Fluid Mechanics MTF053

Numerical Simulation of Boundary Layer Flows

Computer Assignment 2

Division of Fluid Dynamics
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Contents

1 Document Archive

1.1 Related course material

You can find detailed descriptions of the physical phenomena studied in this exercise and associated formulae in the course book *F. M. White Fluid Mechanics 9^{th} ed.* (White, 2016) and in the following documents:

1. `MTF053_C06.pdf` (lecture notes chapter 6)
2. `MTF053_C07.pdf` (lecture notes chapter 7)
3. `MTF053_Equation-for-Boundary-Layer-Flows.pdf`
4. `MTF053_Turbulence.pdf`
5. `MTF053_Formulas-Tables-and-Graphs.pdf`

2 Assignment instructions

1. Read through the introduction and theory sections in this document
2. Follow the detailed guide provided in this document to do two `Star-CCM`$^+$ simulations.

   simulation 1: flat plate boundary-layer flow
   download the following file `MTF053_CA2_Task-1.sim` (a `Star-CCM`$^+$ case definition file that initiates the solver for simulation 1)

   simulation 2: flow around a cylinder
   download the following file: `setupPost.java` (a `Star-CCM`$^+$ `Java` macro that sets up the postprocessing for simulation 2)

3. Download the following file: `MTF053_CA2.ipynb` (a `Jupyter NoteBook` that contains the `python` code needed to do the assignment)
4. Start `Jupyter Lab`: `https://jupyter.org/try-jupyter/lab/`
5. Upload the `Jupyter NoteBook` (`MTF053_CA2.ipynb`) in `Jupyter Lab`
6. Go through the assignment step by step in `Jupyter Lab`. Write down answers to questions in the `Jupyter NoteBook`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.
7. When done with the assignment, download your modified `Jupyter NoteBook` from `Jupyter Lab` and submit it in Canvas along with any images/photos that you would like to add to the document. To download the `Jupyter NoteBook` right click on the notebook file in the file tree on the left side of the `Jupyter Lab` user interface and click on download in the drop down menu.

3 Introduction

In this assignment you will use a commercial Computational Fluid Dynamics (CFD) software called `Star-CCM`$^+$. Two different simulations will be done. The first involves flow over a flat plate and you will extract data from the CFD simulation to compare with the analytical/empirical formulations for laminar and turbulent boundary layers. In the second simulation you will simulate the flow over a cylinder and compare the simulated flow field with data from an experiment (provided data).

4 Running `Star-CCM`$^+$ remote

You will run `Star-CCM`$^+$ in Linux but please note that you don't have to sit at a Linux computer to do that. It is possible to access the Chalmers Linux system remotely and thus you can run `Star-CCM`$^+$ from your own computer or you can do the simulations from a Chalmers non-Linux workstation. To set up remote access to the Chalmers Linux systems on your own computer, please use the guides linked below:

Set up remote access in `Windows`

Set up remote access in `MacOS`

Set up remote access in `Linux`

If you are asked to chose which graphics to use, please select `MATE`.

Once you are connected to the Linux machine remotely, open a terminal (konsole) by proceeding as shown in the Figure 4.1. From the terminal you will be able to start `Star-CCM`$^+$ by typing the command `starccm+`

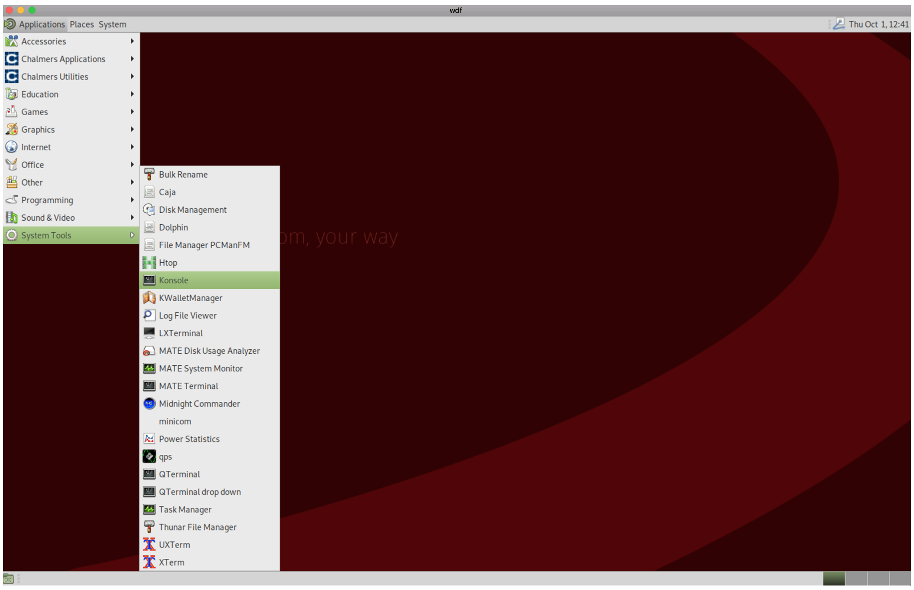


Figure 4.1: Microsoft Remote Desktop

5 CFD workflow

Computational Fluid Dynamics (CFD) is a collective name for methods solving the Navier-Stokes equations numerically. In this assignment you will use the CFD software `Star-CCM`$^+$ to simulate a boundary layer over a flat plate and the flow around a cylinder. As you did in CA1, the domain of interest is divided into discrete elements for which you approximate the field to be a single value. E.g., for each element there is only a single pressure, and velocity. For each element you will then solve a balance of momentum and mass, the relations derived using Reynolds Transport Theorem in White (2016). After discretizing the flow equations, we end up with a matrix system that usually needs a lot of numerical work to be solved. In contrast to CA1, in modern CFD software such as `Star-CCM`$^+$, all numerics and math is carried out in the background. What the user must do is to generate a suitable discrete representation of the geometry at hand. Such a representation is commonly referred to as a *mesh* or *grid*. The tasks then progress to specify appropriate settings for flow physics and boundary conditions.

To better understand the steps required to set up and perform the two simulations in this exercise, it is important to understand what the key steps in a typical CFD simulation are. Generally, there are three main stages in a simulation workflow; *pre-processing*, *running* the simulation and *post-processing*. Where, it is only the pre- and post-processing steps that requires user interaction. However, depending on the type of application studied, the time spent on the different stages in the workflow may vary significantly.

