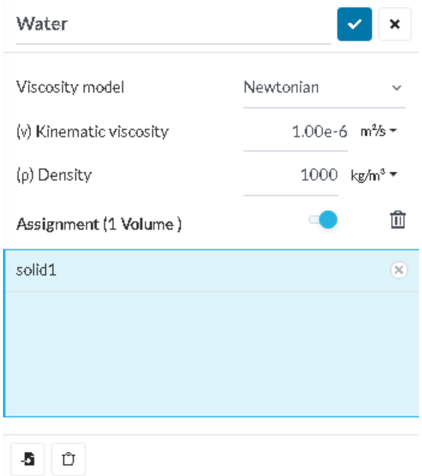

Figure 3.8: Water properties.

Once the values are changed click the tick to save. Now, we are going to specify the boundary conditions. Click *"Boundary Conditions > Velocity Inlet"*. On the inlet, we have the value of velocity flow ($U_x = 0.01 m/s$) and we have to select the left face of the cylinder (whit the Top view selected).

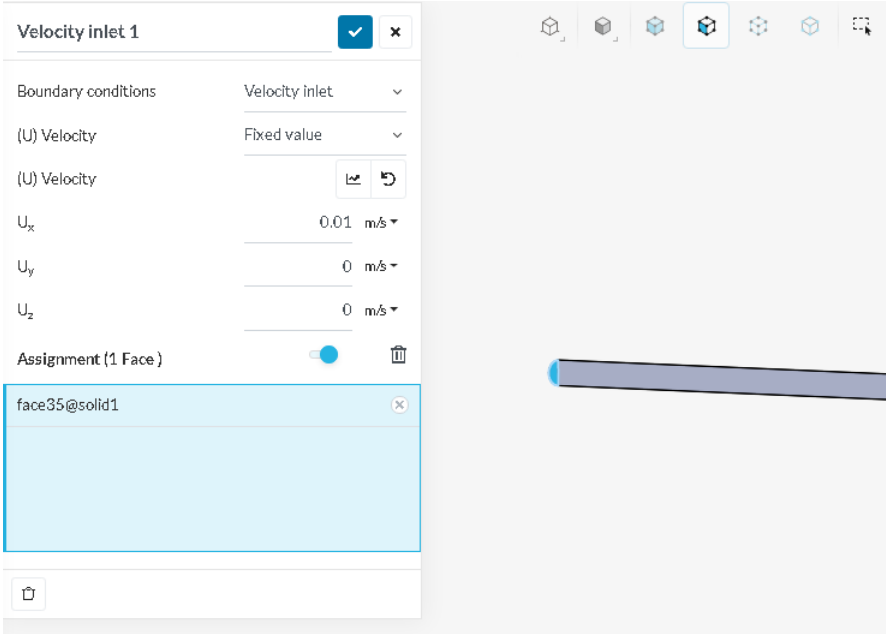

Figure 3.9: Velocity inlet.

Click the tick to save. Next boundary condition is the pressure outlet. Click "+" next to *"Boundary Conditions"* and select *"Pressure Outlet"*. Make sure to select the right face.

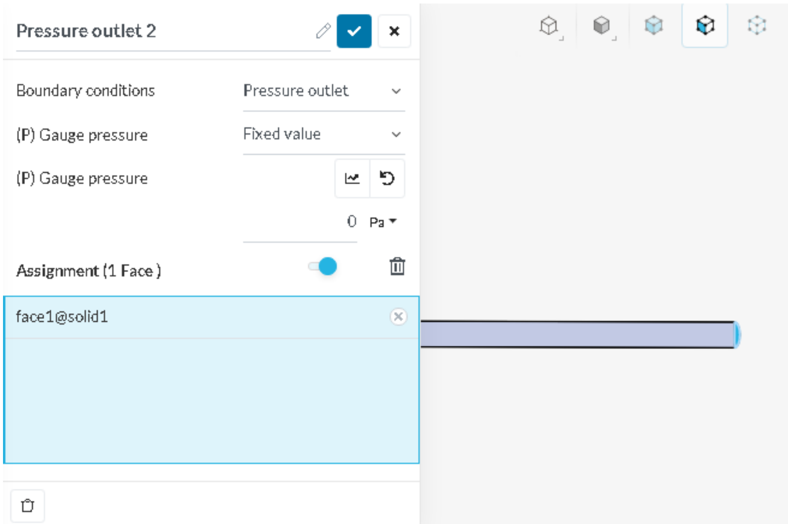

Figure 3.10: Pressure outlet.

We will keep the default Gaussage pressure as 0 Pa. Click the tick to save. For the no-slip condition at the wall of the cilinder we have to click "+" next to *"Boundary Conditions"* and select *"Wall"*.

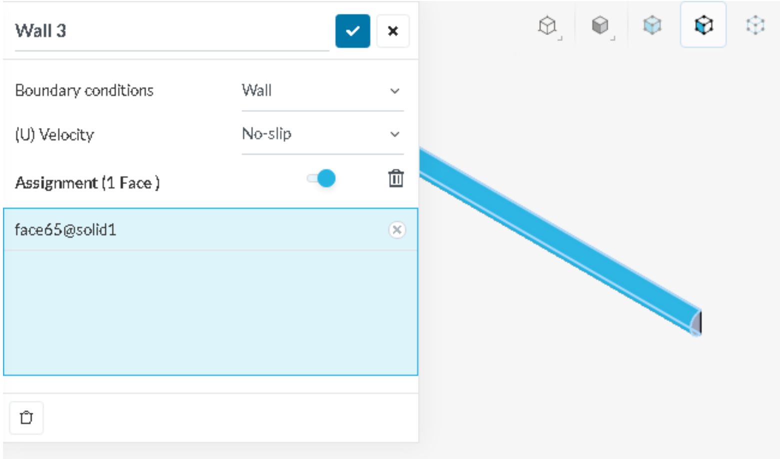

Figure 3.11: Make sure that the *"No-slip"* condition is selected

Select the face showed in the previous figure and click the tick to save. The last boundary condition is, as we have half pipe, we need to apply a symmetry to have a hole pipe. This configurations is very useful to save more than a half of the computation time.

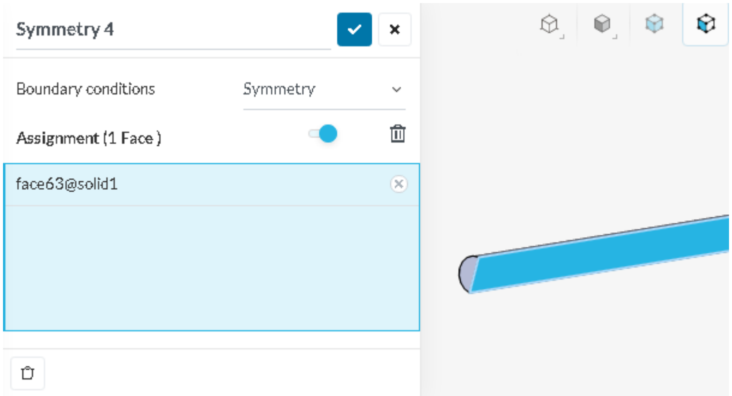

Figure 3.12: Symmetry boundary condition.

Select the blue face showed in the figure 3.12.Click the tick to save.
Now it is time to configure the mesh. In this case, as it was explained in the subsection 3.2 we want a homogeneous mesh. There is no need to refine it near the wall because it is not a requisite to visualize the effects near the wall.
In order to do so, click mesh and rename it to *Homogeneous_Mesh*. First, we choose the algorithm. Click *"Hex-dominant parametric (only CFD)"*. We could use *"Hex-dominant (only CFD)"*, which is very useful in most o the cases, where the algorithm computes automatically the mesh. Therefore, in our case we need to specify the cells in X,Y,Z direction in order to obtain slimmer cells. With slimmer cell we reduce the number of them in the X direction and we save some computation time. The results are not affected.

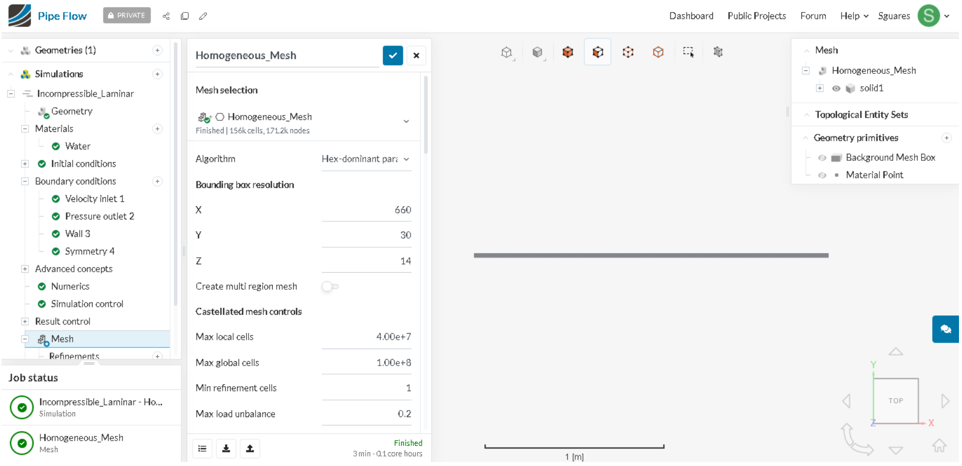

Figure 3.13: Hex-dominant parametric configuration

Change the parameters of the *"Bounding box resolution"* as showed in the figure 3.13. After changing the parameters click *"Generate"*. It should take around 4 minutes to compute the mesh.

Figure 3.14: Homogeneous mesh result.

While the mesh is computing or once is finished, click *"Simulation Control"*. In a steady-state simulations, there is no need to have the Courant number < 1 to achieve numerical stability. This number is calculated with the equation 1, with $\delta_y = 0.025/30 \approx 0.001$ m. Thus, the $\delta_t = 0.1$ s. However, in order to ensure even more the convergence of this case, we will set up thee time step ten times smaller ($\delta_t = 0.01$ s). Actually, we are using an implicit solver for the steady-state simulation and this ones are usually less sensitive to numerical instability, so larger values of the Courant number may be tolerated. But, just in case, we will keep it equal to 1 [1].
To calculate the number of iterations it can be used the following expression:

$$Iterations = \frac{SimulationTime}{\Delta t}$$

[1]For more information about the CFL condition and solvers look at the Annex.

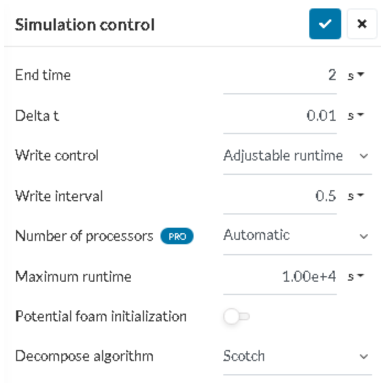

Figure 3.15: Simulation control configuration.

In addition to this, change the other parameters as is showed in the figure 3.15. Then, click the tick to save.

All the pre-process has been completed, therefore, we can run the simulation. Click in *"Simulation Runs"*, change the name to *"Homogeneous_Mesh_Laminar"* and click *"Start"*.

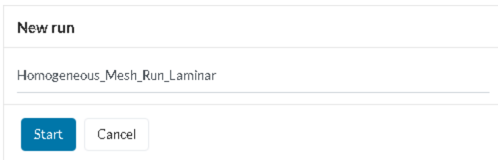

Figure 3.16: Start the new run.

While is running, we can check if the results are converging and click *"Convergence Plot"*. Once the simulation is finished, we will download the results to use as post-processor ParaView, which is much more complete than the the one that SimScale offers. Click on the symbol of download.

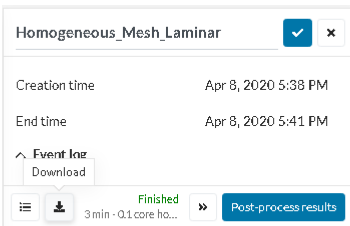

Figure 3.17: The simmulation should take around 3 minutes.

In order to check that our results are more or less the ones that we are looking for, we can click *"Post-process Results"* and *"Results"* to see the pressure and velocity distribution. We can observe that they are as expected.

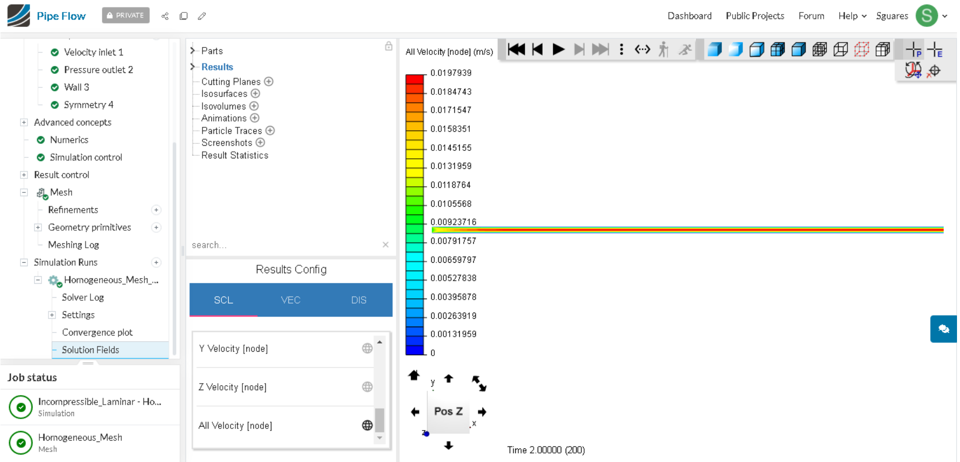

Figure 3.18: Pressure distribution overview.

Once is checked that we may have good results, we can proceed to work with ParaView.

3.3.3 Post-processing with ParaView

ParaView is an "open-source, multi-platform data analysis and visualization application" [13] and we can download the application in his web side or clicking this **link**. In this link there is multiple options to download depending on the software we are using. Once the results are downloaded and the program installed, we need to decompress the archive in a folder and open the file called *"case"* and ParaView will be launch.

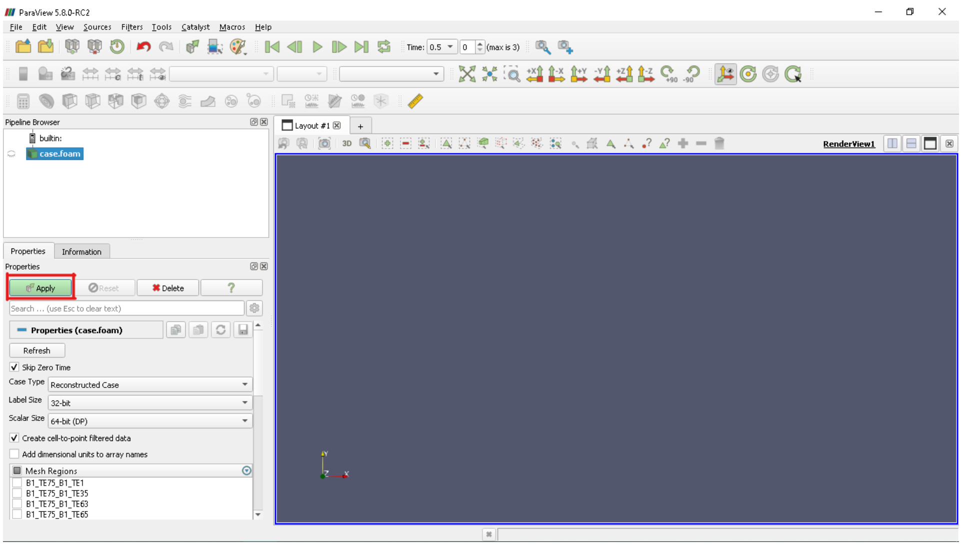

Figure 3.19: Click apply to open the results

One time the results are displayed, you will see the pipe in the Layout #1. To visualize the right results we need to change the time step and, also, we will work with another color scale to observe more details. First, change the time to *"2"* if is not set right and then click the symbol showed in the figure 3.20 to open the Color Map Editor.

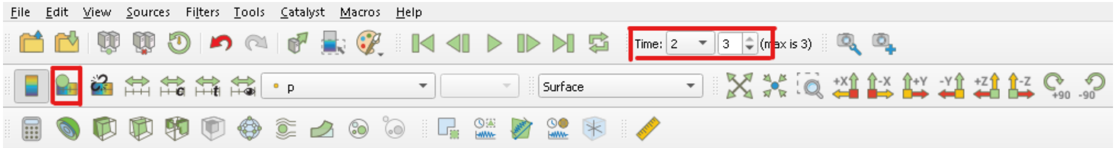

Figure 3.20: Change the time!

In the Color Map Editor we can change the colors of the results, the scale, etc... Click the symbol with the heart.

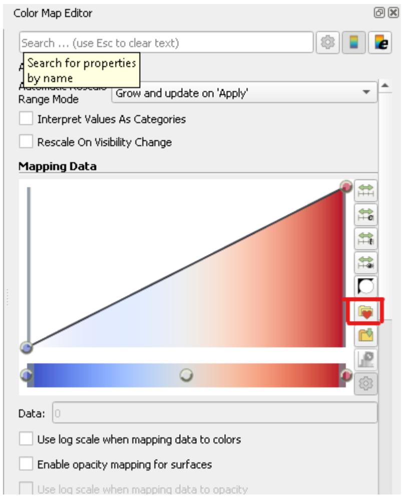

Figure 3.21: Select a new color scale.

After clicking this symbol a new window will be open and we need to type in the search bar *"Blue to red Rainbow"*, select the Preset, click apply and close the window. Now we can close the Color Map Editor. Our screen should look like this:

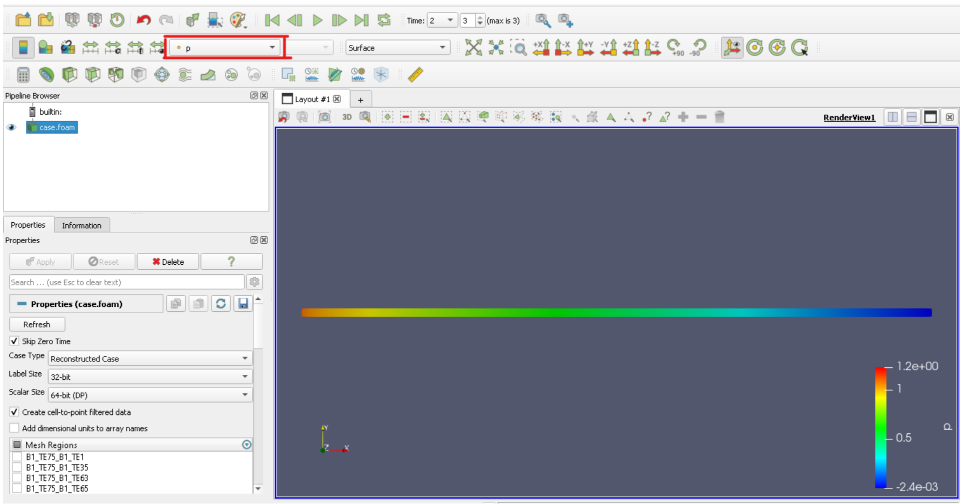

Figure 3.22: Pressure distribution with the new color scale.

Now we have displayed the pressure distribution over the pipe but we can change it to display the velocity clicking the box showed in the figure 3.22.

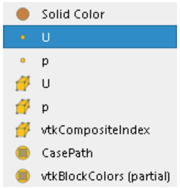

Figure 3.23: Click ”· *U*”

Now the velocity is displayed but with the default color. To change it we need to follow the same procedure as before.

Figure 3.24: Velocity distribution with ”Blue to red Rainbow” color scale.

For the laminar case, to check the results, we need to have the pressure distribution along the x axis and the velocity profile when we have a fully developed flow. In order to do so, we need to plot this to components over a line. First, we are going to obtain the pressure.

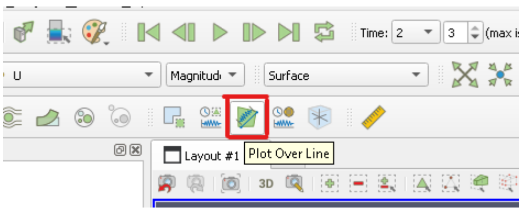

Figure 3.25: Click the ”Plot over the line” symbol.

Rename the action from ”Plotoverline” to *”Pressure”* and define the line as the figure 3.26.

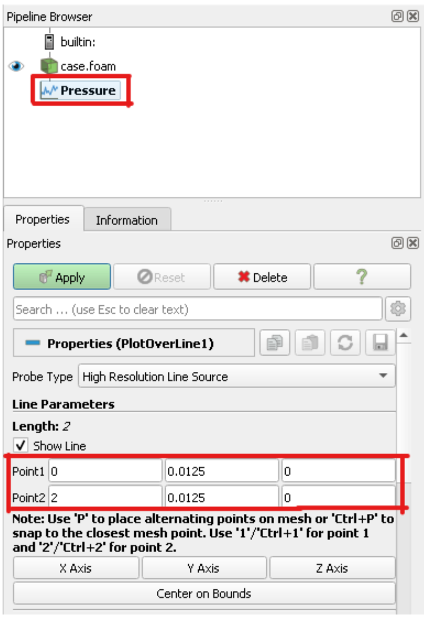

Figure 3.26: Create a line along the pipe.

The line showed in the layout should be in the center of the pipe. Then, click apply. Uncheck "U_Magnitude" to show only the pressure distribution.

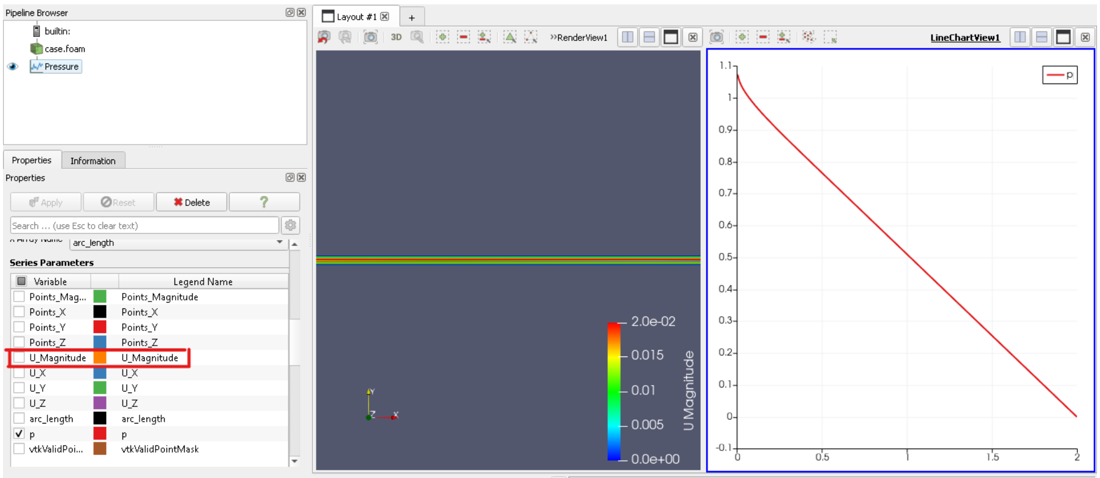

Figure 3.27: Select only *"p"*.

To set the axis titles and custom range scroll down an set the following parameters:

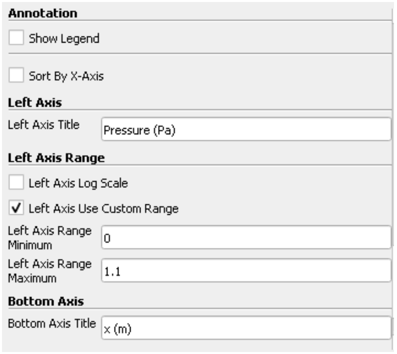

Figure 3.28: Configure the axis.

The final plot should look like this:

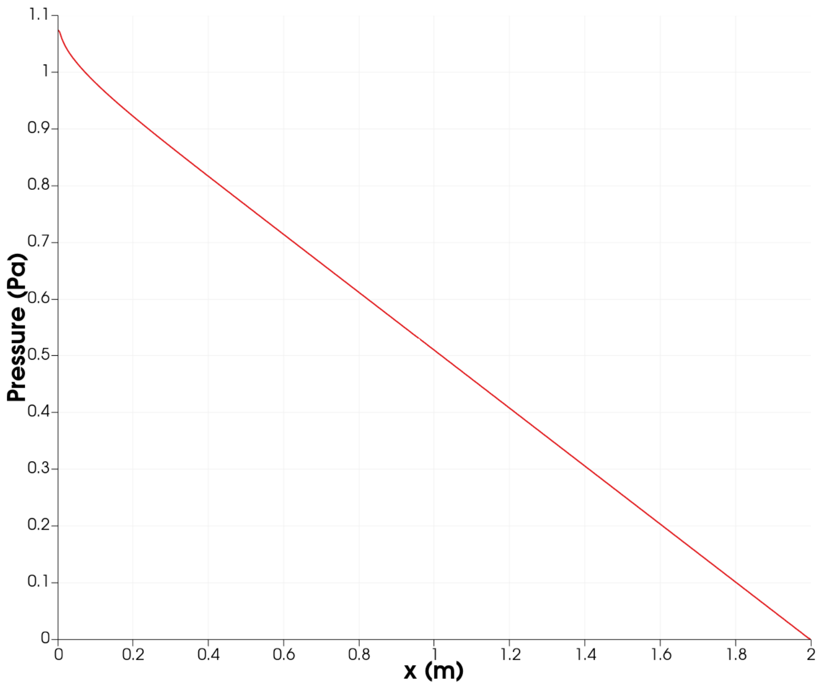

Figure 3.29: Pressure distribution of a laminar flow over the central line.

The results will be discussed in the subsection 3.3.6 along with the turbulent results. To plot the velocity profile we will open a new layout and also we need the dimensionless Y coordinates and the velocity average. For that, we can use the *"Calculator"*. Click the "+" and then select *"case.foam"* again as showed in the figure 3.30.

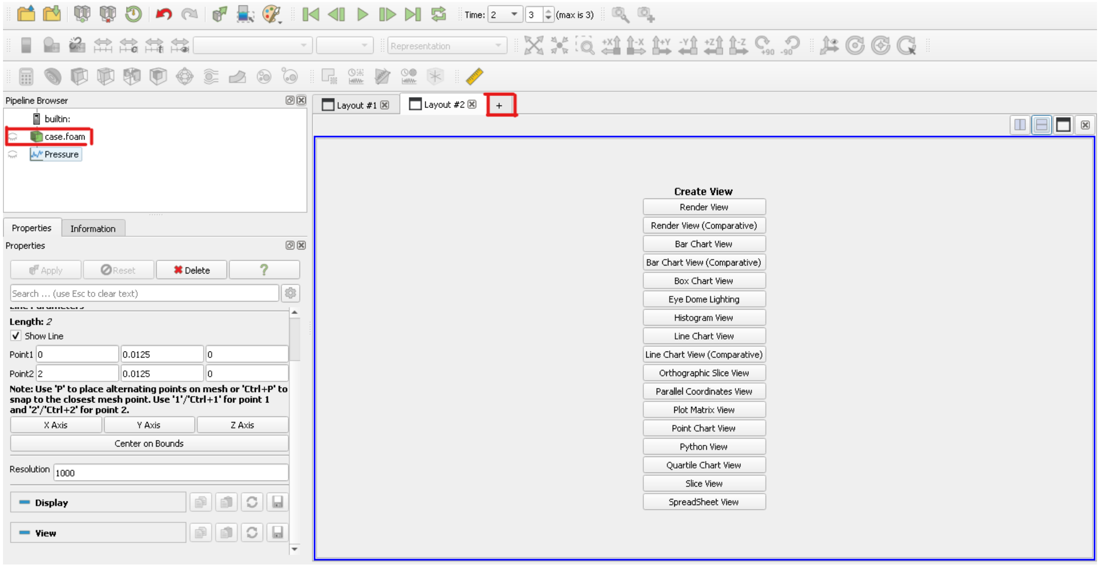

Figure 3.30: Select *"case.foam"*.

After, click the calculator symbol to open it. Calculate the average velocity as in the figure 3.31 and click apply (we can write the U vector or select it from the *"Vectors"* list).

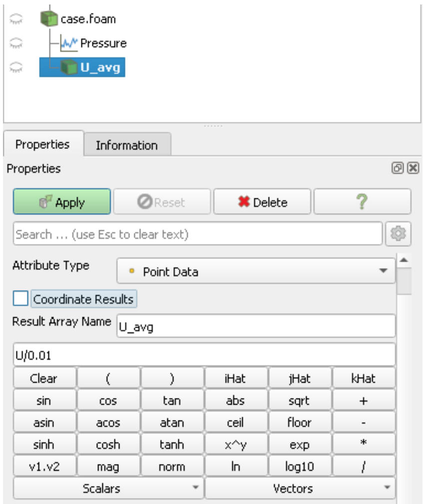

Figure 3.31: U_avg calculation.

Then, click again the calculator to calculate the dimensionless Y coordinates and configure it as in the figure 3.32 (we can write the Y scalar as *"CoordsY"* or select it from the *"Scalars"* list) and click apply.

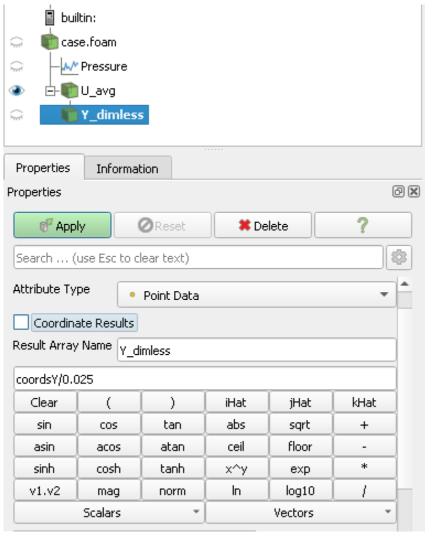

Figure 3.32: Y_dimless calculation.

Now, as we have done for the pressure distribution, create a plot over a line and configure the line as in the following figure:

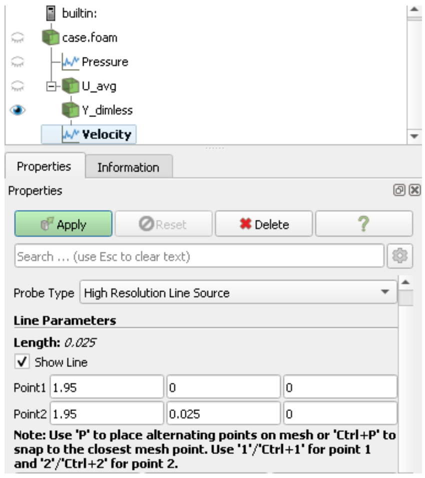

Figure 3.33: Create a line over the section at the exit of the pipe.

We create the line at x = 1.95 to avoid perturbations that could appear at the exit of the pipe. Click apply. After the plot appears, we need to change the *"X array name"* to *"U_avg_Magnitude* and from the Series Parameters select only the *"arc_length"*. Also, click the "x" showed in the figure 3.34 to close the pipe view.

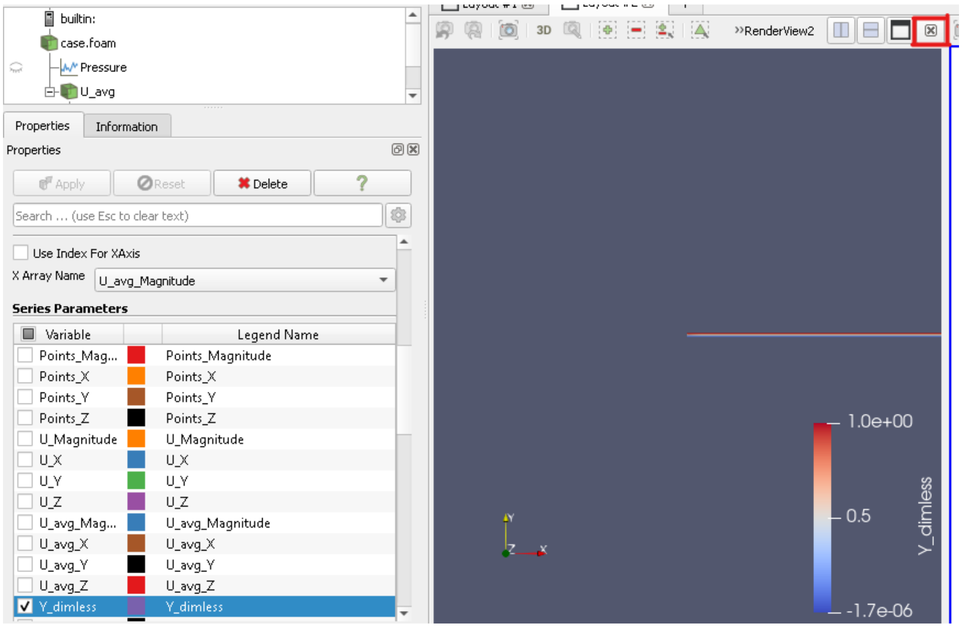

Figure 3.34: Select *"arc_lenght"* as the variable.

We can set the axis titles an change the range.

Figure 3.35: Configure the axis.

The final plot should look like this one:

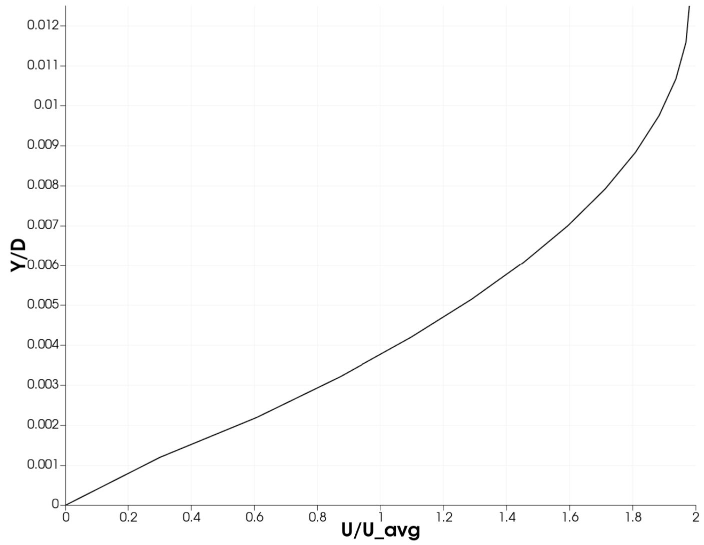

Figure 3.36: Velocity profile at the exit of the pipe for the laminar flow.

To finish our post-processing with ParaView we can save the state.

3.3.4 Turbulent Flow

The second part of the analysis is the study of the turbulent flow. It will be simulated with an homogeneous mesh and an other one with some refinement at the boundary layer to observe the effects of the wall layer.

The first simulations is going to be with the refined mesh. This mesh is very useful because it is only used the cells needed and the effects near the wall are better captured. With an homogeneous mesh there a lot of unnecessary cells at the middle of the mesh, which supposes a lot of computation time.
First, we will create a new simulation in the Pipe Flow workbench. Click "-" next to *"Incompressible_Laminar"* to hide the previous configuration and click the "+" next to *"Simulations"*.

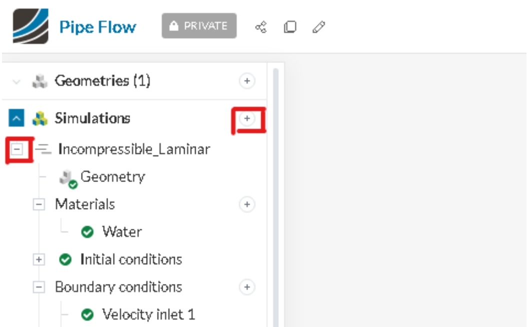

Figure 3.37: Create a new incompressible simulation.

Create a new incompressible simulation an rename it *"Incompressible_Turbulent"*. Leave the *"k-omega SST"* turbulence model and click the tick to save.

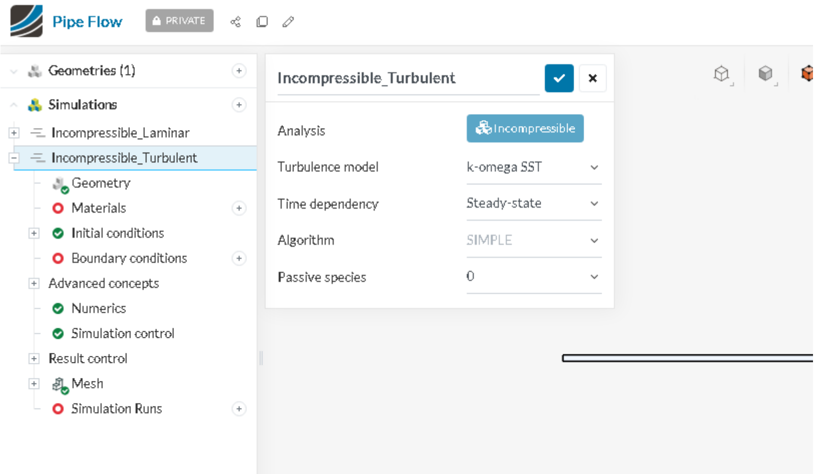

Figure 3.38: New simulation configuration.

For the materials, choose water as before and change the parameters as the figure 3.9. Now we need to configure the boundary conditions. Click "+" next to *"Boundary Conditions"* and *"Velocity inlet"*. This time the velocity value is $4m/s$:

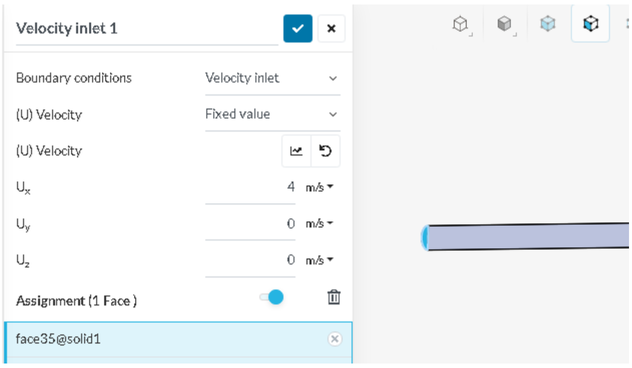

Figure 3.39: Velocity inlet.

For the *"Wall"* condition the configuration will be also different. In this case we are going to choose *"Full resolution"* modeling. This supposes an increment in the computation time but *"This approach explicitly models the boundary layer all the way down to the laminar sub-layer"* [7] and it is necessary if we need to obtain the shear stress near the wall and we are going to have a mesh with a node near the wall with a distance $y^+ < 5$ as explained before in the *"Theoretical Framework"* (subsection 3.1).

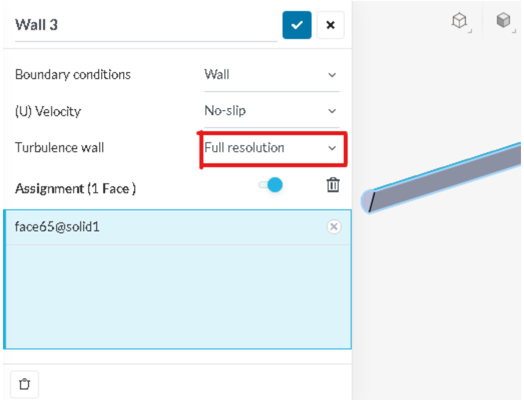

Figure 3.40: Wall boundary condition.

On the other hand, *"Wall functions"* are empirical equations used to satisfy the physics of the flow in the near wall region and the first cell center needs to be placed in the log-law (overlap layer) region to ensure the accuracy of the result. Nevertheless, the results are more approximated than with a good mesh with *"Full resolution"*[2].
Then, configure the *"Pressure outlet"* and *"Symmetry"* as in the laminar case (figures 3.10 and 3.12, respectively). After, we can start creating the new mesh. Go to *"Mesh"* and click *"Create new mesh"*.

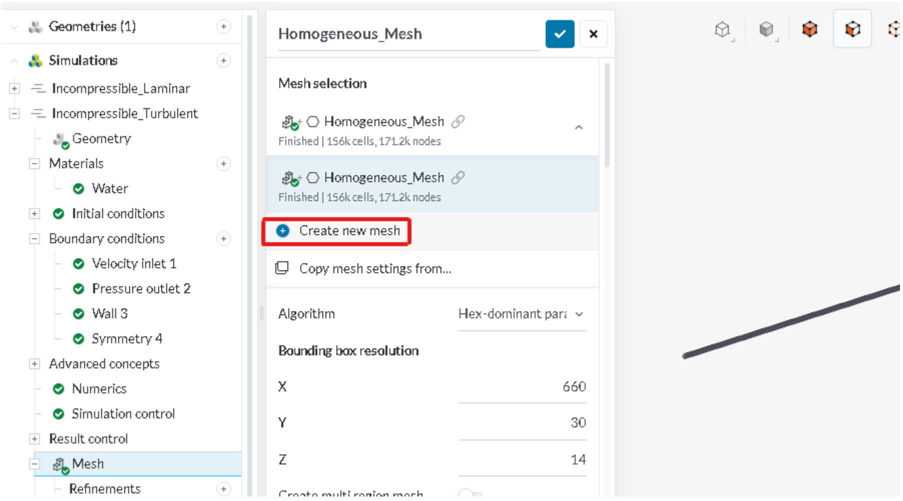

Figure 3.41: Create a new mesh.

Choose the *" Hex-dominant parametric"* algorithm and set the Bounding box resolution parameters as in the figure 3.42.

[2]Go to Annex for more information about Wall Function

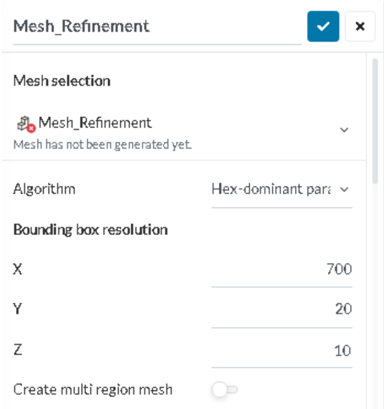

Figure 3.42: Bounding box resolution parameters.

Rename it *"Mesh_refinement"*. Scroll down an under *"Snap Controls"* change the parameters showed in the figure 3.43. This will help us to obtain a better mesh, more fitted to the solid.

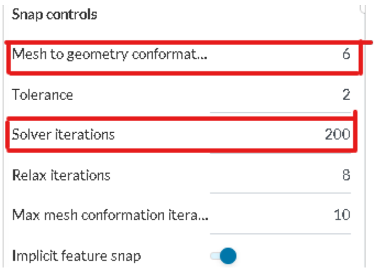

Figure 3.43: Change the *"Snap Controls"*.

Then go to *"Refinements>Inflate Boundary Layer"*. This setting "adds a volume mesh with cells aligned to the surface" [14].In other words, cells will be added near the surface selected. Configure the refinements as the following figure:

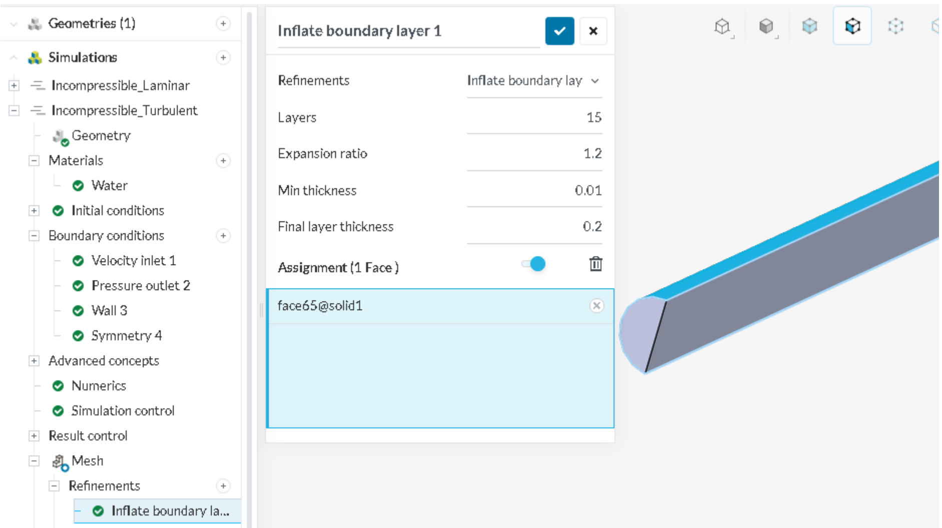

Figure 3.44: Do not forget to select the right face.