5.1 Pre-processing

This stage involves largely five different objectives.

1. **Import or create a geometry**

   For most industrial applications the geometry will be imported from a CAD model, and you then go directly on to meshing (it might, however, be necessary to simplify the CAD first). For some simpler applications there is a possibility to create the geometry directly in `Star-CCM`$^+$, which is what you will do for the two simulations in this assignment.

2. **Generate a computational mesh from the geometry**

   Generating the mesh can be everything from automatically done to an art form. Depending on the flow phenomena of interest the demands for mesh quality may be very high, the mesh generation process can be a significant part of the simulation work. In this assignment, however, you will use a mostly automated approach.

3. **Select the models to use in the simulation**

   Choosing physical and numerical models is of course solely dependent on what you are trying to simulate and is different from case to case. More detail on what models you will use is given in the procedural description that follows.

4. **Specify boundary conditions**

   Choosing boundary conditions is central to a CFD simulation since they define how surrounding systems, i.e., not included in the simulation, will affect the simulated system.

Sometimes they are simple, e.g., defining a free stream, while they sometimes are complex, e.g. describing pulses from engine cylinders. In the simulations you will perform you will use some of the more common boundary conditions.

5. **Set up any analysis and data collecting that is to run during the simulation**

   Sometimes it is useful to do on-the-fly analysis or monitor your simulation. Probably the most common way to monitor a simulation is to plot the solver residuals, which gives you a measure of how well your matrix is solved and can be used to determine when the simulation has converged. Most software do that automatically but sometimes there are other aspects that are interesting to monitor during the simulation. These monitors need, of course, to be set up prior to running the simulation. In the tasks you will learn to set up such monitors in `Star-CCM`$^+$.

5.2 Post-processing

This is the other stage that involves major user interaction, but the objectives are far less defined than for the pre-processing stage. Post-processing can involve any number of analyses of the results, and it is of course largely dependent on the problem that is solved. For this assignment you will look at some of the common ways to do post-processing and here it is also greatly encouraged to experiment yourselves.

Ever wondered what the pressure field looks like? Check it out!

5.3 `Star-CCM`$^+$ specific workflow

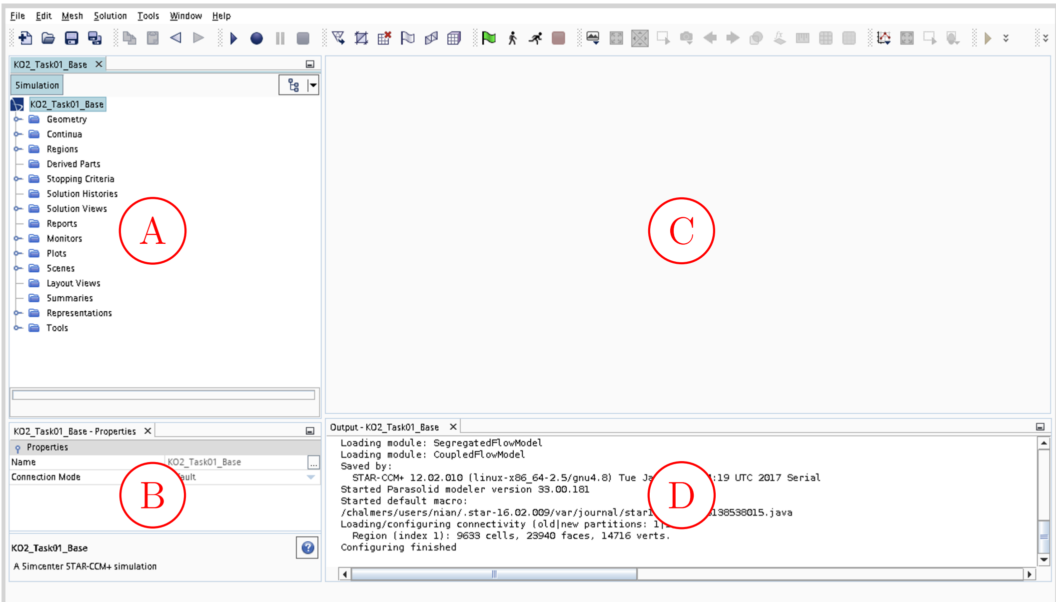


Figure 5.1: `Star-CCM`$^+$ graphic user interface

Figure 5.1 shows the user interface of `Star-CCM`$^+$. In `Star-CCM`$^+$the CFD workflow is mostly represented in the tree structured menu to the left in the screen (A). Each node in the tree gathers similar simulation settings and functionality, usually in more than one level. This effectively end up as a node tree. In this document, node structures will be given in the following format:

```
Geometry -> Parts.
```

Which is a reference to the `Parts` node, which in turn is a sub-node to the `Geometry` node. Roughly speaking the grouping is based on functionality. In `Geometry` you will create your geometry and define your mesh. The `Continua` node will let you choose physics and solver models, under `Regions` you will find information about your computational domain such as mesh information and boundary conditions. Then there are a few nodes associated to running the simulation (`Stopping Criteria`, `Solution Histories`, `Solution Views`), some nodes associated with post-processing (`Monitors`, `Plots`, `Summaries`) and the utility node, `Tools`. The node tree is dynamic, which means that as you choose models and set up your simulation, additional dependency nodes might appear to allow the user to set additional options.

The bottom left part (B) is a window that will display the options of a specific node, the large middle empty part (C) is the graphical window and the lower box (D) is an output terminal.

Note that the workflow of `Star-CCM`$^+$ is designed to go from the top down. If you step away from the top-down procedure, you will find that some operations will not be available unless you do the intended previous operation first. E.g. you cannot set boundary conditions if you have not defined any geometry.

6 Simulation 1 – Flat-plate boundary layer

In your first simulation you will simulate the flow over a flat plate and extract data that will be compared to experimental data. In this task you are provided with a file to start from that contains a pre-defined geometry, a mesh and some basic post-processing settings (`MTF053_CA2_Task-1.sim`). This will allow you to get aquatinted with the available physics models, numerical setup, boundary conditions and post-processing.