Click the tick to save. Go to mesh and click *"Create"*. It should take around 4 minutes to compute the mesh. Once is created we need to make sure that the first node is at a distance to the wall smaller than $y^+ = 5$. Select the Top view and rotate to have the pipe in vertical position.

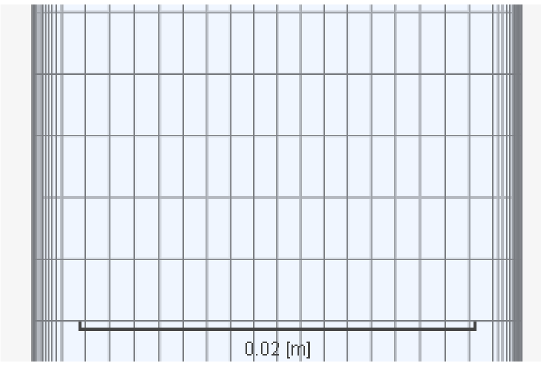

Figure 3.45: Refined mesh

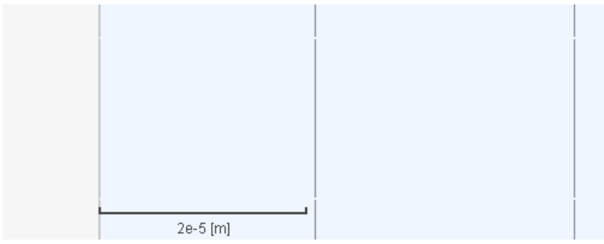

Figure 3.46: First node position

Notice that the first node position is almost at $y = 2e-5$ m with is less than the value calculated and showed at the table 3, so we can say that with this mesh the effects of the wall layer should appear in the results.

Finally, we go to *"Simulation Control"* and configure the parameters as the figure 3.47.

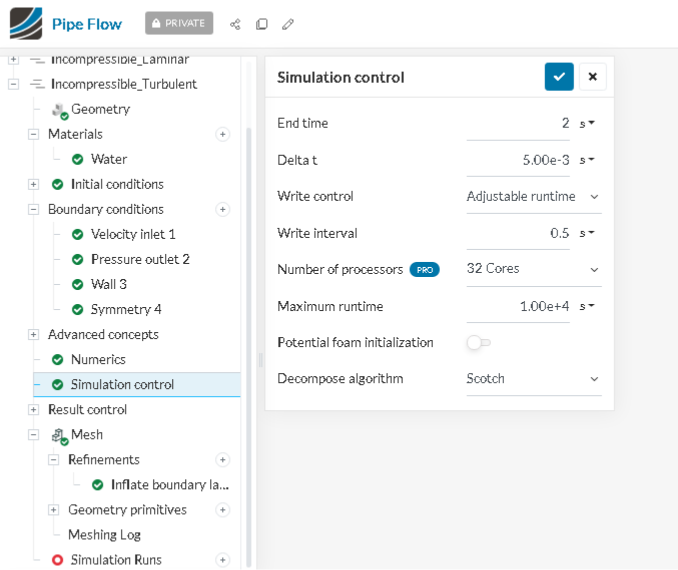

Figure 3.47: We should have all the parameters configured with a green circle with a white tick.

Set up the number of cores to 32 to reduce the computation time. Click the tick to save. Click to *"Simulation Runs"* and rename the new simulation: *"Refined_Mesh_Turbulent"*. Start the simulation.

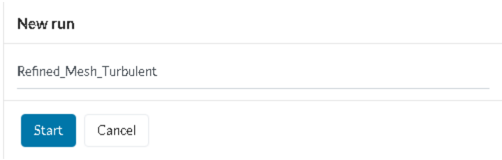

Figure 3.48: The simulation should take around 25 minutes.

You can check the convergence plots wile is running. Once the run is finished download the file and check in the SimScale post-process results if the simulation seems good.

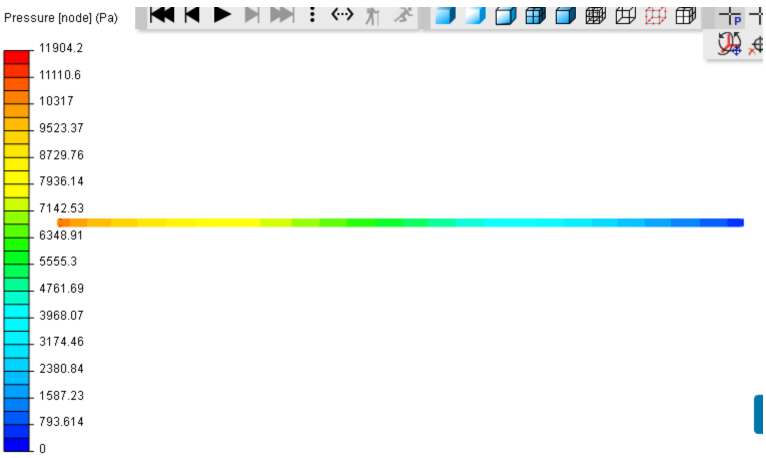

Figure 3.49: Pressure distribution along the pipe.

Open the decompressed file with ParaView.

3.3.5 Post-processing with ParaView

In this part, some of the post processing steps to follow will not be detailed if they have been explained for the laminar flow, because they have already been done (subsection 3.3.3). Therefore, plot the pressure distribution along the pipe and the velocity profile as done in the laminar case. Do not forget to change the time to 2.The plots should look like the following ones.

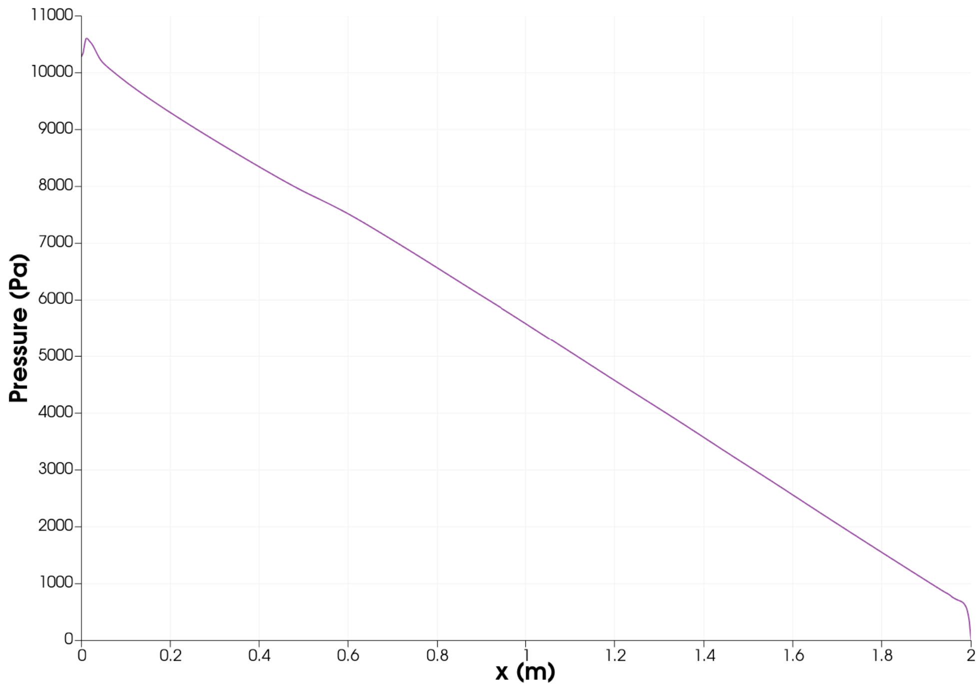

Figure 3.50: Pressure distribution of a Turbulent flow over the central line.

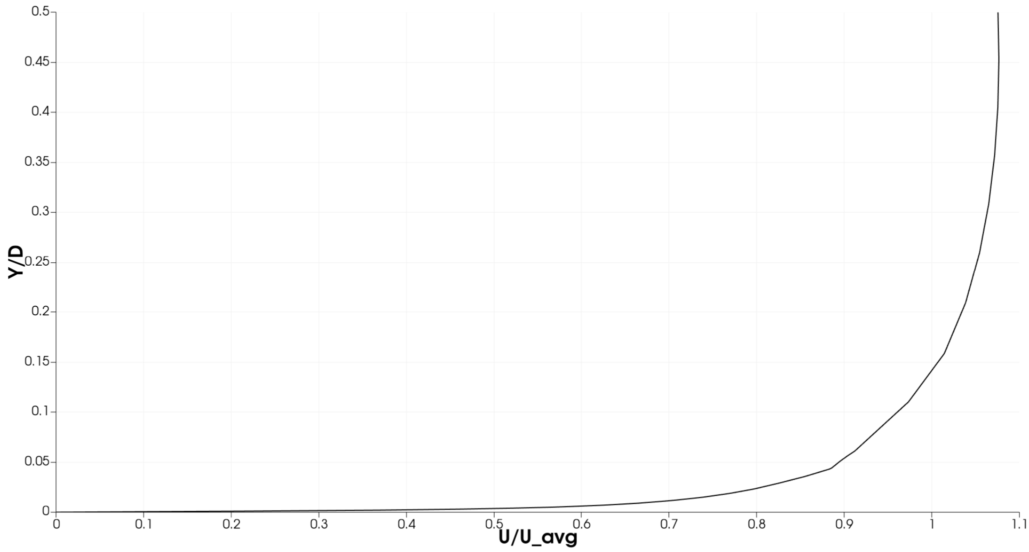

Figure 3.51: Velocity profile at the exit of the pipe for the Turbulent flow.

Once is done, we will plot the variation of the shear stress along the section at the exit of the pipe over y^+ and the u^+ over y^+. Firstly, open a new layout and click case foam.

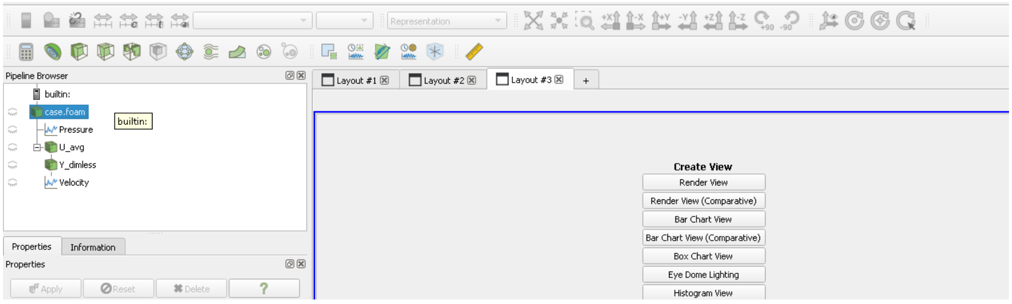

Figure 3.52: Select *"case.foam"*.

To calculate the laminar and turbulent shear stress we need the velocity gradient (equation 2). Go to *"Filters > Alphabetical > Python Calculator"*.

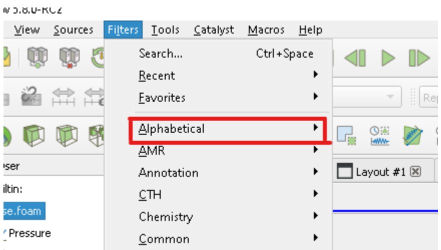

Figure 3.53: Select *"Python Calculator"*.

The Python calculator allows us to apply calculations that are available in Python. In this case, we will use *"gradient(U)"*. This expression returns, for a three component input (u,v,w) array, a 9 array component (du/dx, du/dy, du/dz, dv/dx, dv/dy, dv/dz, dw/dx, dw/dy, dw/dz). Each derivative has the same dimension as U [15]. Set the Python calculator properties as the figure 3.54 and click apply.

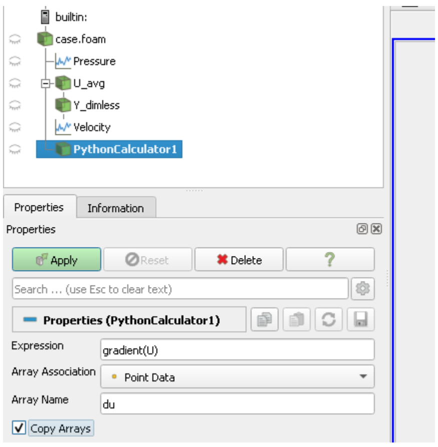

Figure 3.54: Configure the calculation.

Now, we will use the ParaView calculator to compute the turbulent and laminar shear stress and y^+. First, the tau turbulent. Click the calculator and configure it as the figure:

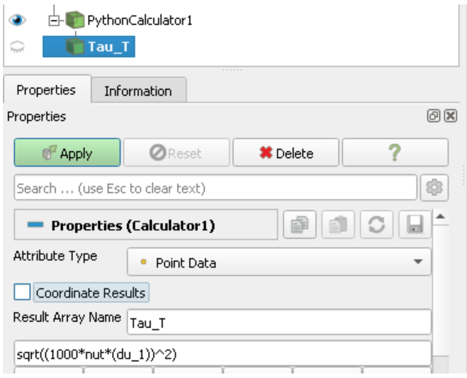

Figure 3.55: Calculate the turbulent shear stress.

Where *"nut"* is the kinematic turbulent viscosity or turbulence eddy viscosity that returns SimScale after the simulation. With this parameter we can calculate easily the turbulent shear stress. *"du_1"* is the derivative of the velocity in x direction over y. Click apply. Secondly, the tau laminar. Click the calculator symbol and configure it as the figure:

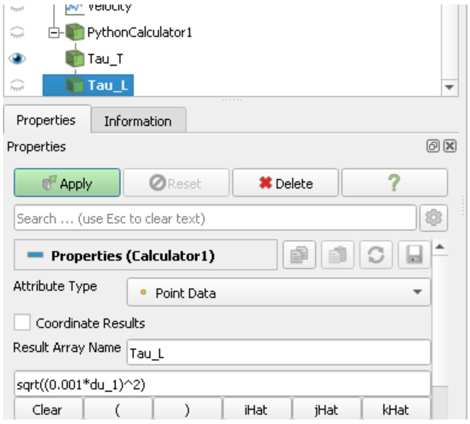

Figure 3.56: Calculate the laminar shear stress.

Rename the operations for the sake of the post process. Click apply. Thirdly, compute the total tau. Click the calculator symbol and add both stresses. We can find the laminar and turbulent shear stresses at the scalar list.

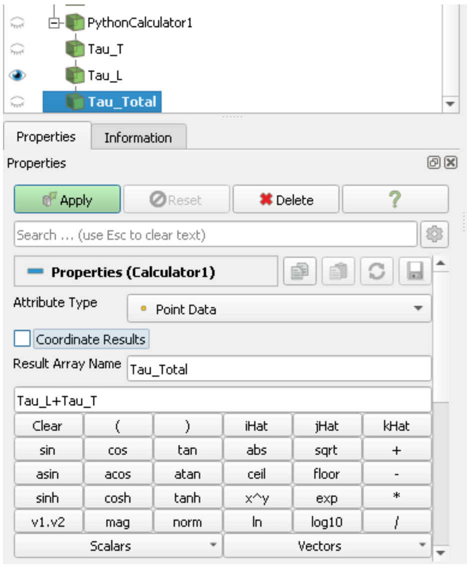

Figure 3.57: Calculate the total shear stress.

Finally, we have to calculate the y^+ as the equation 4. Click once again the calculator and configure it as the figure:

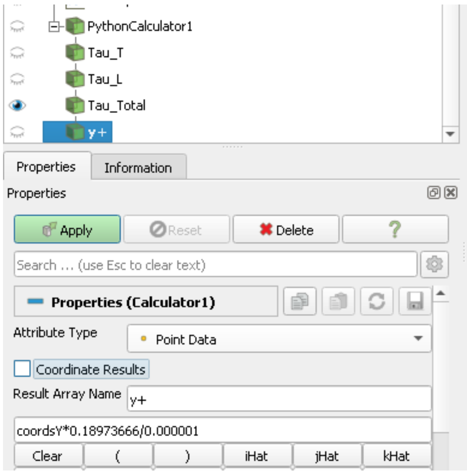

Figure 3.58: Calculate y^+.

Click apply. After all the parameters needed are calculated we can proceed to plot the shear stress. In this plot, tau is in the x axis and ParaView only allows one parameter in the x axis so we will have to create several plots and overlap them. So, create a plot over a line at the exit of the pipe (x=1.95) and select in the *"x array name"* Tau_T. Under *"Series parameters"* select only *"*y^+.

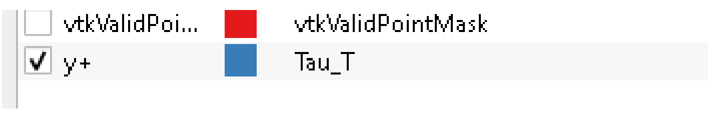

Figure 3.59: Rename y^+ to Tau_T

Scroll down and set the axis as in the figure. Use the logarithmic scale in the left axis.

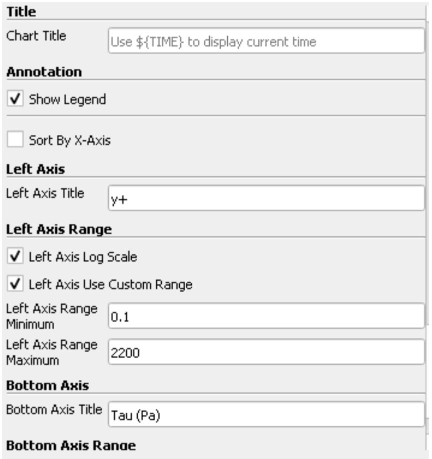

Figure 3.60: Configure the axis of the plot.

Create a new plot over the same line as before. This time in the *"x array name"* select Tau_L.

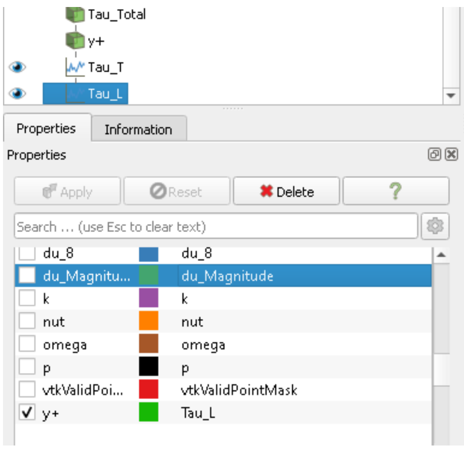

Figure 3.61: Rename y^+ to Tau_L and change the color.

Finally, create a new plot over line and select Tau_Total. Change the y^+ name and the color. The finale plot should look like the following one:

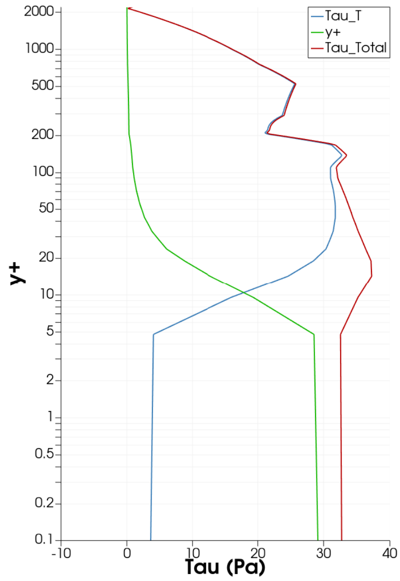

Figure 3.62: Shear stresses at the exit of the pipe for a turbulent flow over y^+ with the refined boundary layer mesh.

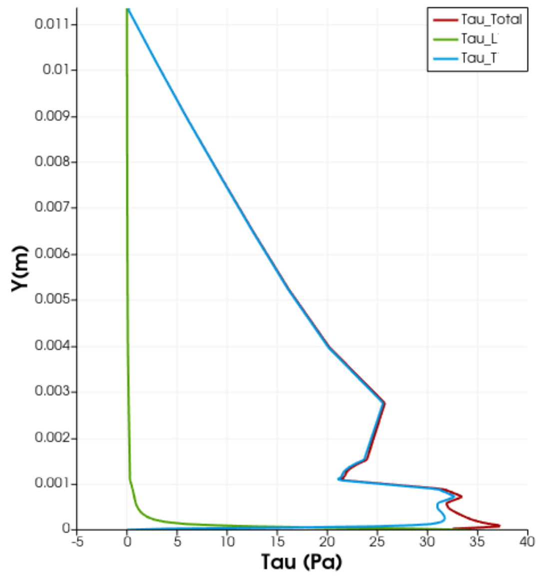

Figure 3.63: Shear stresses at the exit of the pipe for a turbulent flow over y with the refined boundary layer mesh.

As is possible to observe, there are some errors, which will be studied deeply in the next section 3.3.6. The last plot will be u^+ over y^+ at the exit of the pipe (similar to figure 3.2). First, we need to calculate the u^+. In order to do so, use the calculator and configure it as in the figure:

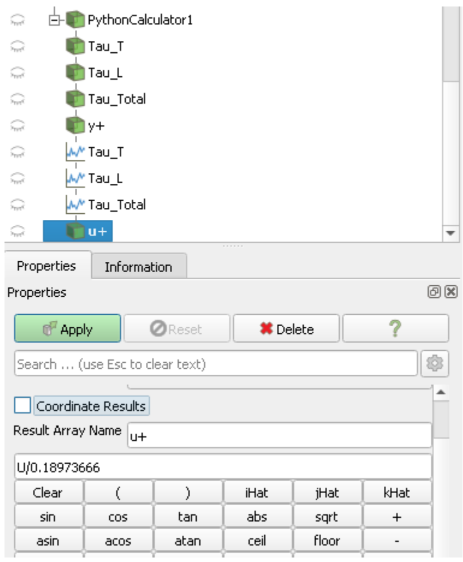

Figure 3.64: Calculate u^+.

Open a new layout and create a new plot over line (x=1.95) and click apply. In order to have a better plot, **rise the line resolution to 4000**. Select in the *"x array name"* y^+ and in the *"Series Parameters"*; *"u+_Magnitude"*. Configure the axis to have the right view and select the logarithmic scale for the bottom axis. We may have to change the *"Bottom Axis Range Minimum"* to 1 in order to allow ParaView to plot with a log.

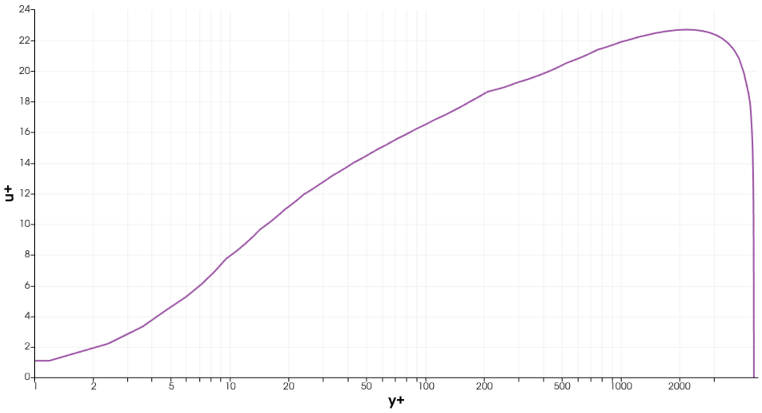

Figure 3.65: u^+ over y^+ at the exit of the pipe for the refined boundary layer mesh.

With this plots, now we can analyse the results. Save the state.

3.3.6 Numerical Results

The results have not been analysed during the tutorial for the sake of it. Therefore, once is done, we can start extracting some conclusions.

Firstly, it is mandatory to check if the results are close to reality. In order to do so, the pressure drop over the pipe and the velocity profile will be used. For the laminar flow, the expression used to calculate the pressure drop over the pipe analytically is showed at the equation 8 and the result is showed at the table 2. In the figure 3.29 it is showed the pressure drop for the simulation. It is important to remember that the Darcy equation do not consider the entrance pressure drop, neither when the flow is not fully developed. Taking this into account, a linear regression has been made for the points were flow is developed (3.66) and the pressure drop is:

$$\Delta P_{simulation} = 1.02$$

Pa

If the error is calculated it can be seen:

$$Error = \frac{\Delta P - \Delta P_{simulation}}{\Delta P} = \frac{1.024 - 1.02}{1.024} = 3.85 \cdot 10^{-3} < 1\%.$$

Therefore, for the **laminar flow** the simulation is very close to reality. The velocity profile (figure 3.36) have a parabolic form as expected. From the figure 3.66 it can be seen that the flow is fully developed before it is calculated with the equation 9. Nevertheless, at the figure 3.67 the velocity profile is not completely developed after the length calculated.

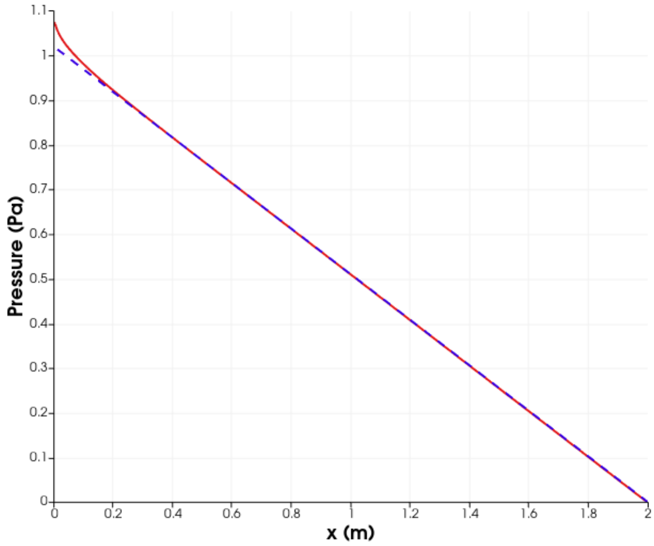

Figure 3.66: Pressure distribution with a linear regression of the fully laminar developed flow points.

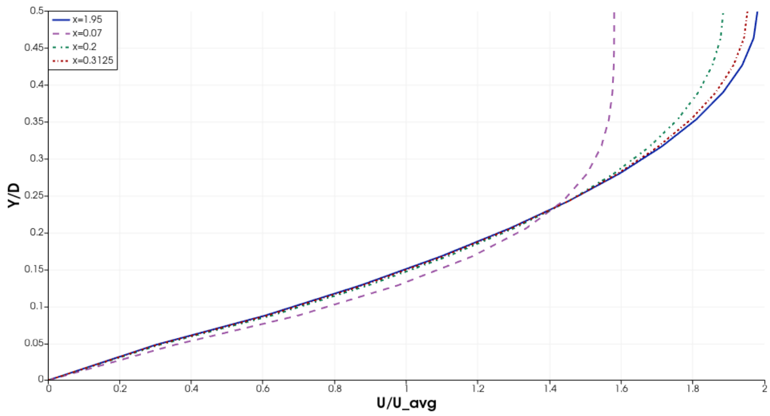

Figure 3.67: Velocity profile at different positions along the pipe.

For the **turbulent flow**, for a refined mesh at the boundary layer, the pressure drop is smaller than expected (figure 3.50), having an Error = 9%, which is considerable. The velocity profile (figure 3.51) is correct, with a higher variation near the wall. At the figure 3.62 it is possible to observe the shear stresses at the exit of the wall and that with this mesh, they are well captured, with the exception at y^+=200 were there is a perturbation. It could be caused by an small erroneous change in the velocity profile which causes a bad computation of the gradient. Apart from this, it is showed at the figure 3.65 that do not correspond perfectly to the theory at a small y^+, it should be tending to 0. The behaviour at high y^+, where u^+ decreases, happens because it is close to the upper wall.
If these results are compared with the results for simulation with an homogeneous mesh without refinement at the boundary layer (figures 3.68, 3.73, 3.72 and 3.69) clearly confirms the affirmation that the results depends highly on the mesh (it has been followed the same procedure at the post-processing as in the tutorial).

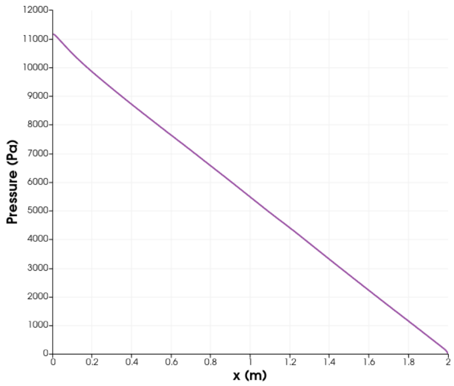

Figure 3.68: Pressure distribution of a turbulent flow for an homogeneous mesh.

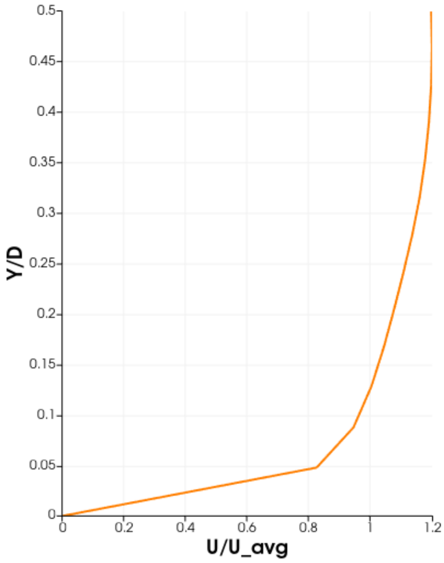

Figure 3.69: Velocity profile at the exit of a turbulent flow for an homogeneous mesh.

With this mesh, the pressure distribution is closer to the analytical results (Error = 3%). There is a bigger error in the refined mesh and this may occur because the refinement is not perfect (figure 3.70) and the transition between small cells and bigger cells creates some perturbations (figure 3.71). This error sometimes appears with the mesh solver and dealing with this situation requires full knowledge on the mesh configuration and management of the detailed parameters.

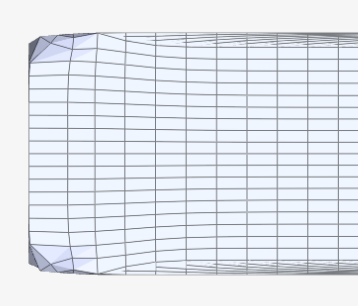

Figure 3.70: Transition from small cells to bigger at the refined mesh.

Figure 3.71: Pressure perturbations caused by a bad transition.

The velocity profile is not as continuous as before. If this profile is not good enough, the stresses will not be well captured (figure 3.72). Theoretically, the laminar shear stress should be higher that the turbulent near the wall, which is not the case. This confirms the difficulty to capture this effects for even for simple geometries without having huge number of nodes which would suppose a large calculation time.

It draws a lot of attention the fact that the u^+ over y^+ plot seems better than figure 3.65. This is probably by the fact that in the refined mesh, the values from y^+=0 to 5 are obtained with a linear interpolation and the homogeneous mesh uses "wall-functions to

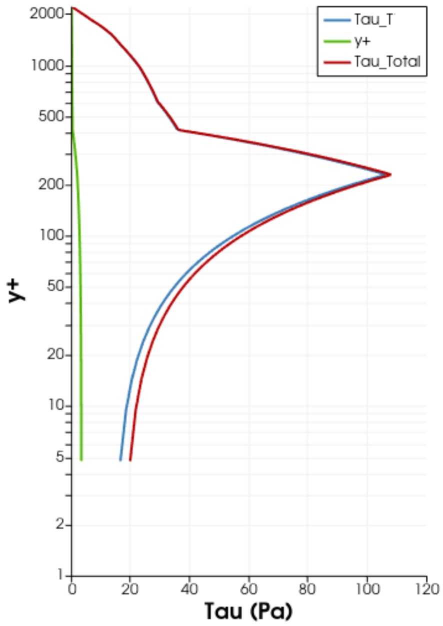

Figure 3.72: Shear stresses for a turbulent flow for an homogeneous mesh.

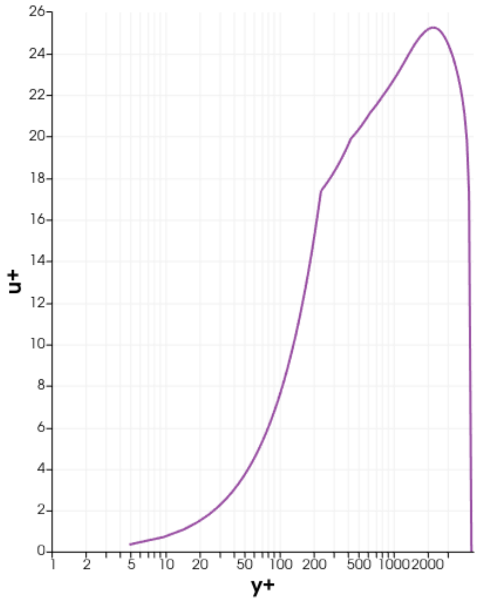

Figure 3.73: u^+ over y^+ at the exit for a turbulent flow for an homogeneous mesh.

resolve the near-wall region" as it was said before. Nevertheless, in the figure 3.73, near $y^+ = 200$ there is a non-derivable point that causes a bad computation of the shear stress (figure 3.72) and at low y^+ the values are not correct. For a better shape, it should be guaranteed an smooth velocity profile with an $y^+ = 5$, at least.

4 External Flow

The next case study it is going to be the flow over a thin plate, another classical example, done multiple times in class. The objective in this example is to see how the results varies with a laminar or a turbulent model and which is the best one. In addition to this, as in the previous case, the mesh will take an important roll in this study.

4.1 Theoretical Framework

For the turbulent flow regime, the theory about the boundary layer is very similar to the one considered before (section 3.1). The main characteristic of the boundary layer could be the thickness δ. For a thin plate, the expression to calculate this value, for a laminar and turbulent flow are presented [3]:

$$\delta_{laminar} = \frac{4.91x}{Re_x^{1/2}} \tag{11}$$

$$\delta_{turbulent} = \frac{0.38x}{Re_x^{1/5}} \tag{12}$$

In this case, it will be calculated analytically the thickness of the boundary and then compared with the result obtained with the simulation. As it can be observed it the equations 11 and 12 the layer is strongly related to the Reynolds number and to the the characteristic length of the geometry, which, for a flat plate, is the length of the plate in the flow direction. This means that the flow be laminar at the beginning of the plate but it will became turbulent at a certain point. Therefore, it will be necessary to calculate the boundary layer with a laminar model and then the same case with a turbulent model to ensure the right δ of the boundary layer.

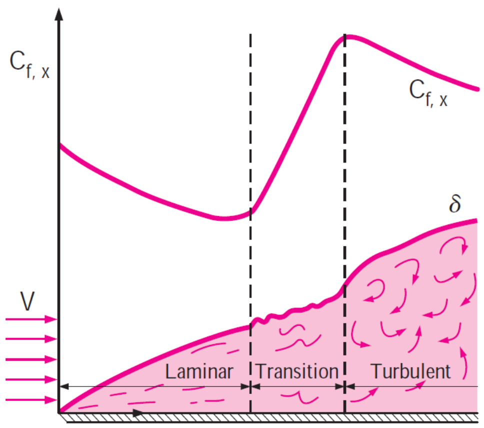

Figure 4.1: The variation of the local friction coefficient for flow over a flat plate [3].

In this simulation, it will be used a refined mesh near the plate trying have the first node at $y^+ = 5$ as done at the previous case. It is imperative to have a refinement at the plate because if the node is too far away from it, the simulations cannot capture the effects of the boundary layer (figure 4.1). For example, the shear stress is one of this effects. With this coefficient can be obtained the friction coefficient (C_f), which will be one of the

main focus in the post process.
In order to calculate the grid spacing for $y^+ = 5$, it will be followed the same procedure as with the pipe flow (3.1). It will be used equations 4 and 6, nevertheless, the friction coefficient for a turbulent flow will be calculated as in [4]:

$$C_{f,x} = \frac{0.026}{Re_x^{1/7}} \tag{13}$$

In addition to this, it is possible to calculate the first node positions with some websites as [16].
Later on, the values of τ through the flat plate obtained from the simulation will be compared with the ones obtained theoretically with the equation 14 from [4].

$$\tau_w = \frac{C_f \rho U^2}{2} \tag{14}$$

The analytical values of local friction coefficient will be calculated with the equation 15 for $Re < 3.5 \cdot 10^5$, laminar regime and the equation 16 for $Re > 3.5 \cdot 10^5$, turbulent regime, from [3].

$$C_{f,x} = \frac{0.644}{Re_x^{1/2}} \tag{15}$$

$$C_{f,x} = \frac{0.059}{Re_x^{1/5}} \tag{16}$$

Finally, it will be used the equation 17, extracted from the equation 14 to calculate the friction coefficient form the numerical values of τ_w obtained from the simulation.

$$C_f = \frac{2\tau_w}{\rho U^2} \tag{17}$$

For parallel flow over a flat plate, the pressure drag is zero, and thus the drag coefficient is equal to the friction drag coefficient, or simply the average friction coefficient [3]. Some expressions are used to calculate analytically the friction coefficient integrated to the hole plate, as the equations 18 and 19 for a laminar and turbulent flow, respectively:

$$C_f = \frac{1.33}{\mathrm{Re}_L^{1/2}} \quad \mathrm{Re}_L < 5 \times 10^5 \tag{18}$$

$$C_f = \frac{0.074}{Re^{1/5}} \quad 5 \times 10^5 \leq Re_L \leq 10^7 \tag{19}$$

And from [4]:

$$C_D = 2C_f(L) \tag{20}$$

$$C_D = \frac{7}{6}C_f(L) \tag{21}$$

For turbulent and laminar regime, respectively. For the cases were the flow has not become completely turbulent and the transition region cannot be neglected, the following expression to calculate the average friction coefficient is used [3]:

$$C_f = \frac{0.074}{Re^{1/5}} - \frac{1742}{Re_L} \quad 5 \times 10^5 \leq Re_L \leq 10^7 \tag{22}$$

4.2 Meshing Methods

In this case, the mesh selected is a Hex-dominant parametric based on SnappyHexMesh as the Pipe Flow (section 3.3). It will be an homogeneous mesh with refinement near the plate (similar to the refined mesh used in the Pipe Flow), with the first node placed at $y^+ < 5$.

4.3 Flat Plate Case Study

As said before, it is presented a flat plate immersed in a fluid (water) which moves at a constant velocity. Three simulations will be analysed for three different velocities in order to observe the behaviour of the fluid in a laminar, transient and turbulent flow. The dimensions and characteristics of the plate and fluid are presented in the following table.

Plate's longitude	L	0.35	$[m]$
Plate's thickness	e	Infinitesimal	
Fluid's density	ρ	1000	$[kg/m^3]$
Fluid's dynamic viscosity	μ	0.001	$[kg/m.s]$
Inlet velocity*	u	10;2;1	$[m/s]$
Reynolds number at L*	Re_L	$3.5\cdot10^5$;$7\cdot10^5$;$3.5\cdot10^6$	$[-]$

Table 4: Data of the problem, flat plate. *The three values of the velocity/Reynolds number are for the turbulent, transient and laminar regime, respectively.

This plate's longitude has been chosen to compare the turbulent results with the article [2] and the fluids characteristics had been simplified for the sake of the problem.

4.3.1 Analytical Results

Before running the simulations, as part of the pre-processing, the analytical results for the flat plate are presented. The first simulation done is a laminar flow (v = 1 m/s), where the equations 11, 15 and 14 to calculate the boundary layer, the local friction coefficient and the shear stress along the plate, respectively. This results are showed and compared with the numerical results in the subsection 4.3.4. The same procedure has been followed for the other two simulations (with v = 2m/s and v = 10 m/s) but with the corresponding expressions for each regime. The table 5 shows the result of the average friction coefficient over the entire plate calculated with the equations 18, 19 and 22 for laminar, turbulent and a flow with a transient region not negligible, respectively.

$Cf_{laminar}$	$2.248 \cdot 10^{-3}$
$Cf_{turbulent}$	$3.634 \cdot 10^{-3}$
$Cf_{transient}$	$2.525 \cdot 10^{-3}$

Table 5: Theoretical average friction coefficient for inlet velocities of 1, 2 and 10 m/s, respectively.

4.3.2 SimScale Tutorial

Once the pre-processing is finished, it can be started the simulation configuration. In order to do so, firstly, create a *"New Project"* from your SimScale Dashboard. You can name it *"Flat Plate Flow"*. Click this **link** to download the solid used for the simulation. Import the geometry and create a new incompressible simulation.

Laminar Regime

The first simulation that we are going to run is with a completely laminar flow, therefore, select a *"Laminar"* turbulence model and name the new simulation *"Incompressible_Laminar"*

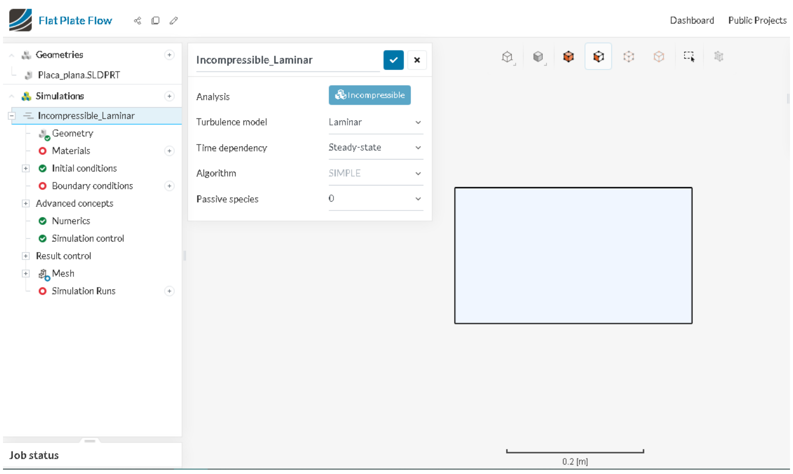

Figure 4.2: Configuration of the laminar simulation.

Then, select the material and set his properties, click *"Materials"* > *"Water"* > *"Apply"* and introduce the Kinematic viscosity value; $1e-6m^2/s$, and the density; $1000kg/m^3$. Now, configure the boundary conditions as in the figures 4.3, 4.4, 4.5, 4.6 and 4.7.

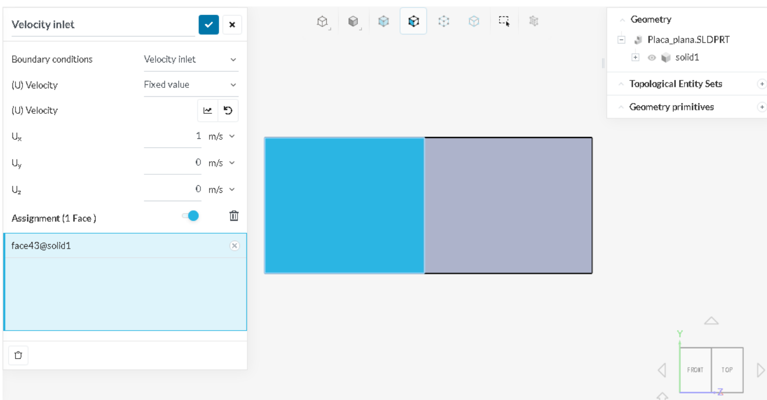

Figure 4.3: Velocity inlet.

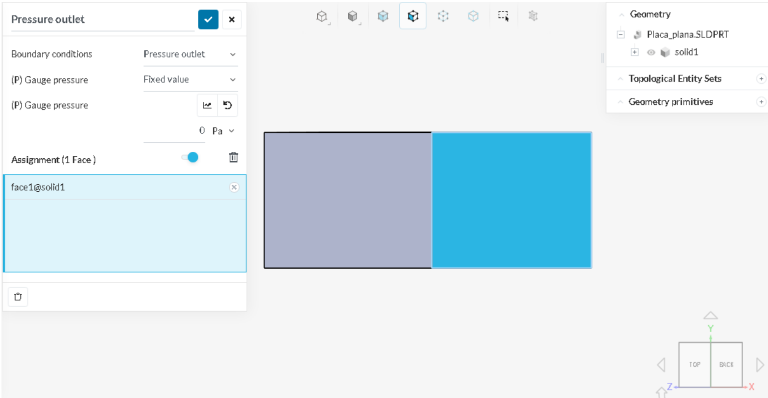

Figure 4.4: Pressure outlet.