6.1 Loading the simulation file

If you are running `Star-CCM`$^+$ remotely, first make sure to setup `Microsoft Remote Desktop` on your computer (see Section 4).

Open a terminal and write `starccm+` in the command line to start the program.

```
[user@remote1 ]$ starccm+
```

The `Star-CCM`$^+$ graphical user interface should now open

Step 1.1 – Open the provided simulation file:

1. Go to `File` in the upper menu and choose `Load...`
2. Activate `Serial`
3. In the `License` drop-down menu select `Simcenter STAR-CCM+ Power on Demand`
4. Fill in the provided `Power On Demand (POD)` key in the `key` field
5. Browse for the file: `MTF053_CA2_Task-1.sim` (*that you should have downloaded now*)
6. Press `OK`

Now the file is loaded and you should see something like what is shown in Figure 5.1

To get aquatinted with the software, browse around and open the different tree nodes. You will notice that a geometry is already set up for you under the `Geometry -> Parts` node. A meshing instruction is also set up for you in the `Geometry -> Operations` node and the mesh is already generated. In this task you will hence focus on choosing relevant physics models, specifying boundary conditions and post-processing of the results.

Step 1.2 – View the mesh:

1. Open the `Scenes` node
2. Double click on `Mesh Scene 1`

This will open the scene in the viewing window as well as opening a new tab (see Figure 6.1)

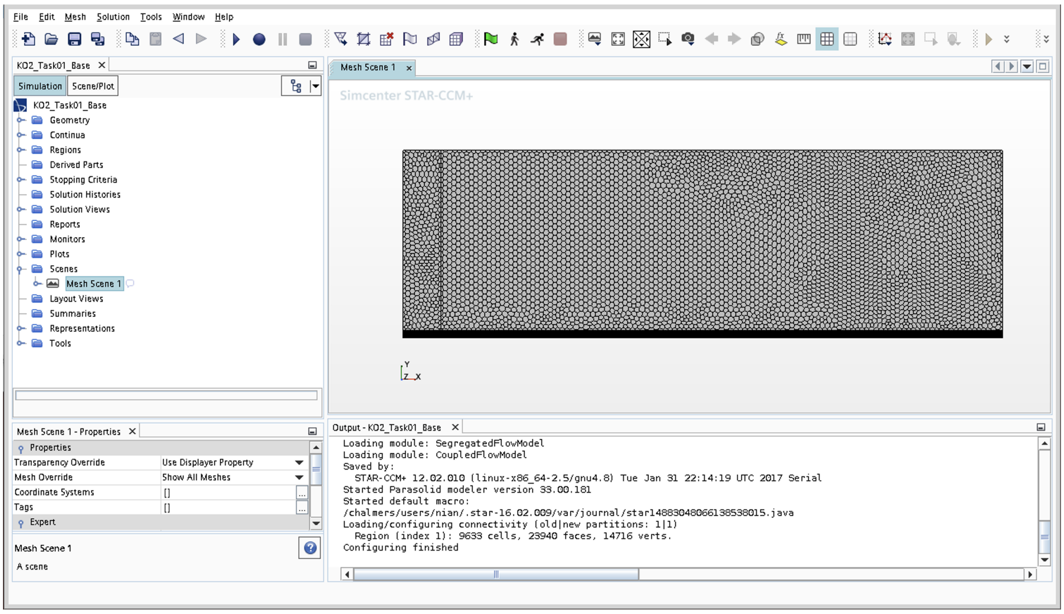


Figure 6.1: Flat-plate boundary layer mesh

6.2 Preparing the simulation

The first step towards running the simulation will be to choose appropriate physics for the simulation. This is a step that can differ significantly from case to case. We will do a rather simple and straight forward setup with an incompressible gas, initially without turbulence models.

Step 1.3 – Physics models setup:

1. Open the `Continua` node
2. Right click on the `Physics 1` sub-node
3. Choose `Select Models...` and select the following models:

   `Two dimensional`

   `Steady`

   `Gas`

   `Segregated flow`

   `Constant density`

   `Laminar`

Most of the selected models are rather self-explanatory, the only one not so easy to understand is probably the `Segregated flow` option. It is an option dictating how the matrix systems should be set up and is not something we need to spend time on in this course.

With the physics set up you can now proceed to set appropriate boundary conditions for the simulation. Under the `Regions -> Region -> Boundaries` node you will find all boundaries of your computational domain. Clicking on a boundary, you will find its type in the properties window to the lower left (Figure 6.2). Choose the appropriate boundary conditions according to the table below.

Step 1.4 – Update boundary conditions:

1. Update boundary condition types: `Regions -> Region -> Boundaries`, update the `Type` for each of the boundaries as follows:

Inlet	`Velocity inlet`
Free Stream Edge	`Symmetry Plane`
Outlet	`Pressure Outlet`
Plate	`Wall`
Top	`Symmetry Plane`

   (Note: the default `Type` is `Wall`)

2. Open up the `Inlet -> Physics Values` node and set the `Velocity Magnitude` to 6 m/s

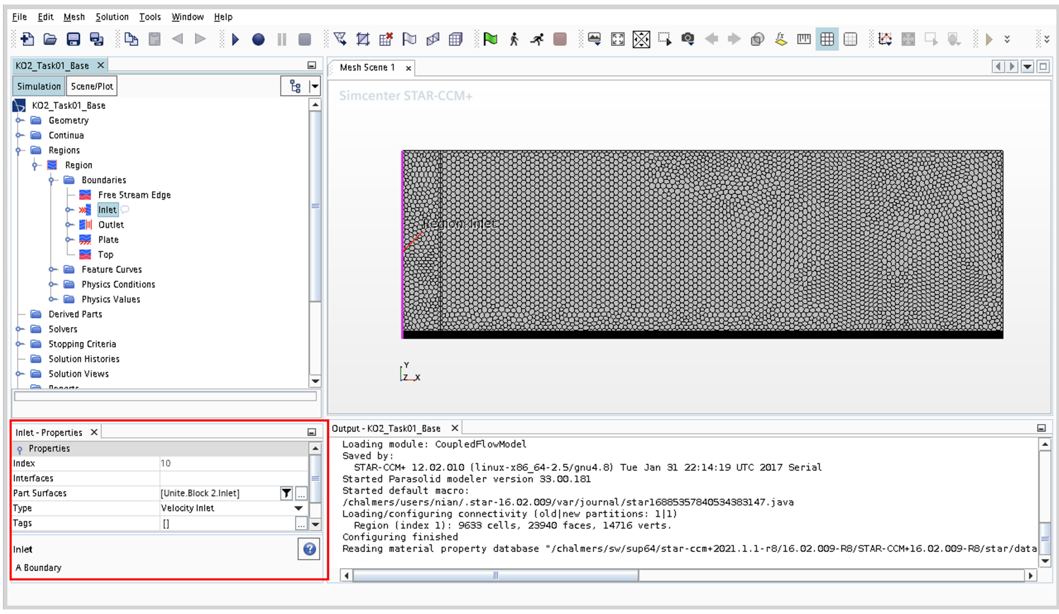


Figure 6.2: Boundary condition properties

In `Star-CCM`$^+$ you can, and probably will, have multiple scenes open to view different aspects of your simulation. One common and useful scene is a scalar scene, which can be used for visualization of a scalar field in your simulation, e.g., a velocity component or pressure.

Step 1.5 – Create a scene to visualize the x-velocity field:

1. Right click on the `Scenes` node, hover over `New Scene` and choose `Scalar`, this will open a new scalar scene, Figure 6.3
2. Right click on the color bar and find `Velocity-i` to display the x-velocity

Note that the scene will not show anything due to the fact that the flow field is not initialized yet.

Step 1.6 – Initialize the flow field:

Click on the green flag in the top menu, Figure 6.4

Now you are almost ready to run the simulation, but before doing so you should specify when the simulation should stop (if you do not stop it manually).

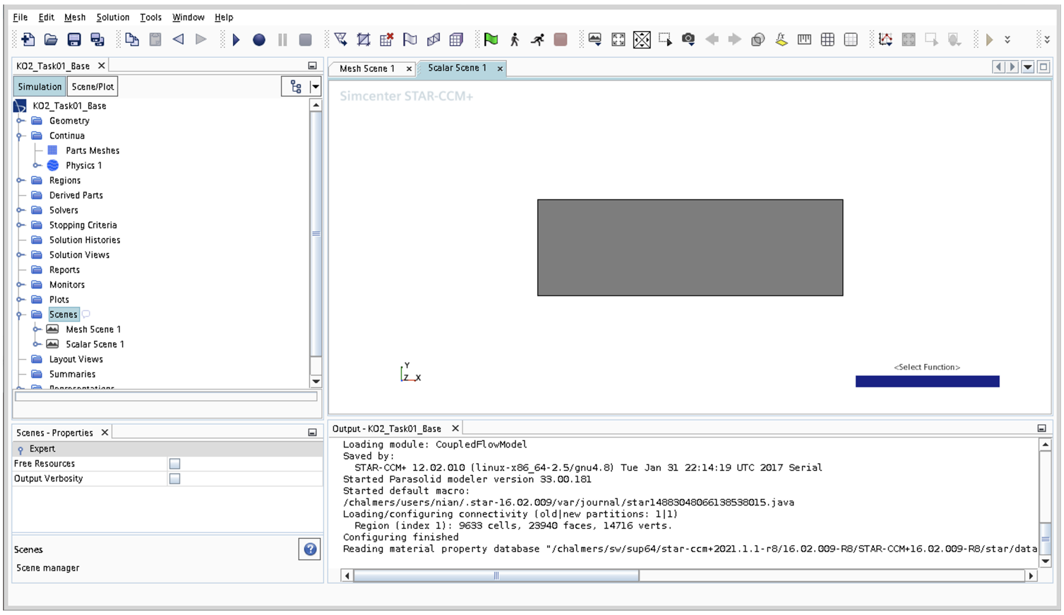


Figure 6.3: Adding a new scene

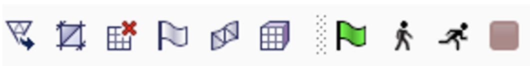


Figure 6.4: `Star-CCM`$^+$ top menu

Step 1.7 – Solver control:

Open the `Stopping Criteria` node and check that maximum steps are set to at least 1000 and that the criterion is enabled.

The only thing left now is to run the simulation. When the simulation is started you will see a monitor showing the solver residuals (Figure 6.5).

Step 1.8 – Run the simulation:

You start the simulation by pressing the running person symbol close to the green initialize flag in the top menu, see Figure 6.4.

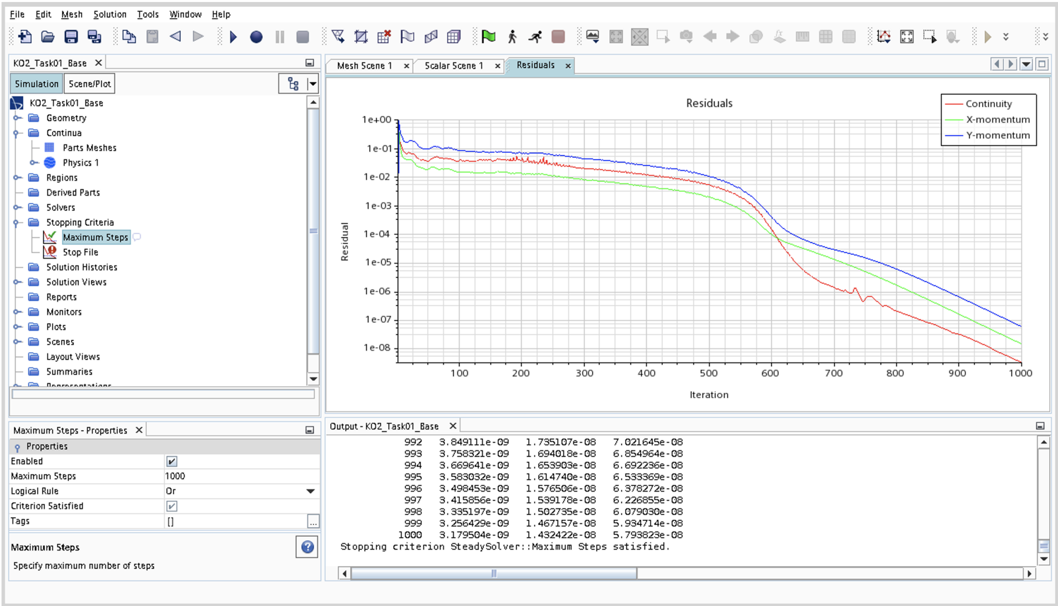


Figure 6.5: Flat-plate boundary layer residual log

6.3 Post-processing

The results from the simulation should be compared to measured velocity profiles extracted at three axial positions in a boundary layer and therefore we need to define probes at these three axial locations ($x = 0.35\ m$, $x = 0.60\ m$, and $x = 0.90\ m$).