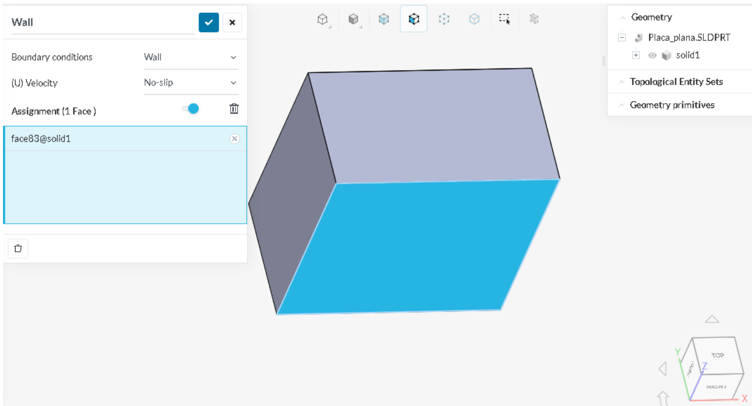

Figure 4.5: No slip condition.

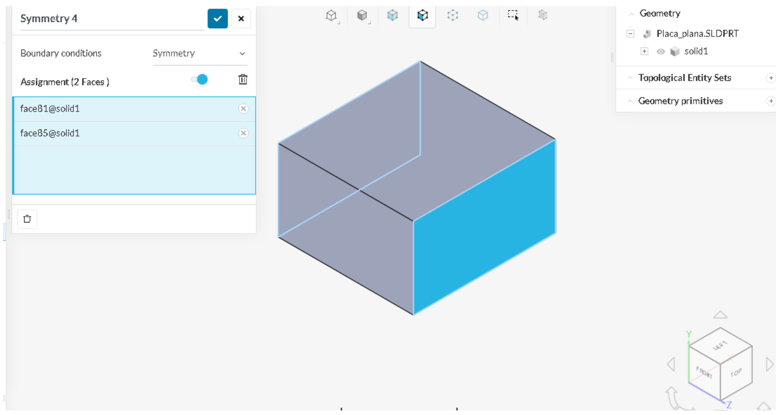

Figure 4.6: Symmetry condition.

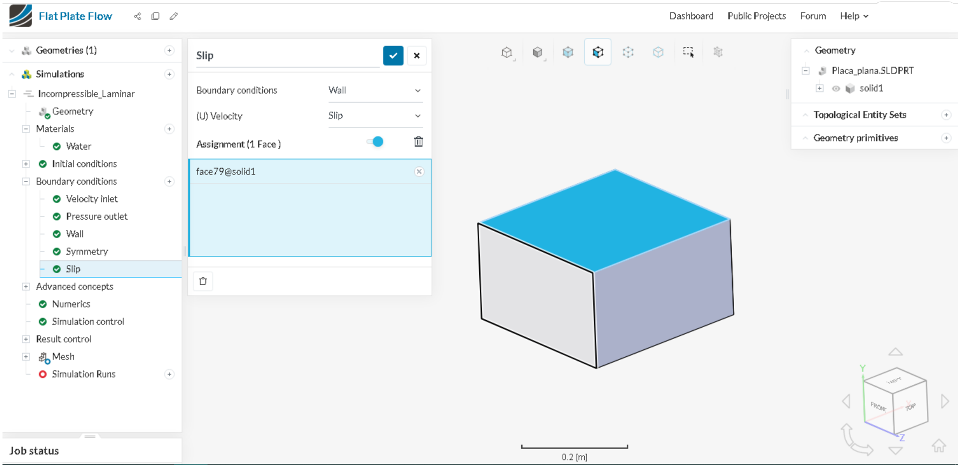

Figure 4.7: Slip condition.

It is important to set properly the Slip condition at the top of the solid and also the symmetry condition at both sides to have a 1 m plate depth.
After setting up the boundary conditions, the mesh can be created. As it was said before (section 3.3), a laminar flow does not need a mesh refinement with the first node at $y^+ = 5$, but to ensure to capture a possible transient region, the mesh will be refined. As we use a Laminar model for the analysis, the calculation time increase will not be too much. Go to *"Mesh"*, select the meshing algorithm *"Hex-dominant parametric (only CFD)"*. Rename it *"Mesh 1_Laminar"*. Click *"Refinements"* > *"Inflate boundary layer"* and set the parameters as the figure 3.10. Select the face of the solid that represents the plate.

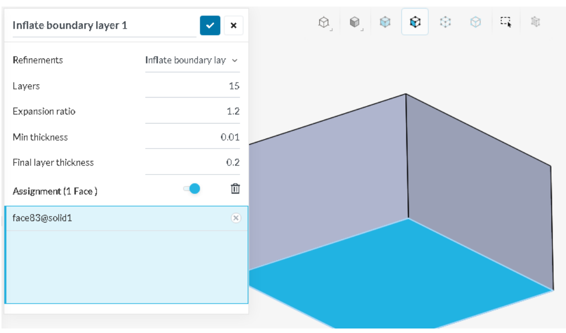

Figure 4.8: Boundary layer refinement.

Generate the mesh. From SimScale's TOP view the mesh should look like this one:

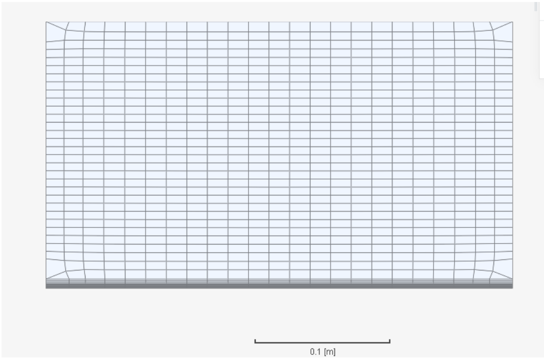

Figure 4.9: Mesh for the laminar flat plate case.

Then go to *"Simulation Control"*, an configure it as the following figure.

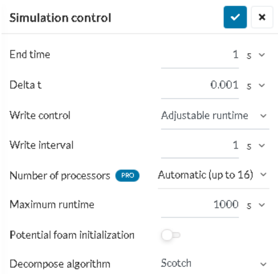

Figure 4.10: Simulation control configuration.

The writing interval, for a steady flow, can be the same as the end time. Before running the simulation, we need to obtain the shear stress from the simulation. In order to do so, we need to go to *"Result Control"* > *"Field Calculations"* > *"Wall fluxes"* and click the tick to save. Once all is configured, and no red circles are in the workbench (except from the simulation run) Run the simulation.

When the simulation is finished, make sure that the convergence plots are correct (the residual should be stabilized) and download the results. To start the post-processing right away, go to section 4.3.3.

Turbulent Regime

The second simulation to create is for a velocity inlet of 10 m/s. Therefore, hide the *"Incompressible_Laminar"* simulation and create a new one. Rename it *"Incompressible_Turbulent"* and make sure to choose the *"k-omega SST* turbulence model. Set the *"Materials"* and *"Boundary Conditions"* as before, but now with an inlet velocity equal to **10 m/s** and the wall no-slip condition to *"Full resolution"*.
Now is the time to select the mesh, unfortunately, if we calculate the $y(y^+ = 5)$, the value is smaller than in the laminar case. Hence, we need to refine the mesh even more than before. So, create a new mesh with the following parameters:

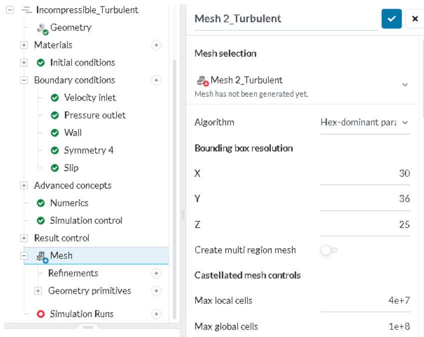

Figure 4.11: Mesh turbulent configuration.

In this case we have to change a parameter called *"Min determinant"* due to the fact that the area of some cells will be too small. This quality parameter is described at [17] like this: *"Minimum normalised cell determinant. This is the determinant of all the areas of internal faces. It is a measure of how much of the outside area of the cell is to other cells. The idea is that if all outside faces of the cell are 'floating' (zeroGradient) the 'fixedness' of the cell is determined by the area of the internal faces"*. If we do not change this parameter, the boundary layer will not be computed.

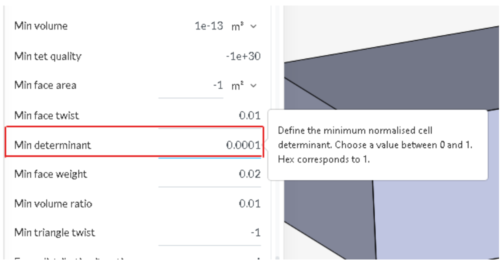

Figure 4.12: Minimum determinant value.

Now, refine the mesh near the plate as in the figure 4.13.

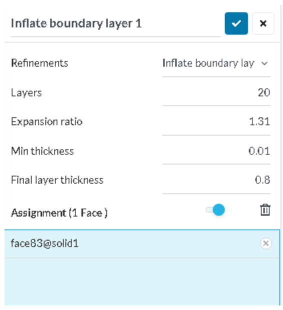

Figure 4.13: Inflate boundary layer.

Generate the mesh. Configure the simulation controls as in the laminar case. Do not forget to compute the *"Wall fluxes"* and, then, run the turbulent simulation. Once the simulation is finished, make sure that the residuals have stabilized and download the results.

For the third simulation, it is not necessary to create a new simulation. Just change the velocity value from the boundary conditions to 2 m/s and run a new simulation called *"Run 2_Transient"*.

4.3.3 Post-processing

Open the **Laminar** results with ParaView. The post-processing with this program is going to be a little bit more tricky than in the Pipe Flow. We will plot the boundary layer, the velocity profile and shear stress over the plate. In addition to this, we will plot some graphs to see if the results are acceptable; y^+ vs u^+ at the end of the plate, u/U_{avg} vs y/δ and, finally, Re_L vs C_D.

The first plot is the velocity profile. To do this plot, we need to create a *"Plot over line"* at different x positions. In this case, the x chosen can be seen in the figure 4.14. ParaView allows us to change the color of each profile (and the line style) and change the legend at our will. In this case, the first line has been placed at x = 0.001 m.

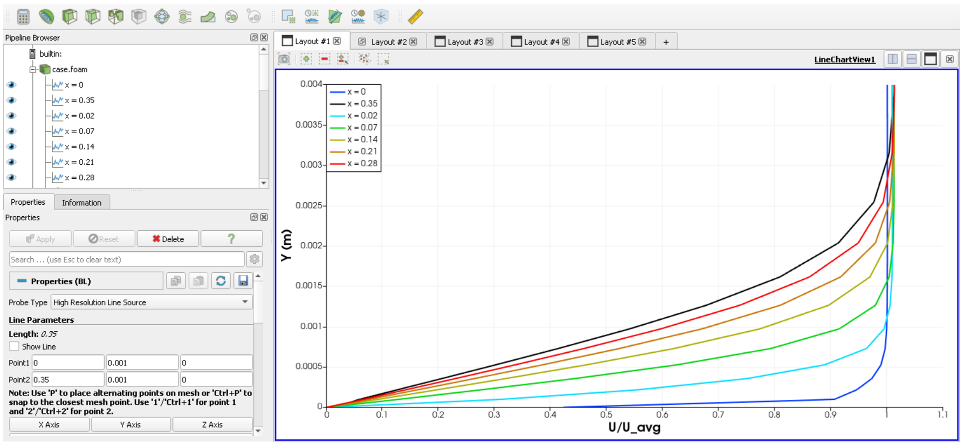

Figure 4.14: Laminar velocity profile configuration.

The next plot to create is the boundary layer. Unfortunately, ParaView does not obtain this plot automatically, therefore, we need to select the points were $u = 0.99U$. Firstly, we will plot the analytical results and then they will be compared with the simulation result. So, use the *"Calculator"* to calculate the Reynolds number over x (Re_x) and use the laminar and turbulent expressions (equations 11 and 12). Plot δ_T and δ_L over x (figure 4.14). Then, go to the velocity profile and write the points were the boundary layer ends in a *"text file"* (figure 4.15.

To upload the file just click the open symbol at the top left. The resultant plot should look like the following one:

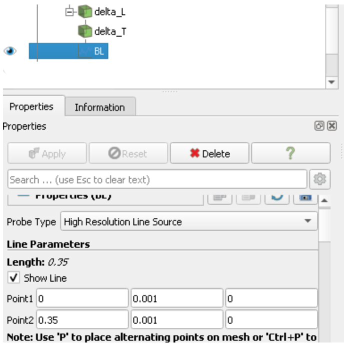

Figure 4.15: Delta line configuration.

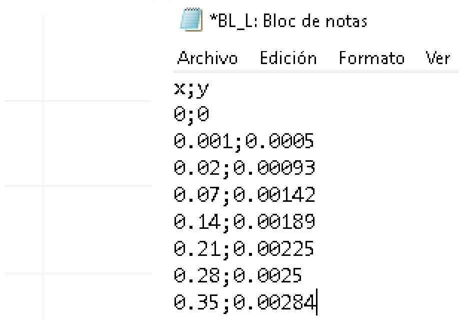

Figure 4.16: Boundary layer points for the laminar case.

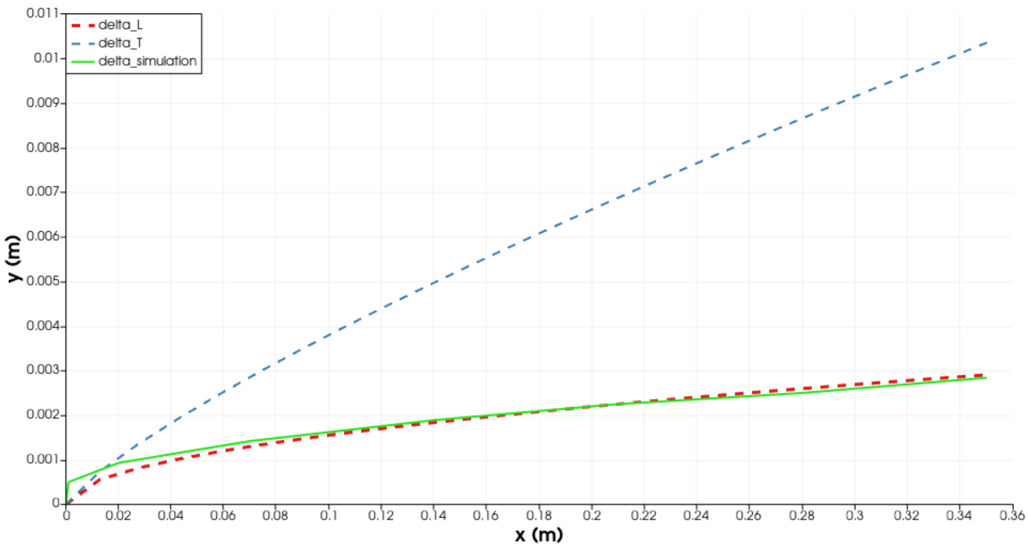

Figure 4.17: Laminar boundary layer.

As it can be observed, the boundary layer from the simulation is adapted completely to the analytical result. The only difference is that at the beginning of the plate the transition is not so "smoothly".

The following plot is the shear stress over the plate. From the Re_x calculated before, calculate the $C_{f,x}$ from the equation 15 and then T_w from the equation 14. Plot this result in a line at y = 0 and also select in the *"Series Parameters: WallShearStress_Simulation"*. The plot obtained is:

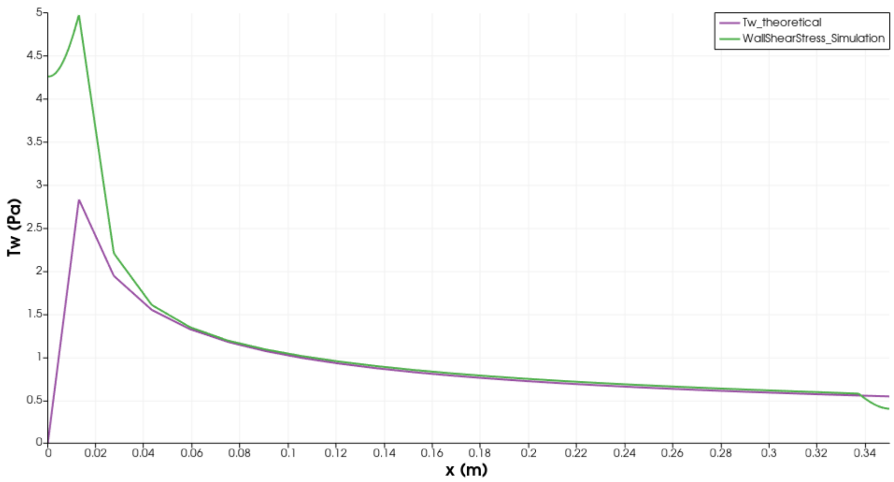

Figure 4.18: Laminar Shear stress over the plate.

Maybe, for the turbulent and transient case, you will need to create the line very close to the wall but not at y = 0, because some values there are not computed. So you can try with y = 1e-6 m.

Finally, the last plot to compute is u^+ vs y^+ (figure 4.19. To create it, we need to use the *"Calculator"* and calculate the overall friction coefficient (equation 18) for laminar flow) and the shear stress (equation 14) to calculate the shear velocity (equation 6) and, lastly, u^+ and y^+.

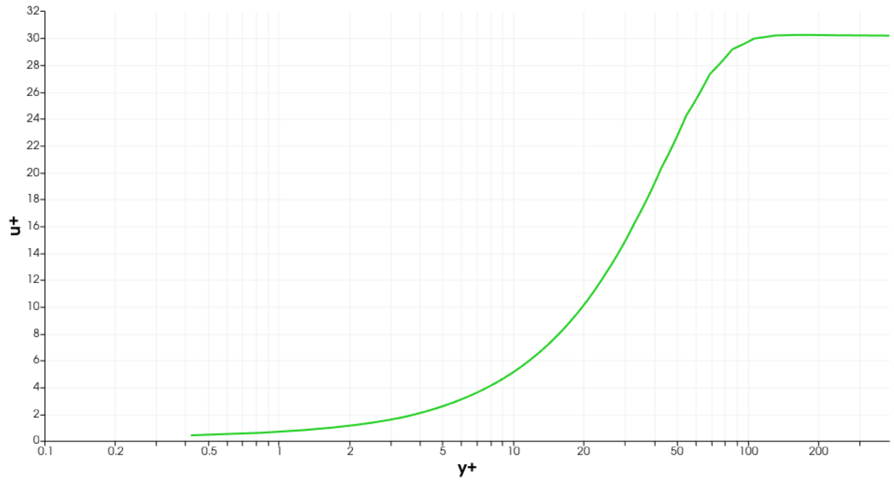

Figure 4.19: u^+ vs y^+ at the end of a flat plate, laminar flow.

To obtain a better result for this plot, select a line with a smaller length and rise the resolution (by default SimScale uses 1000 points to create a line). For the turbulent and transient case, it has to be followed the same procedure but using the correspondent expressions explained at the section 4.1. The results are showed in the following section (4.3.4).

4.3.4 Numerical Results

The laminar results were showed in the previous section and they looked very promising and similar to the theory. The figure 4.19 confirms this supposition, the profile of the plot is very good and it is stabilized $y^+ \approx 100$, that it is when the u = U. This is another way to see where the boundary layer ends. If it is used the equation 4 it is obtained:

$$y = \frac{y^+ v}{u_t} = \frac{100 \cdot 10^{-6}}{3.3527 \cdot 10^{-2}} = 2.9827 \cdot 10^{-3} m$$

Which is very close to the value obtained at the figure 4.17. The other particularity of this case has been the peak difference obtained at the shear stress plot (figure 4.18). The main issue here is that at the analytical calculations at $Re_x = 0$, ParaView is dividing by 0 to calculate the friction coefficient. Theoretically, the shear stress should tent to infinite, having a vertical asymptote at 0. But, after some investigation, it has been observed that the software sets this first value to 0 (instead of ∞) and then it does a linear interpolation until the next point calculated (at a certain distance, x). This "next point calculated" corresponds to the peak. Hence, this means that the wall shear stress in the simulation is closer to the analytical result than it is showed with ParaView.

Once the laminar case is closed, it can be analyze the next two cases. It has been followed the same procedure as in the laminar case but with the laminar/transient/turbulent expressions depending on the situation. The results obtained for a velocity inlet v = 2m/s:

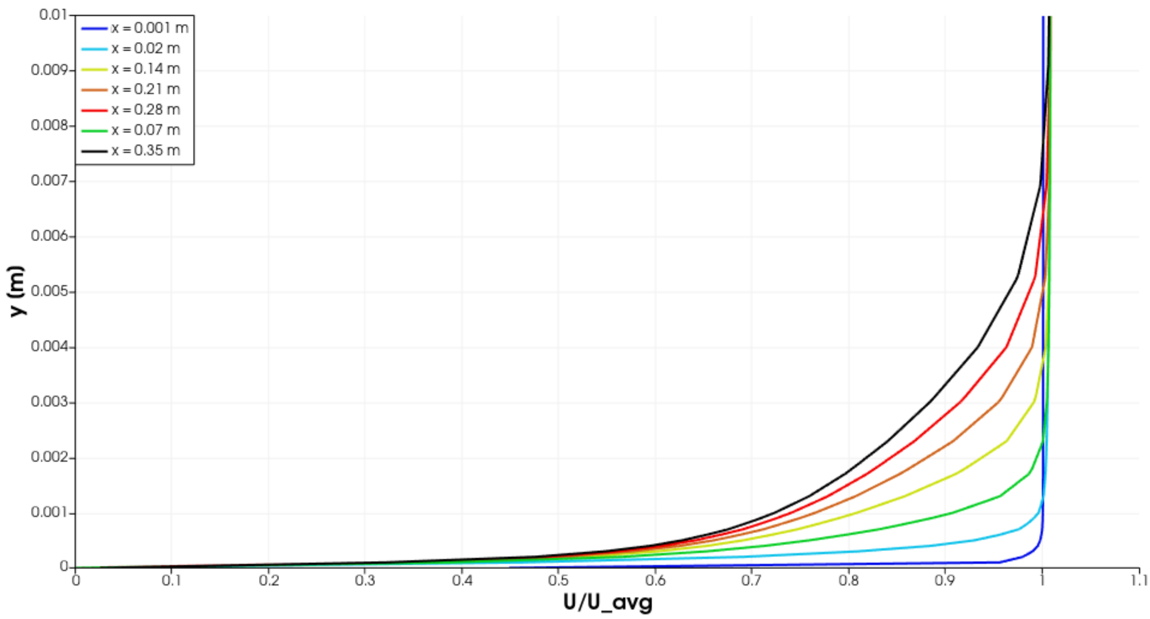

Figure 4.20: Velocity profile at a flat plate, v = 2 m/s.