Step 1.9 – Generate probe arrays:

1. Create a line along which you can sample properties of the flowfield:

   (a) **Make sure you have a scalar or mesh scene open**

   (b) Right click on the `Derived Parts` node, hover over `New Part -> Probe` and choose `Line`. This will open a new dialog for creating a line probe. You can drag and drop the line end points here, but we want slightly more precision

   (c) Enter the following coordinates for the line: (0.35,0,0), (0.35,0.01,0). This will generate a vertical line at $x = 0.35m$.

   (d) Press `Create`

2. Repeat for $x = 0.60m$ and $x = 0.90m$

Step 1.10 – Generate probe array plots:

1. Create a plot to view the x-velocity along the line:
   (a) Right click the `Plots` node, hover over `New Plot` and choose an `XY plot`. This creates a `XY Plot 1` (or some other number if you have already defined `XY plots`).
   (b) In the properties window (field B in Figure 5.1), choose one of your newly created line probes in the `Parts` menu.
   (c) For the `X Type` properties choose `Scalar` as `Data Type` and make sure that `Scalar` is set for the `Y Types -> Y Type 1` node as well.
   (d) Expand the `X Type` node and in the `Scalar Function` select the (`Velocity -> Laboratory -> i`) as the `Field Function`
   (e) Select the `Centroid -> Laboratory -> Y` as the `Field Function` for the `Y Type 1` scalar function.
2. Repeat for all three line probes generated in the previous step ($x = 0.35\ m$, $x = 0.60\ m$, and $x = 0.90\ m$)

The above methodology is generic for a lot of post-processing done in `Star-CCM`$^+$. Specify where you want to measure. It can, for example, be a point, a line, a plane, or an iso-surface. Then you choose a method of displaying the data, plot or scene, and finally what you want to measure and display. In addition to the plots needed to complete the assignment, you can of course also create your own plots and scenes if you are curious about some flow properties.

Step 1.11 – Export the data in the plot:

Right click on the plot and choose `Export`.

6.4 Simulation 1b – Turbulent flow

Besides solving the laminar case described in the previous section, we are also interested in the turbulent case. Luckily it is rather straight forward to switch to turbulent flow.

Step 1.12 – Set up a new case:

1. Save the laminar simulation
2. Save a copy of the laminar case that you now will convert into a turbulent case

Let's convert the updated the laminar flow settings for simulation of a turbulent boundary layer.

Step 1.13 – Physics models setup:

1. Right click `Continua -> Physics 1` and choose `Select Models...`
2. In the models dialog deselect `Laminar` and instead choose:

   `Turbulent`

   `Spallart-Allmaras`

3. Change the inlet boundary condition to $9.1 m/s$
4. Increase the maximum number of steps by 2000 in the `Stopping Criteria`.

Now, run the new simulation and export data for later use.

Step 1.14 – Run the simulation and export data:

1. Start the simulation by pressing the running person symbol in the top menu
2. Export the turbulent boundary profile for the line located at $x = 0.90\ m$

7 Simulation 2 – Flow over a cylinder

In the second simulation, you will simulate a two-dimensional flow over a cylinder and compare to measured data. This is a flow application that contains a lot of physics, and you will, for example, get to see how the pressure is distributed around the cylinder and see what gives rise to the drag force on a body. This task will require you to set up the geometry and mesh yourselves, which means that you will learn the steps left out in the first simulations. After this simulation you will have done all the basic steps of a CFD simulation in `Star-CCM`$^+$ and should be able to set up a simulation on your own.

7.1 Pre-processing

For this simulation you will start from scratch.

Step 2.1 – Setup a new simulation:

1. Go to `File` in the upper menu and choose `New...`
2. Activate `Serial`
3. In the `License` drop-down menu select `Simcenter STAR-CCM+ Power on Demand`
4. Fill in the provided `Power On Demand (POD)` key in the `key` field (probably already filled in)
5. Press `OK`

7.1.1 Creating the geometry

To simulate the flow around a cylinder you need a cylinder shape and a box around it describing the computational domain. In `Star-CCM`$^+$, geometry is in general three-dimensional, therefore you will first create a three dimensional geometry that you, in the meshing stage, will use to generate a two-dimensional mesh. Remember that the geometry you will create is the fluid domain, all solid shapes (such as the cylinder) should be removed from your domain. Thus, the procedure will be to create a cylinder and a box around it and finally remove the cylinder from the box.

Step 2.2 – Create the cylinder:

1. Right clicking the `Geometry -> Parts` node, hover over `New Shape Part` and choose `Cylinder`
2. Set the start coordinates to (0,0,-0.1)
3. Set the radius $r = 0.015m$
4. Set the end coordinates to (0,0,1.1)
5. Press `OK`

Remember: if you want to view your geometry you need to create a scene first.

Next, create the box around the cylinder. The box will constitute the computational domain (the fluid volume); hence it needs to extend to all areas where the interesting flow phenomena occurs. For this case it is sufficient to have a shorter section before the cylinder, we expect nothing to happen until the flow reaches the cylinder, and then a longer section is required after the cylinder. This region is where we expect to see separation and a vortex street.

Step 2.3 – Generate the outer boundaries of the computational domain:

1. Right click on the `Geometry -> Parts` node, hover over `New Shape Part`, select `Block`
2. `Corner 1` should be (-0.06,-0.075,0)
3. `Corner 2` should be (0.2,0.075,1)
4. Press `OK`

When the block is created, you will under `Block -> Surfaces` notice that the block consists of a single surface, which will make the definition of boundary conditions impossible.

Split 2.4 – Split the surface outer surface of the computational domain:

1. Right click the surface (`Geometry -> Parts -> Block -> Surfaces -> Block Surface`) and choose `Split by Angle`
2. Use the default 89 degrees and press `OK`

This splits the computational domain boundary into six different surfaces, which will allow you to set correct boundary conditions later. For future convenience rename the surfaces as follows:

Step 2.5 – Rename the generated boundary surfaces:

`Block Surface` -> Front

`Block Surface 2` -> Bottom

`Block Surface 3` -> Inlet

`Block Surface 4` -> Top

`Block Surface 5` -> Back

`Block Surface 6` -> Outlet

Note: the automatic surface numbering may be different from the example above. In order to make sure that the surfaces are renamed correctly, set up a geometry scene by right-clicking `Scenes -> New Scene` and selecting `Geometry`. Now, with the help of the new `Geometry Scene`, make sure that the renaming is done in accordance with Figure 7.1 (again, please note that surface numbers may be different than in this example).

The last thing you need to do is to subtract the cylinder from the flow domain.

Step 2.6 – Subtract the cylinder from the flow domain:

1. Right click on the `Geometry -> Parts -> Block` node and choose `Create Mesh Operation -> Boolean -> Subtract`
2. Select both `Block` and `Cylinder` as `Input Parts`
3. Select `Block` as `Target Part`
4. Press `OK`

This will add the subtract operation to the `Geometry -> Operations` node and also create a new part called `Subtract`. As you might notice it is marked with a warning sign, which indicates that it is not updated. In general, operations do not automatically execute when you create them but rather when they need to be executed. The subtract will take place when either you explicitly choose to execute it, or when its needed, e.g., when you choose to generate a new mesh.

The geometry for the simulation is now defined – save the simulation

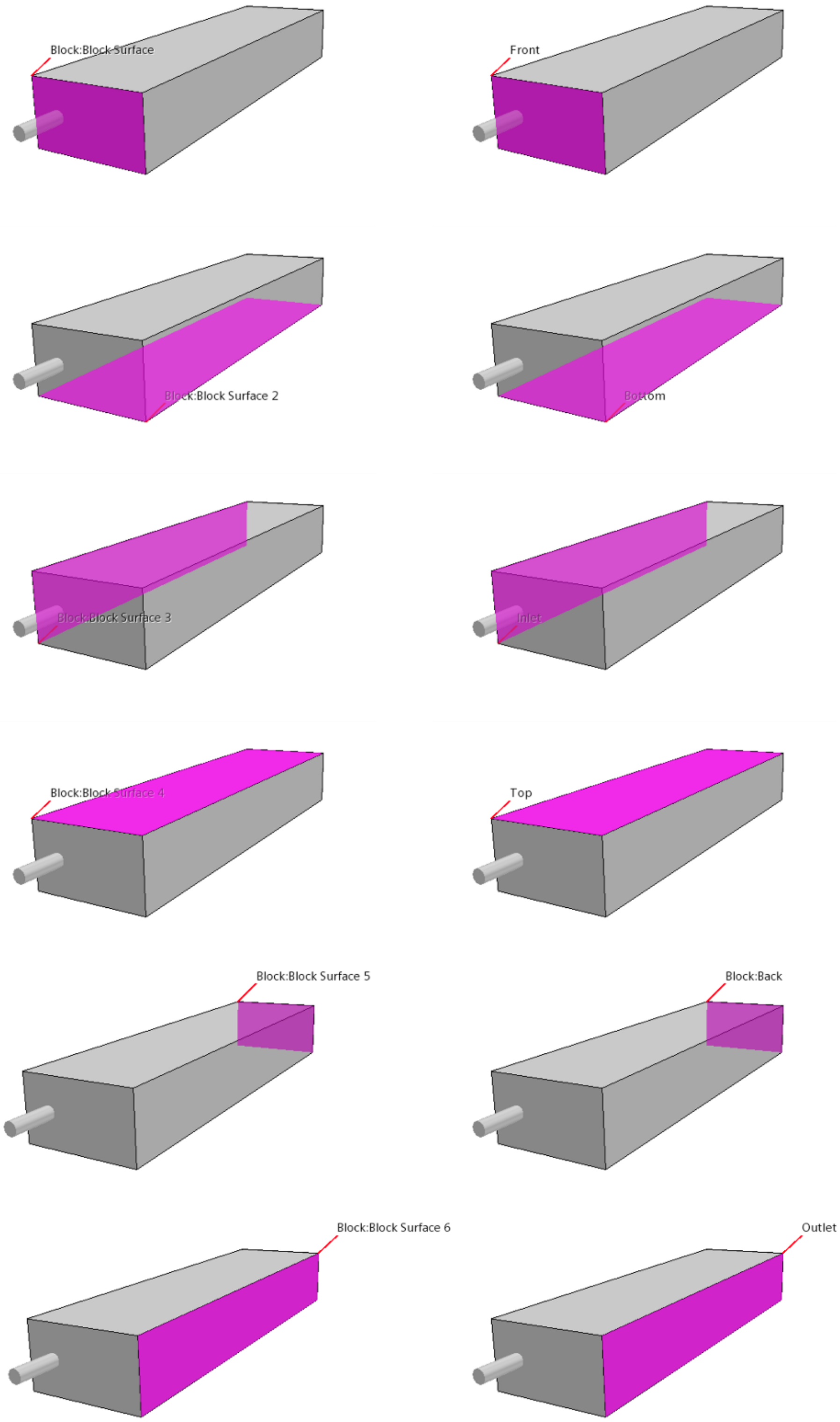


Figure 7.1: Renaming of generated boundary faces

7.1.2 Generating the mesh

The first step is to convert the three-dimensional geometry into a two-dimensional representation. In `Star-CCM`$^+$ this is done with an operation called `Badge for 2D Meshing`:

Step 2.7 – Create a two-dimensional representation:

1. Right click the subtract part (`Geometry -> Parts -> Subtract`), choose `Create Mesh Operation -> Mesh -> Badge for 2D Meshing`
2. Press `OK` in the dialog that pops up
3. Right click the newly created operation (`Geometry -> Operations -> Badge for 2D Meshing`), and select `Execute`

The last step before choosing the mesh options is to generate the container for the computational domain, this type of container is called a `Region` in `Star-CCM`$^+$:

Step 2.8 – Create a `Region`:

1. Right click the subtract part (`Geometry -> Parts -> Subtract`), and choose `Assign Parts to Regions...`
2. Switch the option `Create a Boundary for Each Part` to `Create a Boundary for Each Part Surface`, then press `Apply`.