The first plot is the velocity profile, and it seems well computed. There are no sharp points and the lines are homogeneously distributed. A bad mesh, with, for example, too wide cells in the y direction could suppose a choppy profile (figures 4.21, 4.22). In this case, there is a big gap between the end of the refinement and the next node. For this reason, in the figure 4.22 can be seen a sharp edge right at the end of the refinement.

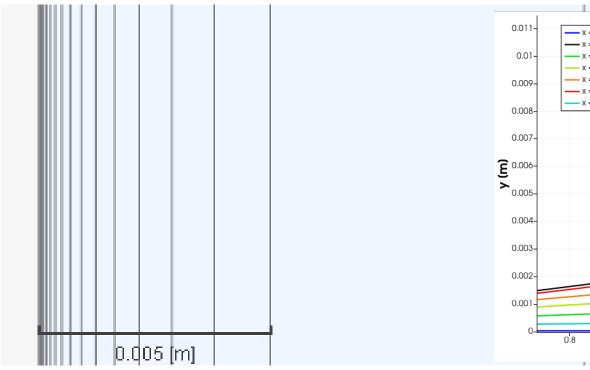

Figure 4.21: Mesh computation test.

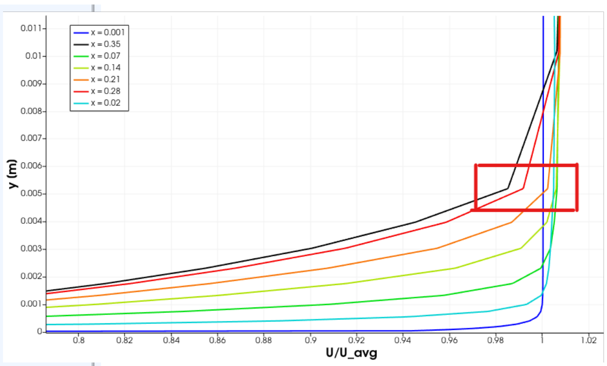

Figure 4.22: Bad velocity profile.

This was one of the test meshes that were not good for the tutorial. If the velocity profile is not good, then it is not also the boundary layer and, therefore, the wall shear stress.

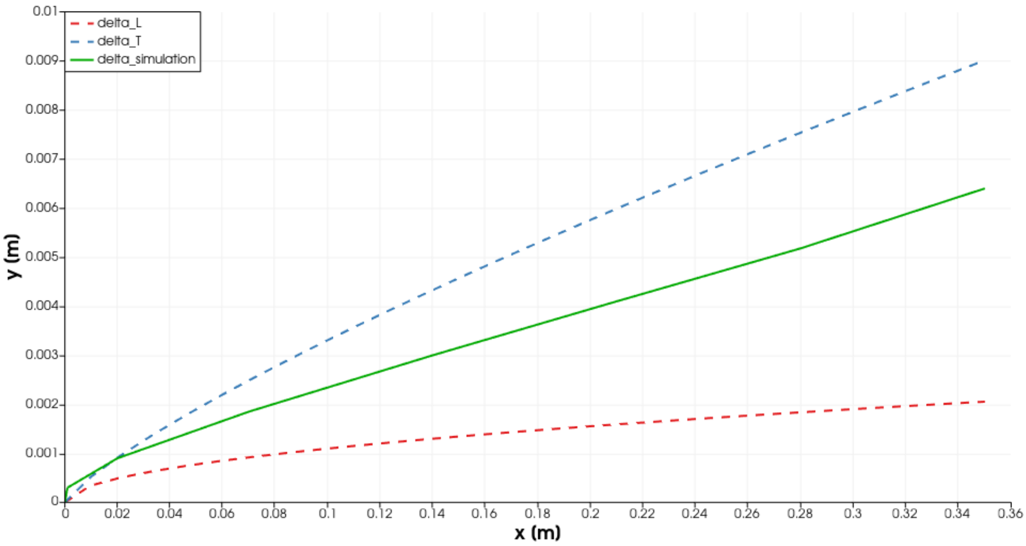

Figure 4.23: Boundary layer at a flat plate, v = 2 m/s.

The figure 4.23 is very interesting. There is no change in the boundary layer when the flow passes form laminar to turbulent (that, theoretically is at $x \approx 0.175m$). This bad computation is maybe due to the fact that transient flow characteristics are not well resolved by a steady-state solver [18].

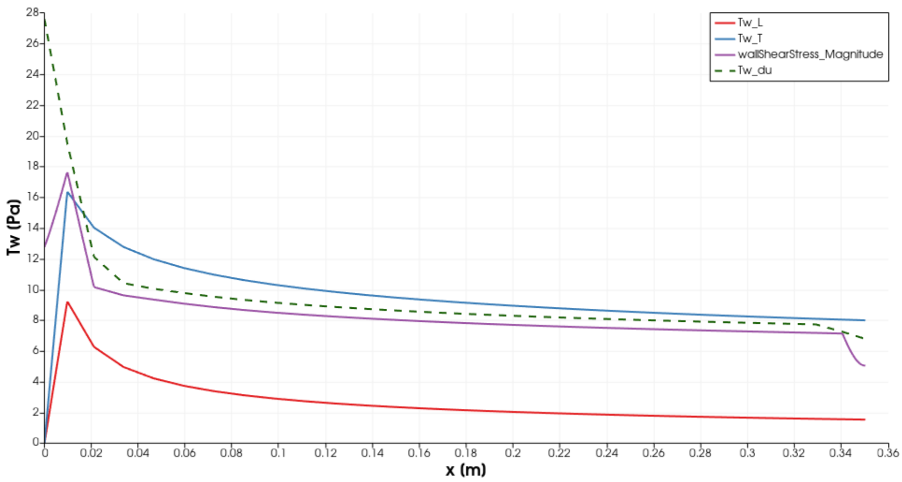

Figure 4.24: Shear stress over a flat plate, v = 2 m/s.

The shear stress plot shows us that the value of the simulation stays close to the turbulent modeling. As with the boundary layer plot, at the fist half of the plate the shear stress distribution should be similar to the laminar (4.18) but it is not the case. In the figure 4.24 it can also be seen a dashed line. This belongs to the shear stress distribution obtained with the gradient velocity from the simulation at the wall (equation 3). This plot shows that this is also a good way to obtain τ_w.

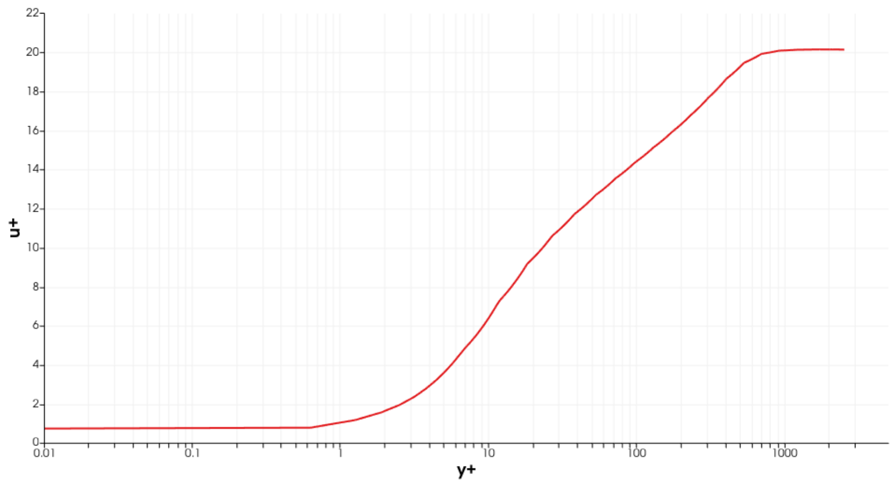

Figure 4.25: u^+ vs y^+ at the end of a flat plate, v = 2 m/s.

This figure tells us that the simulation results look good. And for v = 10 m/s:

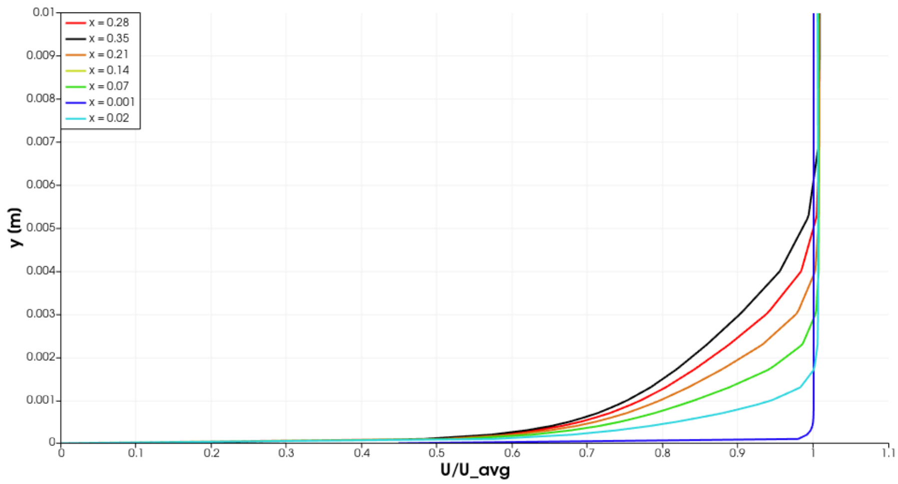

Figure 4.26: Velocity profile at a flat plate, turbulent flow.

The first plot is the velocity profile, and it seems well computed. There are no sharp points and the lines are homogeneously distributed.

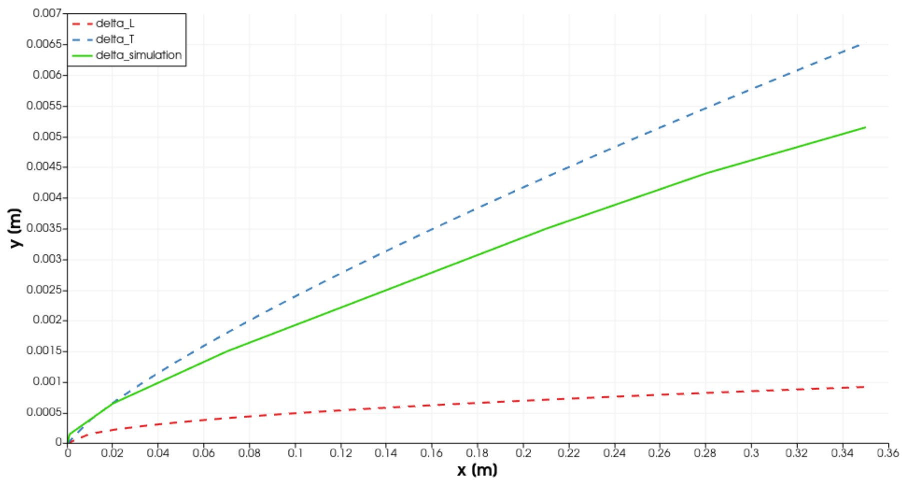

Figure 4.27: Boundary layer at a flat plate, turbulent flow.

This time, the boundary layer is closer at the turbulent distributions than in the transient simulation (figure 4.23). Nevertheless, it is still relatively far from the optimal result. This can be by the fact that maybe the domain was not tall enough and, then, the free stream velocity is slightly higher than 10 m/s, because the section is reduced by the boundary layer. As the points has been selected "manually" at $0.99 \cdot U$, being U = 10 m/s. the boundary layer from the simulations is smaller than it should.

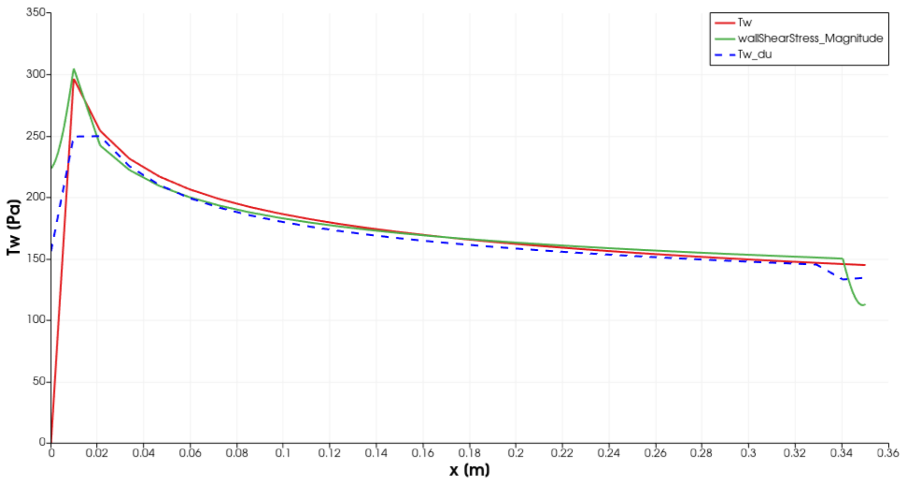

Figure 4.28: Shear stress over a flat plate, turbulent flow.

In this plot can be seen the supposition made for the transient case: the shear stress at the wall can be computed with the gradient velocity. In addition to this, the wall shear stress is completely adapted to the turbulent model. Finally, the 4.29 shows us that the results are good.

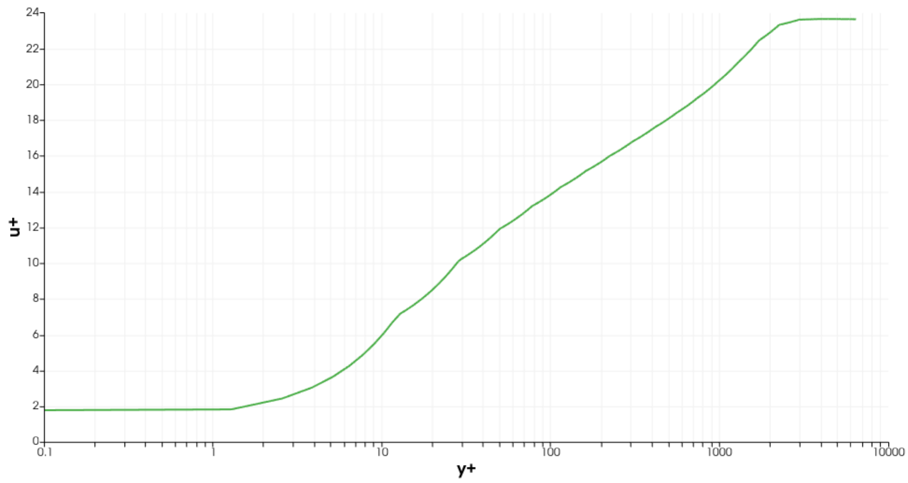

Figure 4.29: u^+ vs y^+ at the end of a flat plate, turbulent flow.

One way to validate the results obtained is to compare and replicate them with the ones from the known books as [4]. Particularly, it has been plot the laminar and turbulent dimensionless velocity profile, figure 4.30. The profiles should be in accordance with the same plot found at the book mentioned before.

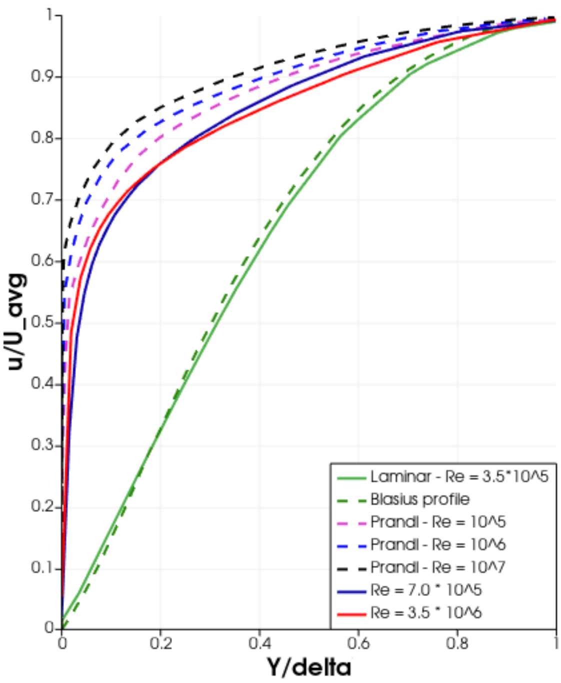

Figure 4.30: Comparison of dimensionless laminar and turbulent flat-plate velocity profiles at the end of the plate with the profiles from[4].

The dashed lines are the ones extract form the Prandl (the seventh root profile) and Blasius theory. As it can be seen, the laminar profile is clearly distinguished from the others and is perfectly adapted to the Blasius profile. The turbulent profiles are more mixed than they should and they are not , therefore, this confirms us again that the results are good but not extremely accurate as they should be. Nevertheless, the difference on the Reynolds number in our simulations is not as big as the ones in [4] (between the two turbulent simulations), and that is why it is a little bit more complicate to distinguish between the profiles. To work with a turbulent flow is always more complicated than with a laminar flow.

Finally, in the table 6 the average friction coefficients obtained from the simulation are presented. This table can be compared with the analytical results (table 5):

$Cf_{laminar}$	$2.255 \cdot 10^{-3}$
$Cf_{turbulent}$	$3.537 \cdot 10^{-3}$
$Cf_{transient}$	$4.176 \cdot 10^{-3}$

Table 6: Average friction coefficient from the simulation for inlet velocities of 1, 2 and 10 m/s, respectively.

As said before, the average friction coefficient is equal to the drag coefficient, so, with this values it can be obtained. The turbulent and the laminar average friction coefficients are very similar to the theoretical values. On the other hand, the for the simulation with a not negligible transient region (with v = 2m/s), the coefficient is distant from the theory. This is due to the fact that the simulation is perfect, as seen before and also because the flow behaviour in a transient region is not completely described or understood. For example, in [3], in order to consider the transient region on the average coefficient (equation 22), they integrate the local friction coefficient in two part, the turbulent and the laminar, considering the transient region as turbulent. That is a supposition acceptable but, also, approximated. This shows us the uncertainties of this region. With the C_d it can be compared with the theoretical values from the following plot:

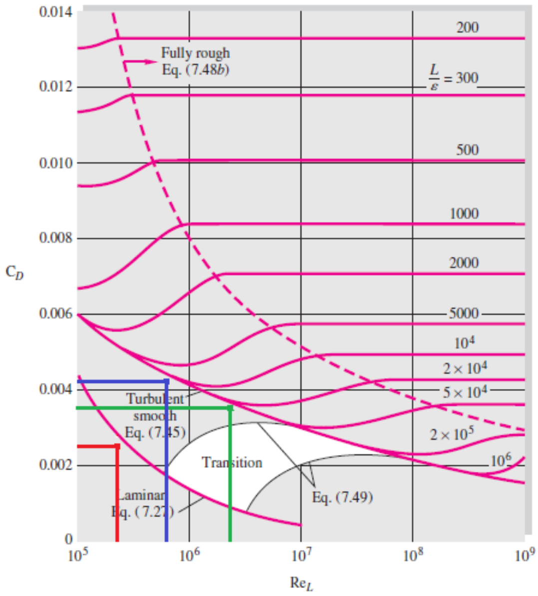

Figure 4.31: Drag coefficient of laminar and turbulent boundary layers on smooth and rough flat plates[4].

In this figure, the red color is the laminar case, the blue is the transient and the green is the fully turbulent case. As said before, the laminar and turbulent values are almost over the theoretical line, but the transient value should be at the transition region. Therefore, it can be concluded that the transient flow is not well resolved by a steady-state solver.

Before concluding this case study, it has to be pointed out that some simulation has been performed with a bigger domain, specially in the x direction, but the results were not good due to the fact that the transition between big cells to smaller was not good, as it has happened in the Pipe Flow (section 3.3). So the idea that SimScale struggles to deal with this situations gains weight.

5 Compressible Flow

In this section the intention was to perform an complete study of the compressible flow. In this case, it would be considered four cases. All of them were a a compressible flow through a nozzle, but with different situations. The first three would be an analysis of an over-expanded, under-expanded and ambient nozzle. The last simulation was going to be the flow but with a normal shock wave. The aim was to use different meshes to see how to capture best the effects of the flow in each situation.

Unfortunately, compressible simulations are much more complex and sensible than expected. It has been not possible to achieve some meaningful numerical results and, therefore, it has not done the previously mention study neither a good tutorial for the students. Nevertheless, this has been an opportunity to learn more about the solvers and understand the simulations. As a result of this, it has been decided to present the problems faced and the solutions found for some of them in order to help the students and possible future researches.

5.1 The Nozzle Compressible Flow: Problems and Solutions

The first thing to do is to introduce the case. A simple nozzle with a throat. The aim of the simulation is to observe the behaviour of the flow, and in this case the shear stresses play a very small roll. For this reason it was used a Laminar model and the time dependency was chosen the steady-state. Choosing a transient simulation would suppose a huge increment in the time and also is much more sensible to the *"Numerics"* of the solver and the Courant number.

Initially the boundary conditions were:

- Absolute Pressure inlet: $5 \cdot 10^5$ Pa
- Absolute Pressure outlet: 101325 Pa
- Temperature inlet: 1000 k
- Temperature outlet: 293 k
- Slip condition at the walls.

Before everything, it was done some analytical calculations. From the Saint Venant equation:

$$\frac{T}{T_0} = \left(1 + Ma^2 \frac{\gamma - 1}{2}\right)^{-1} \tag{23}$$

And the isentropic relations:

$$\frac{p}{p_0} = \left(\frac{\rho}{\rho_0}\right)^{\gamma} = \left(\frac{T}{T_0}\right)^{\frac{\gamma}{\gamma-1}} \tag{24}$$

With $\gamma = 1.4$. It is known that the Mach number at the outlet of the nozzle is:

$$Ma_{outlet} = 1.7$$

With this equations it can also be calculated the pressure, temperature and density at any point of the nozzle. Once the analytical calculations are done, it can be started the configuration of the simulation.

I was used the *"Hex-dominant parametric"* mesh algorithm to compute the mesh. The resultant mesh is presented in the following figure:

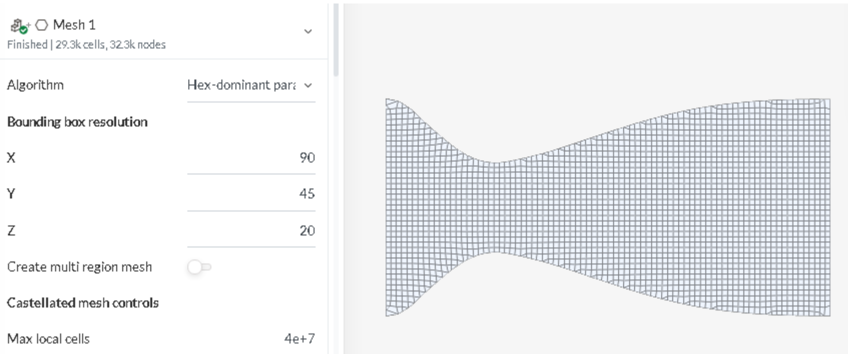

Figure 5.1: Homogeneous with hexahedral cells mesh.