If you now open up the `Regions` node, you will find the computational domain `Region` there. To create the mesh, you first need to update the meshing instruction:

Step 2.9 – Update the meshing instructions:

1. Right click `Geometry -> Parts -> Subtract` node and select `Create Mesh Operation -> Mesh -> Automated Mesh (2D)`. This will bring up a new dialog where you can select mesh type and add a new operation in the `Operations` node.
2. For this project you will select

   `Polygonal Mesher`

   `Prism Layer Mesher`
3. Press `OK`

In the last step before generating the mesh, you will need to set some parameters defining how the meshing is to be done. Under the newly created `Automated Mesh (2D)` node under `Operations` there are three different settings nodes, `Meshers`, `Default Controls` and `Custom Controls`. The `Meshers` are some general settings for the mesh operations, you will not change anything there. However, under the `Default Controls` you will find settings that apply to the entire mesh and under `Custom Controls` you can set up special rules for surfaces or volumes. We will do both now, start by changing the following parameters under the `Default Controls` node:

Step 2.10 – Update global mesh controls:

Open the node `Geometry -> Operations -> Automated Mesh (2D) -> Default Controls` and make the following changes:

`Base Size`	0.004 m	
`Target Surface Size`	200%	`Percentage of Base`

These baseline settings will give a reasonable starting point for the mesh. However, these settings alone will not yield a sufficiently fine mesh around and after the cylinder. These are regions where we expect some interesting physics, e.g., separation and a vortex street, to occur, thus we need to make sure that we do an accurate calculation of that region. Therefore you will use the `Custom Controls` to specify mesh refinements around the cylinder.

Step 2.11 – Update local mesh controls:

1. Right click on the `Geometry -> Operations -> Automated Mesh (2D) -> Custom Controls`, hover over `New` and choose `Surface Control`. This will generate a new `Surface Control` node under `Custom Controls`.

2. Click on the newly created node and, in the properties window (field B in Figure 5.1), select the `Cylinder Surface` (`Subtract -> Cylinder -> Cylinder Surface`) in the `Part Surfaces` option

3. Under the `Surface Control` node you will find two additional nodes, `Controls` and `Values`. In the `Controls` node you select what controls you want to customize while in the `Values` node specify the custom values for the selected controls.

4. In the `Controls` node switch the following controls from `Parent` to `Custom`,

   `Target Surface Size`

   `Prism Layers`

5. When `Prism Layers` was changed to `Custom`, additional options appear. Of these new options, you should activate the following three:

   `Customize Number of Layers`

   `Customize Total Thickness`

   `Customize Distribution`

6. Enable the `Specify wake refinement options` in the `Wake Refinement` controls

7. In the `Values` node, specify the following:

`Target Surface Size`	25%
`Custom Prism Values -> Prism Layer Total Thickness`	25%
`Wake Refinement -> Isotropic Size`	25%

   All of the above should be given as `Percentage of Base`. Now click on `Custom Prism Values` and update the values in the properties window as follows:

`Number of Prism Layers`	15
`Prism Layer Stretching`	1.2

The updates to `Custom Controls` will allow you to choose a finer mesh resolution around the cylinder, increase the resolution of the boundary layer, and make the mesh finer behind the cylinder, i.e., in the wake region.

Step 2.12 – Generate the mesh:

You can now generate the mesh choosing one of the two ways below:

1. Right clicking on the mesh operation itself and choose `Execute`
2. Click the mesh cube icon next to the green initialization flag in the top menu (see Figure 6.4).

Do not forget that you need to open a new scene if you want to look at the mesh. It should look similarly to the mesh in Figure 7.2.

The mesh is now generated – save the simulation

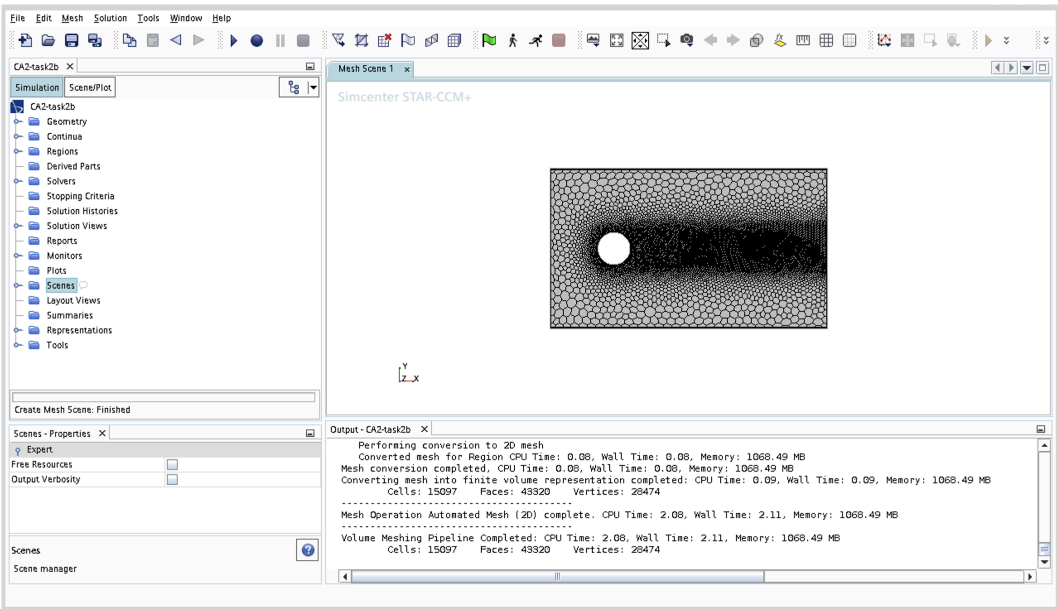


Figure 7.2: Cylinder mesh

7.1.3 Physics and boundary conditions

When the geometry and mesh is generated the simulation is roughly at the same state as the first simulation after loading the provided file. Thus, the process from here on will be similar to that of setting up the first simulation and the instructions will therefore be less detailed. If you do not remember a step, please go back to the instructions for simulation 1.