Steady-State Simulations

After configuring all the simulation, the last step was to choose the time step. In this case, trying to keep the Courant number low but without increasing too much the computation time, the delta t chosen $\delta_t = 0.005$ s. The simulation did not finish. The following error appeared:

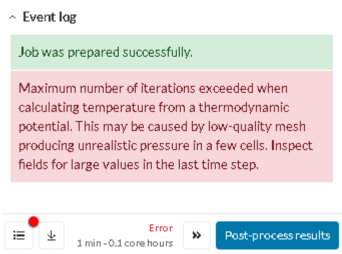

Figure 5.2: Error for the first compressible run.

Looking at the *"Solver log"* it could be seen the following line: *"time step continuity errors : sum local = 10297.457553, global = -2246.09704423, cumulative = -9038.7051981"*. This is a really big error. Any time than the continuity error rises more than 1%, the results should be analyzed very carefully. From the figure 5.2, the software is saying to check the mesh for possible illegal cells. The mesh looked fine and without irregular cells, but to ensure that this was not the problem, it was done a mesh refinement. Nevertheless, this was useful because in the next run, the same problem arise.

It seem that compressible simulations require extra care during the setup phase due to the high velocities. It is important to introduce the pressure differences progressively aiming to improve stability in early iterations [19]. So, the pressure was introduced gradually and the first results appeared. With this pressure introduction, two simulations were run, only changing the time step form 0.005 s to 1 s but the results were similar.

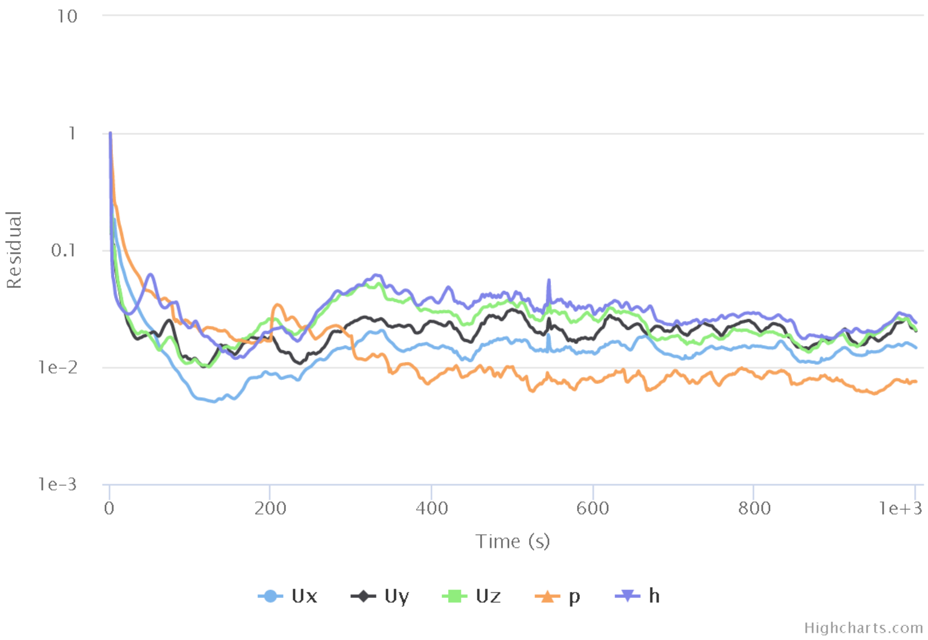

Figure 5.3: Residuals for a compressible run.

As it is possible to observe, the residuals are not bad, but they do not seem completely stabilized. With this plot, it is important to analyze carefully the numerical results, this time using SimScale's post-processor:

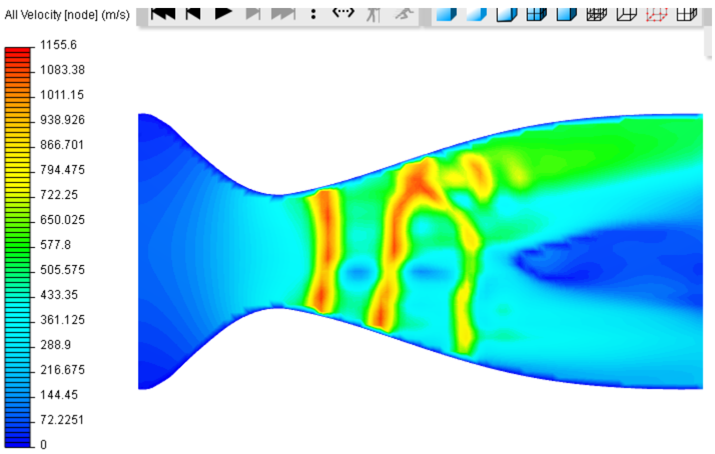

Figure 5.4: Nonphysical numerical compressible results.

As it can be observed the results are not good. The maximum velocity is 1155 m/s at certain points and at that places the pressure is negative. They do not agree with the analytical calculations and, furthermore, this is physically impossible. So, introducing the pressure gradually has help us to stabilize the simulation but not to get good results. As the instabilities started right after the nozzle, maybe a refinement in that region could help stabilize the simulation. Unfortunately, this new refinement did not change at all the results. The mesh did not seem to be the problem, then.

Previously, it had been available the solver *"rhoCentralFoam"*, a density-based solver. With this solver, it was easy to simulate shock waves and the convergence was much easier. With *"rhoSimpleFoam"* it is not possible to generate shock waves in any sufficiently accurate capacity. Maybe that is the source of the problem.

The boundary conditions are determinant in the simulations, so to avoid having a huge pressure difference that could give nonphysical results or induce shock waves, the inlet boundary condition was reduced to 1.5e5 Pa, keeping the temperature.

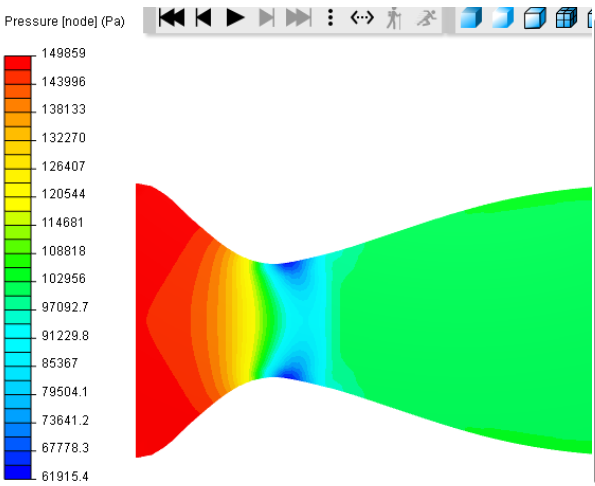

Figure 5.5: Pressure distribution, $P_{inlet} = 1.5e5$ Pa.

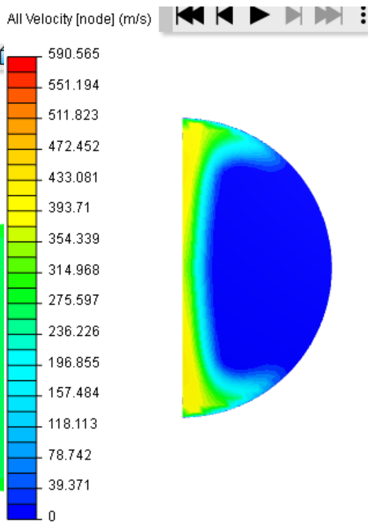

Figure 5.6: Velocity distribution, $P_{inlet} = 1.5e5$ Pa.

In this simulation the residuals were good and stabilized and the pressure seemed also well. Nevertheless, in the figure 5.6, it can be observed that the flow is not homogeneously distributed on the radial direction and it should be. The turbulence model was also changed to *"K-omega SST"* to see if it affected but in the end it is been used the same solver, *"rhoSimpleFoam"*, so the results are more or less similar.

At this point, it had been changed the boundary conditions, time-step and turbulence model but the results still did not show up. After some investigations, it has been seen that the solver used until now, (*"rhoSimpleFoam"*), does not work for transonic steady-flows. It might have to do with the fact that *"rhoSimpleFoam"* is an uncoupled solver and this means that at transonic to supersonic flow, the solution will be unable to converge regardless of numerical dampening [20]. In this website, is also mentioned a paper ([21]) were convergence was achieved with this solver, but is an unknown how. In this situation, the best thing to do was change to another solver. As said previously, ideally, it should be used *"rhoCentralFoam"* but is no longer available. For this reason, the only alternative is *"rhoPimpleFoam"*, which is the solver used in transient time dependency simulations.

Transient Simulations

The odyssey to obtain convergence and physically reasonable numerical results with a transient time dependency simulations starts here. Firstly, to avoid having troubles with the mesh and it has been changed the solid nozzle to one with less ”abrupt” changes from the inlet to the throat. In this case, is has been selected the nozzle described in the chapter related to compressible CFD in [3]. With this geometry, it could be compared the possible results with the ones in the book.

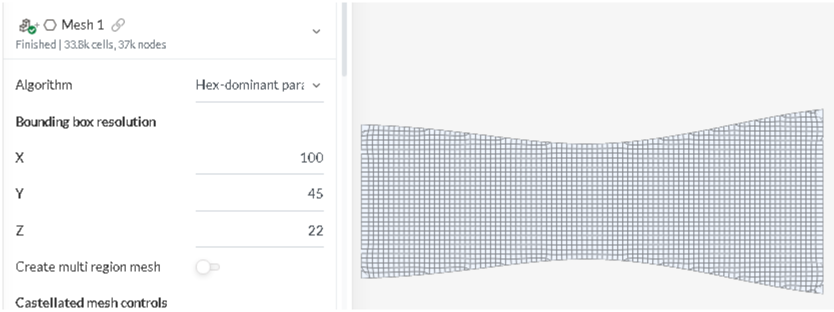

Figure 5.7: Mesh computed for the new geometry.

However, some tests were done with the previous solid, but the results were not good.

In the transient flow, the CFL condition[3] takes special importance and it is a must to have it under 1 not only to have numerical stability but also good results. The Courant number for a velocity of 340 m/s and $\delta_x = 6.42 \cdot 10^{-3}$ m is:

$$C = 1.88e - 5$$

Which was reduced until 1e-5 during the simulations.

The first simulation was with the original boundary conditions (but with $T_{inlet} = 300k$) and without ramping the pressure. In addition to this, to help the solver to converge, in *”Numerics”*, the *”Number of non-orthogonal correctors”* was changed from 0 to 2. The results obtained are presented in the figures 5.8 and 5.9.

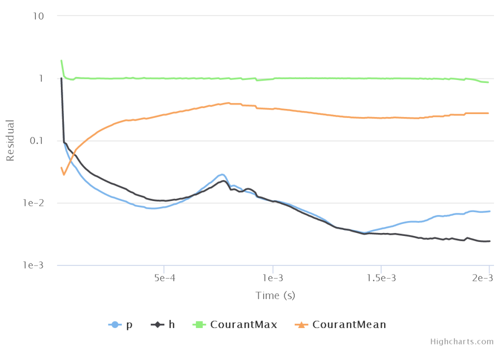

Figure 5.8: Residuals, $P_{inlet} = 5e5$ Pa.

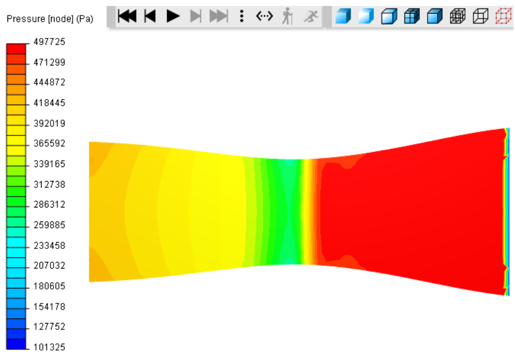

Figure 5.9: Pressure distribution.

[3]For more information look at the Annex.

The results obtained are very interesting. The fist thing to notice is that the simulation converged without ramping the inlet pressure as opposite to the steady-state case. It seem that appears a shock wave right after the nozzle that is perfectly possible. The strange thing here is at the outlet. In order to fulfill the BC, the pressure drops form the maximum value to the lowest in a few cells. That does not seem right. Also the pressure at the inlet is not the one set as BC.

The second meaningful simulation done was the same as before but this time with $T_{inlet} = 1000k$ and the $\delta_t = 1.5 \cdot 10^{-5}$ s. The simulation time was also increased. The residuals obtained were similar to the previous ones. The results are showed at the figures 5.10 and 5.11.

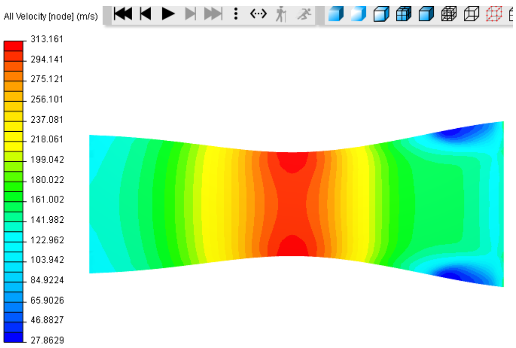

Figure 5.10: Velocity distribution, $P_{inlet} = 5e5$ Pa and $T_{inlet} = 1000$ k.

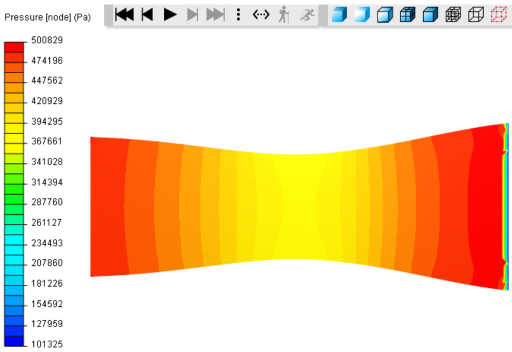

Figure 5.11: Pressure distribution.

Although it was changed just the temperature as BC, the results are much different than before. Despite the fact that in the previous simulation, the Mach at the throat was one, this time is lower. The flow behaves as subsonic when, theoretically, it should be supersonic. Why this has happened is an unknown. As before, it is present the strange pressure reduction at the outlet.

The next simulation was with the same BC as in [3]. With $P_{inlet} = 2.1e5$ Pa, $T_{inlet} = 300$ k and $P_{outlet} = 5e4$ Pa. Unfortunately, the same happened. The flow never reached the supersonic state and there was the same pressure drop as in the figure 5.9 and 5.11. The residuals are the following ones:

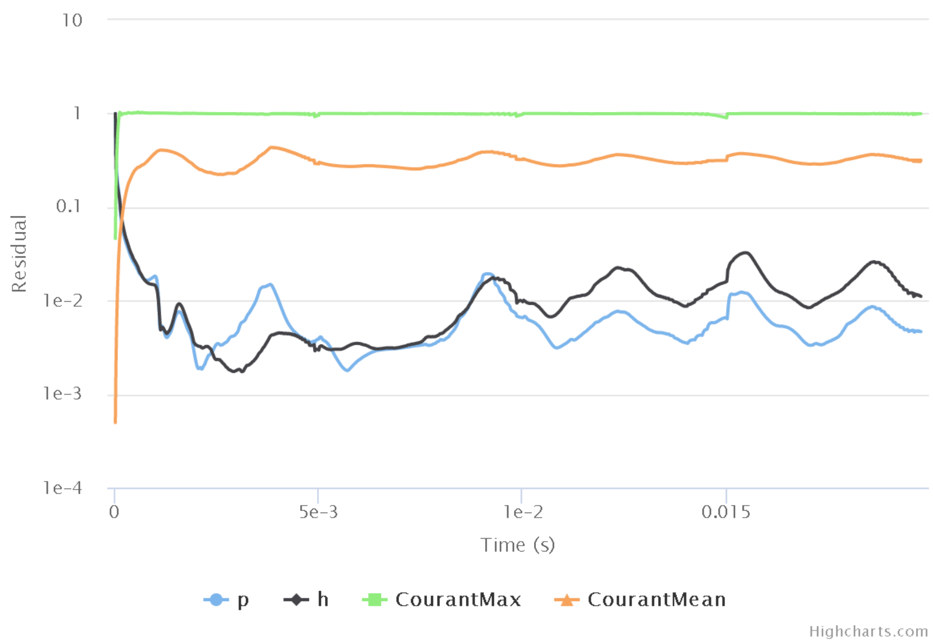

Figure 5.12: Residuals with a ramped pressure inlet, $P_{inlet} = 2.1e5$ Pa and $P_{outlet} = 5e4$ Pa.

The only things left to do was to try to under-relax the simulation in order to help to converge and reduce the residuals. In a transient simulation, it is wanted to ensure convergence of the solution at each step. All the equations that the solver calculates (mass, momentum, etc) need to converge for every time-step. This supposes that the residuals have to reach a small value every time.

Firstly, it has been **under-relaxed** the simulation. Sometimes, there are some cells or regions that causes numerical fluctuations. This is why there are relaxation factors. They help to the simulation to converge, in the case of under-relaxation, or to converge faster (over-relaxation). In this case, it is wanted convergence stability even if it is sacrificed some computation time. Under-relaxation is a simple yet effective technique for updating the fields between iterations. The default way to update the field values in a new iteration would be to simply ignore the old value and replace it with the new values [22]. If the next value is worse than the previous one (because the simulation diverges), maybe, instead of replacing the field values with the new result, it gets updated with a weighted average between the old and the new value. The selected values for the relaxation factors were 0.1 for the pressure field and 0.5 for the velocity equation. The simulation had the original BC and a ramped inlet pressure. However, the simulation diverged. The residuals are presented in the figure 5.13.

Secondly, the **relative tolerance** of the residual controls was reduced to 0.005. If the residual improvement between two consecutive sub-iterations is lower than this value, it is assumed convergence between the step. The simulation was done with the same configuration as the last one (even with the relaxation factors reduced). This simulation also diverged (figure 5.14).

Even if the simulation did not diverged in the end, the residuals looked like in the

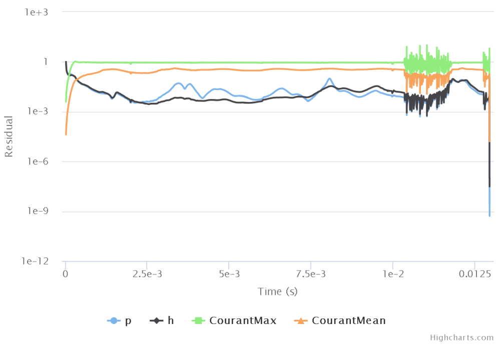

Figure 5.13: Residuals for the under-relaxed simulation.

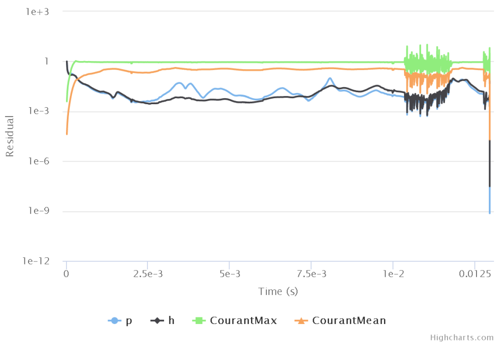

Figure 5.14: Residuals for the under-relaxed simulation and the relative tolerance reduced.

figure 5.12, where the results were not good. Nevertheless, is strange because with this pressure difference but without introducing it gradually, the simulations converged.

After all the simulation done, it can be concluded that the compressible flow is much more sensible than the incompressible flow. The *"rhoSimpleFoam"* cannot compute properly transonic flows or shock waves but, it has been seen that either the *"rhoPimpleFoam"*. What is more, the flow behaves as subsonic almost always and for huge pressure ratios, the solution diverges. Therefore, after SimScale withdrew the density-based solver that was perfect for this situations, is no longer possible to obtain real solutions, at least for the general user.

6 Environmental Impact

This thesis it has been done integrally by computer. Basically, it is the application and research of the fluid's dynamic classic theory with SimScale. There has been no need to do displacements or any other activity outside the laptop. We have tried to keep this investigation as ecological as possible. The communication between the director and the pupil were done by email, with a two exceptions, minimizing at the maximum the CO_2 emissions. In the lasts recent years, telecommuting and online work has increased exponentially and this thesis has followed the tendency in order to be adapted and eco-friendly.

6.1 Environmental Impact from the Study and the Document Synthesis

During the hole project development and synthesis it has been used bu the student the same laptop (with and average consumption of 220 W) and a router (with an average consumption of 12 W). To be more precise, the consumption done by the director is considered 10% of the work done by the student, and this is also computed in the calculations. With the energy consumption we can approximate the CO_2 emissions with a relation from the Spanish Government document; *"Factores de emision de CO_2 y coeficientes de paso a energia primaria"*, from 2016.

	Hours	Energetic consumption [kWh]	CO_2 Emissions [kg CO_2]
Laptop	300	66	21.846
Router	300	3.6	1.197
Director	30	6.96	2.304
Total			**25.35**

Table 7: CO_2 emissions from the energetic consumption.

It should added to this table the energy consumption done by the SimScale servers, unfortunately, that is and unknown for us.

7 Budget

The costs of the thesis has been divided between direct costs and indirect costs. To divide the costs in variable or fixed cost is useless, in this case, due to the fact that all of them would be variable, depending in the number of working hours. Other kind of costs are also discarted.

Direct Costs

As this project has been done completely with the computer, the only expenses has been relative to the personal and the electricity used by the computer.

Concept	Total [€]
Personal	5250
Energy*	5.818
Total	**5256**

Table 8: Direct costs.

*The energy cost is calculated with the mean electricity charges in Spain.

The main expenses has been the Personal costs.

Indirect Costs

The indirect costs are the expenses related to extra electrical expenses, diets, taxes or any other possible factors that can not be computed. All these, are estimated as a 10% of the direct cost. Therefore, the total indirect costs are:

Concept	Total [€]
10 % Direct cost	**525**

Table 9: Indirect costs.

Personal

The working hour distribution are presented in the following table:

Concept	Personal	€/hours	Hours	Total cost [€]
References and state of the art	Researcher	15	10	150
Introduction to SimScale	Researcher	15	30	450
P.F.C.* Research	Researcher	15	25	375
P.F.C. Simulation	Researcher	15	28	420
P.F.C. Post-Process	Researcher	15	12	180
F.P.C** Research	Researcher	15	17	255
F.P.C. Simulation	Researcher	15	30	450
F.P.C. Post-Process	Researcher	15	23	345
C.F.*** Research	Researcher	15	46	690
C.F. Simulation	Researcher	15	19	285
C.F. Post-Process	Researcher	15	25	375
Project Synthesis	Researcher	15	35	525
	Director	25	30	750
Total			330	**5250**

Table 10: Personal working hours breakdown.

With:
*P.F.C: The Pipe Flow Case Study
**F.P.C: The Flat Plate Flow Case Study
***C.F: The Compressible Flow

8 Planing and Scheduling

In the project charter it was different than the final planing. The initial problem was the Pipe Flow, were it was very complicated to find the find an optimum configuration to obtain good results. The were problems to find the right solid configuration, also in the mesh computation and, finally in the parameters for the simulation. This extra problem supposed a delay in the next case and additional ours of work. For the Flat Plate Case happened the same. Finding the right domain and mesh configuration was more complicated than expected and also in the post-processing, learning to use ParaView increased the working hours. The total increment in hours was more than 10. Finally, in the Compressible Flow there has been the problems described in the section 5 that supposed a lot of hours. The addition of the three delays and external inputs made impossible to spent the time necessary to do the final case proposed in the project charter: The Vehicle Flow. The weekly hour resume is the following:

		Hours
Familiarization with SimScale	Week 1 (17/02-23/02)	10
	Week 2 (24/02-01/03)	12
	Week 3 (02/03-08/03)	8
InternalFlow	Week 4 (09/03-15/03)	14
	Week 5 (16/03-22/03)	19
	Week 6 (23/03-29/03)	20
	Week 7 (30/03-05/04)	12
External Flow	Week 8 (06/04-12/04)	16
	Week 9 (13/04-19/04)	8
	Week 10 (20/04-26/04)	16
	Week 11 (27/04-03/05)	3
	Week 12 (04/05-10/05)	4
	Week 13 (11/05-27/05)	8
	Week 14 (18/05-24/05)	18
Compressible	Week 15 (25/05-31/05)	4
	Week 16 (01/06-07/06)	18
	Week 17 (08/06-14/06)	36
	Week 18 (15/06-21/06)	32
Project Synthesis	Week 19 (22/06-28/06)	21
	Week 20 (29/06-05/07)	9
Oral Presentation	Week 21 (06/07-12/07)	5
	Week 22 (13/07-19/07)	5
TOTAL		**300**

Table 11: Hours week breakdown.

This table says that most of the time the weeks were invested in the different concepts but it is not completely strict. This means that the project synthesis has been also taking ours from week 2 or others, for example.
With all the changes done during the development of the thesis, the final Gannt chart is

presented:

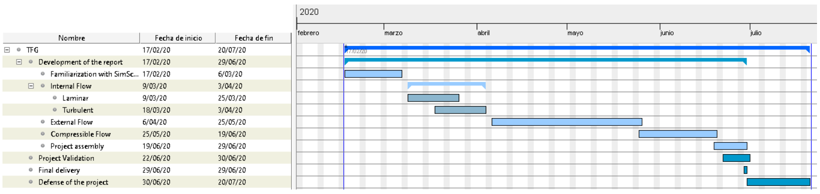

Figure 8.1: Final Gannt Chart.

The External Flow Case has lasted more than expected for three reasons: the first one is that obtaining good results has been more arduous than expected. The second one is that this moths has overlapped with the exams and deliveries of the university. Finally, the COVID-19 pandemic made the situation more complex. It was harder to achieve the academic targets from home and combine it with our day-to-day life.

9 Conclusions and Further Investigation

SimScale is an intended easy-to-learn platform and, after many hours working with this software, it can be said that it really is compared to other products in the market. Nevertheless, finding the optimum and real results for every case is still very complicated. It has been difficult to have a good introduction for the students and also study the meshing in this software. These extra difficulties found in the thesis has avoid us to finish a task of the planned work. Taking all this aside, it can be still be extracted some conclusions.

For the **Internal Flow** it has been verified that without the right refinement in the mesh, the viscous effects cannot be computed. Without a good refinement, there is not a good velocity profile and, therefore, the viscous effects are wrong. However, this refinement could cause perturbations in the pressure and velocity, leading in to worst results. This phenomena has been observed also in the External Flow. SimScale does not have the ease to deal with the transitions from big cells to smaller. There is, obviously, to improve this regions were the bad cells are generated, but, if the aim is to introduce the student to apparently simple problems, this is an issue. Another conclusion is that the ”Wall functions” available to use when the mesh is homogeneous or with a $y^+ > 5$, may seem that returns a good velocity profile, but, usually, this one is erroneous and does not gives the wright shear stress values.

For the **External Flow** it has been also tough to obtain good results for this simple case. The laminar results had been almost identical to the theory as in the Pipe Flow, but for the turbulent case has been different. It has been necessary to re-mesh and refine to have better results, so the boundary conditions influence the mesh procedure, and still, the results had not been perfect. The maximum deviation form the theory has been with a not negligible transient region. This unknown regions have interacted more than it had been thought. The wall shear stresses from the simulation were good in the turbulent case and, as in the Internal Flow, they can be computed with the velocity profile. This case has also help us to see some particularities of ParaView. For example, when divides by 0, the result, instead of been ∞, is 0. This has to be taken into account, because it changes substantially the distribution of a certain variable.

Finally, for the **Compressible Flow** has been very useful to learn more about the solvers. The main conclusion to be extracted is that the transonic flow are not supported by the solvers available at SimScale and, therefore, is better to use another platform for this kind of flow. The *rhoPimpleFoam* solver has resulted useful and it has not responded properly for a BC change either for under-relaxation or a residual tolerance reduction. The *rhoSimpleFoam* solver cannot resolve high velocity situations that may generate shock waves. Also, it is very sensible and requires the introduction of this high pressure gradually. The figure 5.4 shows this instabilities. In addition to this, for low pressure differences the flow velocity is not well distributed along the nozzle. Both solvers has not improved the results for different meshes or geometries.

In the end, this is a very interesting thesis for the students to be introduced into computational fluid dynamics, by, in the first and second chapters, apply the theory, learn how to use a strong commercial software, compute a good mesh for each situation and, then, interpret the result with ParaView. In the last chapter, the student is immersed into the transient simulation, the solvers, CFL conditions and, therefore, simulation control.

9.1 Target - Achievement Synthesis

In the following section is showed a visual summary of the targets and scopes of the project and his respective achievement level (A.L.).

- Target 1 → Internal Flow Meshing Analysis — A.L. - 100 %

  It was been seen that the viscous effects can only be computed with a very refined mesh and it this can be done with SimScale. Nevertheless, high refinement could cause a bad transition between big and smaller cells which can lead to perturbations.

- Target 2 → Pipe Flow Tutorial — A.L. - 100 %

  It has been delivered a complete tutorial, perfect for the student to apply the theory and learn about CFD.

- Target 3 → Internal Flow Meshing Analysis — A.L. - 100 %

  This analysis has helped us to see that the mesh also depends on the boundary conditions and, even with a very refined mesh, the results are tough to obtain. In addition to this, it has been seen that a steady-state solver has difficulties to compute transient regions. Finally, it has been observed that ParaView is an excellent post-processor but also requires knowledge to use it and interpret the results.

- Target 4 → Flat Plate Flow Tutorial — A.L. - 100 %

  It has been delivered a complete tutorial, perfect for the student to apply the external theory from the lessons and learn about SimScale and ParaView .

- Target 5 → Compressible Flow Meshing Analysis — A.L. - 50 %

  The initial tutorial has not been done, were the initial idea was to, also, perform an investigation in the meshing methods. Nevertheless, during this section, different meshes has been tested and compared.

- Target 6 → Compressible Flow Errors Compilation — A.L. - 100 %

  The difficulties found in the previous point has help us to understand deeply this kind of flow and even more the solvers available for SimScale.

Annex

A Wall Functions

In this part of the Annex, a small introduction to wall functions will be illustrated and his principles.

A.1 What are wall functions

As said during the Thesis, wall functions are empirical equations used to satisfy the physics of the flow in the near wall region. To know in which conditions wall functions approach can be used, it has to be remembered some theory from [3]. The boundary layer is divided in three parts:

1. **The wall layer** (with $y^+ < 5$)
   In this layer the fluid is dominated by the viscous effects. The velocity profile is given by

   $$u^+ = y^+$$

   At the figure 3.2 it can be seen a graphical representation.

2. **The Overlap layer** $(30 < y^+ < 200)$
   In this layer, the turbulence and viscous shear stress are present and dimensional analysis indicates that the velocity in the overlap layer is proportional to the logarithm of distance, and the velocity profile can be expressed as:

   $$u^+ = 2.5 \ln y^+ + 5.0$$

3. **The Outer layer**
   The turbulent shear dominates. The normalized velocity profile in the core region of turbulent flow in a pipe depends on the distance from the center-line and is independent of the viscosity of the fluid.

There is also a fourth layer called the **Buffer layer**, between the Wall layer and the Overlap layer. Here viscous and turbulent stresses are of similar magnitude.

Some turbulence models such as k-ε are only valid in the area of turbulence fully developed, and do not perform well in the area close to the wall. In order to deal with the near wall region, two ways are usually proposed [23]:

- One way is to **integrate the turbulence to the wall**. Turbulence models are modified to enable the viscosity-affected region to be resolved with all the mesh down to the wall, including the Viscous sublayer. This approach is used when we are interested in the forces of the wall and with high Reynolds numbers (figure A.2).

- Another way is to use the so-called **wall functions**, which can model the near wall region (figure A.1). Normaly, they are used not that interested in the forces of the wall and with high Reynolds numbers.

Using this wall functions, helps us to reduce computation time since it is not needed a very fine mesh near the wall to capture the viscous effects. So this functions can be applied in the turbulence model, which is actually the describing function of the boundary layer (logarithmic). Nevertheless, to ensure the accuracy of the result, the fist cell center needs to be placed in the log-law regions. If the cell center lies in the Viscous sub-layer, the results from this approach are very inaccurate. Something similar happens in the Buffer layer, since it is complex velocity profile is not well defined and the original wall functions avoid the first cell center located in this region [24]. Here is a visual difference between using wall functions or not:

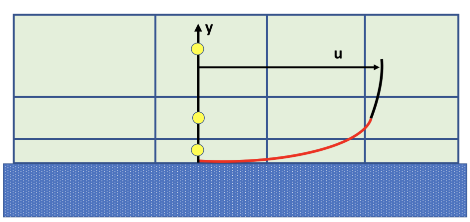

Figure A.1: Wall function approach [23]

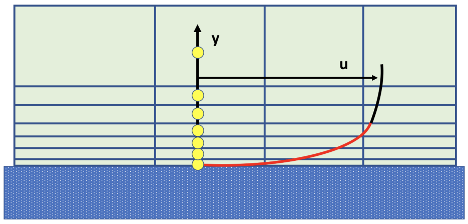

Figure A.2: Full resolution approach.

B Solvers and CFL Condition

B.1 CFL Condition

In many books has been explained detailed the physical meaning of this condition ([6]), so here it will be developed a relatively simplified explanation of this condition.

The convergence condition by Courant–Friedrichs–Lewy (CFL) can be interpreted simply as one of the basic rules that should be satisfied for convergence while solving certain partial differential equations numerically. This condition expresses that the distance that any information travels during the time-step length within the mesh must be lower than the distance between mesh cells. The Courant number is:

$$CFL = \frac{u \cdot \Delta t}{\Delta x}$$

Therefore, to achieve numerical stability the velocity times the time step ($u \cdot \Delta t$) must be smaller than the size of the cell (Δx), and, then:

$$CFL \leq 1$$

To lower this value it is possible to refine the mesh or reduce the time-step. Note that the CFL condition is necessary, but not sufficient, condition for the stability of a numerical scheme. The reality is that this is more complex than this and there is an extensive book about this issue [25].Nevertheless, this theory is going to be hugely simplified in order to introduce the learner to it.

The coupled set of governing equations is discretized in time for both steady and transient calculations, so the CFL number is used to compute the time step in both cases. This temporal discretization is accomplished by either implicit or explicit algorithms. From [6]:

"An explicit numerical method is one in which the dependent variables are computed directly via already known values. In this case any discretization operator can be directly evaluated based on the actual variable values. On the other hand, a numerical method is said to be implicit when the dependent variables are treated as unknowns and assembled to form a coupled set of equations which are then solved via special numerical tools using either a direct or an iterative solution algorithm."

In the explicit formulation the time step is computed from the CFL condition. Nevertheless, with implicit formulation is the Courant number can be higher than one. To see why this is true consider an ODE system with backward Euler (or implicit Euler method):

$$\vec{U}^{n+1} = \vec{U}^n + \Delta t A \vec{U}^{n+1} \Rightarrow (I - \Delta t A)\vec{U}^{n+1} = \vec{U}^n$$

Solving for the solution at t^{n+1} requires knowledge of all values at t^n, meaning that no matter what time step it is used we are pulling in the entire physical domain of dependence[26].

SimScale uses implicit schemes, so is possible to run at high Courant number. Though stability does not imply accuracy, that depends on the time step of simulation and the variation over time expected in your flow. This affects specially the transient simulations, with larger time steps it can be missed some transient features or other fluctuations.

B.2 Simscale Solvers

It is necessary to differentiate between the solver and the turbulence model. The models supported nowadays are the Reynolds-averaged Navier-Stokes (RANS) and the Large eddy simulation (LES). The solvers available according to the chosen turbulence model and time-dependency are presented in the tables 12 and 13, for incompressible and compressible flows, respectively.

In order to understand the solvers, it is presented an simple explanation extracted form [27]:

In a **steady state SIMPLE** (Semi-Implicit Method for Pressure-Linked Equations) loop, velocities and pressures are reduced every iteration until all parameters reached values small enough for convergence. The reduction of these values per iteration may or may not be a very smaller number. As long as the overall reduction meets convergence requirement, the simulation is over.

In a **time dependent SIMPLE loop**, all the parameters (velocities and pressure etc) are required to reduce to values small enough for convergence in each time step. It is easier to view transient SIMPLE as a lot of steady state SIMPLE loops over multiple time steps. Now, because within each time step, there are multiple steady state SIMPLE loops, it is called *'iterative time advancement'*.

The difference between **PISO** (Pressure Implicit with Splitting of Operator) and SIMPLE in terms of time advancement is that PISO does not require an iterative process, and is therefore called *'non-iterative time advancement'*. Here by iterative I mean multiple PISO loops each time step (just like transient SIMPLE). However, for a solution to converge when solving with PISO, it would be needed to do iterations within the PISO loop itself. Note that there is a difference between iterations within each PISO loop, and in between PISO loops. For PISO to yield promising results, it is generally required to go through two pressure corrector loops (each of these two loops will then go through multiple iterations to reduce residuals) and one loop for each velocity component with multiple iterations.

Finally, the **PIMPLE algorithm** is a combination of PISO and SIMPLE, and is very similar to transient SIMPLE with PISO replacing steady state SIMPLE. The PISO loops are repeated until the convergence requirement is reached. All these algorithms are iterative solvers but PISO and PIMPLE are both used for transient cases whereas SIMPLE is used for steady-state cases.

In conclusion, transient SIMPLE is a lot of steady-state SIMPLE loops per time step; PIMPLE is a lot of PISO loops per time step. For transient SIMPLE, solutions converge when the initial residuals at the start of each steady-state SIMPLE loop fall below predefined values; for PIMPLE to converge, the initial residuals at the start of each PISO loop will fall below pre-defined values.

Turbulence Mode	Time dependency	Solver	OpenFOAM solver
Laminar	Transient	PIMPLE	pimpleFoam
		PISO	pisoFoam
		ICO	icoFoam
	Steady-state	SIMPLE	simpleFoam
		ICO	icoFoam
RANS	Transient	PIMPLE	pimpleFoam
		PISO	pisoFoam
	Steady-state	SIMPLE	simpleFoam
LES	Transient	PIMPLE	pimpleFoam
	Transient	PISO	pisoFoam

Table 12: SimScale's available solvers for incompressible flow. [28]

Turbulence Mode	Time dependency	Solver	OpenFOAM solver
Laminar	Transient	pressure-based	rhoPimpleFoam
	Transient	density-based	rhoCentralFoam[4]
	Steady-state	-	rhoSimpleFoam
RANS	Transient	pressure-based	rhoPimpleFoam
	Steady-state	-	rhoSimpleFoam
LES	Transient	pressure-based	rhoPimpleFoam

Table 13: SimScale's available solvers for compressible flow. [29]

References

Books

3. ÇENGEL YUNUS A.; CIMBALA JOHN M. *Fluid Mechanics: Fundamental applications*. ISBN 0–07–247236–7.
4. FRANK M.WHITE. *Fluid Mechanics*. Seventh. ISBN 978-0-07-352934-9.
6. MOUKALLED, F; MANGANI, L; DARWISH, M. *The Finite Volume Method in Computational Fluid Dynamics. An Advanced Introduction with OpenFOAM and Matlab.* 2016. ISBN 978-3-319-16873-9. Available from DOI: 10.1007/978-3-319-16874-6.
25. MOURA, Carlos A. de.; KUBRUSLY, Carlos S. *The Courant-Friedrichs-Lewy (CFL) condition : 80 years after its discovery*. ISBN 9780817683931.

Thesis

8. JOSÉ M AMOR. *Turbulent boundary layer: comparison between a flat plate and a rotating disk with and without periodic roughness*. 2015. PhD thesis. CHALMERS UNIVERSITY OF TECHNOLOGY.

Reports

2. LÓPEZ, Mariano; MANCUSO, Francisco; AGUIRRE, Ricardo M.; HENDERSON, Germán R.; COUSSIRAT, Miguel G. *Uso de la mecánica de fluidos computacional como apoyo a la enseñanza de la mecánica de los fluidos en la universidad.* 2016. Available also from: https://www.researchgate.net/publication/310234143. Technical report.
5. JACKSON, Andrew; COMBEST, Daniel P. *SnappyHexMesh. Theory and Application*. Available also from: www.engys.com. Technical report.
9. MENTER, Florian R. *Improved Two-Equation k-Turbulence Models for Aerodynamic Flows*. Available also from: https://ntrs.nasa.gov/search.jsp?R=19930013620. Technical report.
21. FÜRST, Jiří. *On the implicit density based OpenFOAM solver for turbulent compressible flows*. Technical report.
24. FANGQING LIU. *A Thorough Description Of How Wall Functions Are Implemented In OpenFOAM*. 2017. Available also from: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016. Technical report. Chalmers university of techology.

Websites

1. *SimScale*. Available also from: https://www.simscale.com/.

7. *Turbulent Pipe Flow — SimScale Documentation*. Available also from: https://www.simscale.com/docs/content/validation/TurbulentPipeFlow/TurbulentPipeFlow.html.

10. *2D mesh generation for CFD analysis of a wing profile - Using SimScale / Fluid Flow / CFD - SimScale CAE Forum*. Available also from: https://www.simscale.com/forum/t/2d-mesh-generation-for-cfd-analysis-of-a-wing-profile/61702.

11. *How to create 2D mesh using SimScale tool? - Using SimScale / CAD & Meshing - SimScale CAE Forum*. Available also from: https://www.simscale.com/forum/t/how-to-create-2d-mesh-using-simscale-tool/3647.

12. *Symmetry boundary condition — SimScale Documentation*. Available also from: https://www.simscale.com/docs/content/simulation/model/boundaryConditionTypes/OF_new_symmetry.html.

13. *ParaView*. Available also from: https://www.paraview.org/.

14. *Main Settings for Hex-dominant parametric — SimScale Documentation*. Available also from: https://www.simscale.com/docs/content/preprocessing/meshing/snappyHexMesh/SHM_MainSettings.html.

15. *numpy.gradient — NumPy v1.17 Manual*. Available also from: https://docs.scipy.org/doc/numpy/reference/generated/numpy.gradient.html.

16. *Y+ Calculator - Compute Wall Spacing for CFD*. Available also from: https://www.pointwise.com/yplus/.

17. *meshQualityControls*. Available also from: https://www.cfdsupport.com/OpenFOAM-Training-by-CFD-Support/node129.html.

18. *Convergence, solver selection questions - Using SimScale / Fluid Flow / CFD - SimScale CAE Forum*. Available also from: https://www.simscale.com/forum/t/convergence-solver-selection-questions/274.

19. *Wing Simulation: Compressible Flow Around a Wing — SimScale Tutorial*. Available also from: https://www.simscale.com/docs/tutorials/tutorial-compressible-flow-simulation-around-a-wing/.

20. *Errors in low-pressure compressible flow (Hyperloop) - Project Support - SimScale CAE Forum*. Available also from: https://www.simscale.com/forum/t/errors-in-low-pressure-compressible-flow-hyperloop/81813/17.

22. *Relaxation factors_Multiphase - Using SimScale - SimScale CAE Forum*. Available also from: https://www.simscale.com/forum/t/relaxation-factors-multiphase/82678/4.

23. *What is y+ (yplus)? - SimWiki - SimScale CAE Forum*. Available also from: https://www.simscale.com/forum/t/what-is-y-yplus/82394.

26. *pde - Understanding the Courant–Friedrichs–Lewy condition - Computational Science Stack Exchange*. Available also from: `https://scicomp.stackexchange.com/questions/25398/understanding-the-courant-friedrichs-lewy-condition`.
27. *PIMPLE pressure residual - Using SimScale / Fluid Flow / CFD - SimScale CAE Forum*. Available also from: `https://www.simscale.com/forum/t/pimple-pressure-residual/443/5`.
28. *Incompressible Fluid Flow Analysis I SimScale Documentation*. Available also from: `https://www.simscale.com/docs/analysis-types/incompressible-fluid-flow-analysis/`.
29. *Compressible Fluid Flow Analysis I SimScale Documentation*. Available also from: `https://www.simscale.com/docs/analysis-types/compressible-fluid-flow-analysis/`.