Step 2.13 – Physics models setup:

1. Open the `Continua` node
2. Right click on the `Physics 1` sub-node
3. Choose `Select Models...` and select the following models:

   `Implicit Unsteady`

   `Gas`

   `Segregated Flow`

   `Constant Density`

   `Turbulent`

   `K-Omega Turbulence`

   `Gamma Transition`

The following boundary conditions will be used for this simulation:

Step 2.14 – Boundary conditions:

1. Open the `Regions -> Region -> Boundaries` node and update `Type` for each of the boundaries as follows:

Top	`Symmetry`
Bottom	`Symmetry`
Inlet	`Velocity Inlet`
Outlet	`Pressure Outlet`
Cylinder	`Wall`

2. Open the `Inlet -> Physics Values` node and set the `Velocity Magnitude` to 10.0 m/s

When the boundary conditions are changed in a simulation, sometimes the mesh will need to be regenerated. E.g., you generally do not need prism layers on an inlet, `Star-CCM`$^+$ recognizes that and changes accordingly. Since you now changed the boundary conditions you also need to generate a new mesh – do that now. The mesh should now look like the one shown in Figure 7.3.

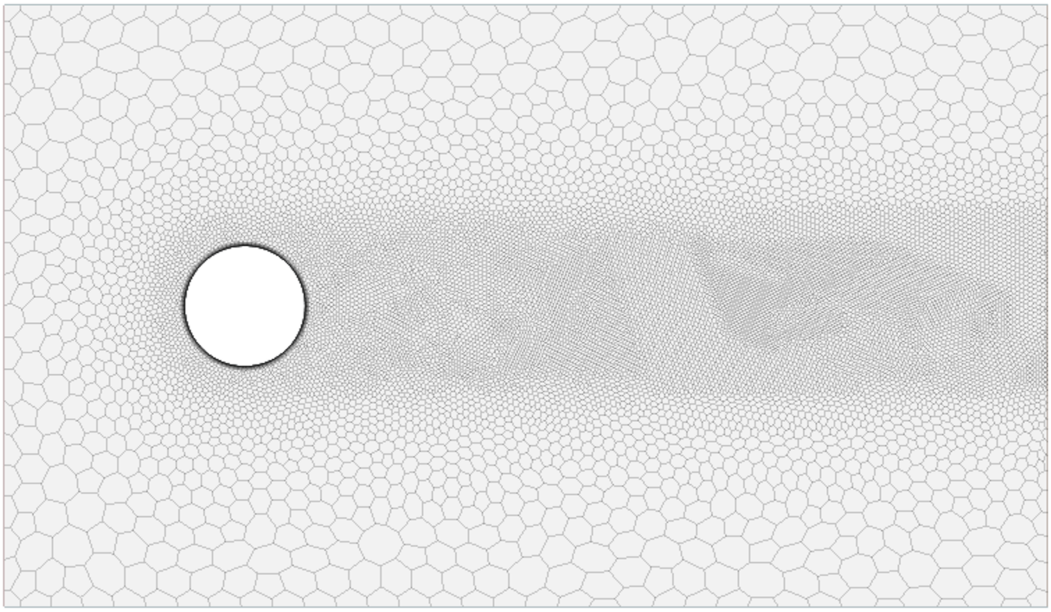


Figure 7.3: Final mesh of the two-dimensional cylinder flow computational domain.

Step 2.15 – Regenerate the mesh:

Click the mesh cube icon next to the green initialization flag in the top menu.

You are soon ready to start the simulation, but first: save

7.1.4 Running the simulation

The simulation you have set up now is a transient simulation, i.e. it will resolve the flow in time, therefore we will use a specified physical time for the simulation as a `Stopping Criteria` rather than a specified maximum number of iterations as done for the first simulation, which was a steady-state case. The solver `Stopping Criteria` should be set up as follows:

Step 2.16 – Solver control:

1. Click on the `Stopping Criteria -> Maximum Physical Time` node and change its value 0.5 s
2. There is also a conflicting criteria, `Maxim Steps`, that you should disable. Do that by making sure that the `Enabled` box is unchecked.

For this simulation there are a number of interesting post-processings you can do. The most important ones are prepared for you in a script, `setupPost.java`, which when you run it will set up:

1. A scalar scene visualizing `Vorticity` (vorticity is a measure of rotation and useful for visualizing vorticies)
2. A plot of the `Static Pressure` and `Mean of Static Pressure` on the cylinder surface
3. The `Maximum` and `Minimum` of the `Mean of Static Pressure` field

Step 2.17 – Run the script:

Press the play symbol at the top of the window, close to the run and initialize symbols, and select the provided `Java` script (`setupPost.java`).

Step 2.18 – Now its time to run the simulation ...

1. Press the running person icon in the top menu to start the simulation
2. When the simulation starts, a residual log window will open. Switch to the `Scalar Scene` showing `Vorticity` and observe what happens with the flow as the simulation runs...

Step 2.19 – Export cylinder pressure data

When the simulation stops (when the specified `Stopping Criteria` is fulfilled), open the `XY Plot` showing `Static Pressure` and `Mean of Static Pressure` on the cylinder surface (the plot should already be open but if that is not the case, expand `Plots` in the right menu and open the corresponding `XY Plot`.

With the plot window open, `right click` somewhere in the plot and select `Export` in the drop down menu to export the data to file. You will use the exported data later.

7.1.5 Cylinder force

As the final part of the simulation work you should now extract the fluctuating force on the cylinder for later analysis. To do that you will need to set up a force monitor for the cylinder. Based on your observations in the previous step, can you think of a reason for fluctuations in the force on the cylinder?

Step 2.20 – Force on the cylinder:

1. Right click `Report` and select `Report -> New Report -> Flow/Energy -> Force`. This will generate a new `Report` named `Force 1` (or some other number number if you already have generated other `Force Reports`)
2. Click on `Force 1` and update the `Report` settings as appropriate (the example below will give a `Report` that monitors the cylinder lift force)
   (a) Set the the `Direction` to (0,1,0), which will give you the lift force on the cylinder, i.e., the force in the flow-normal direction
   (b) Set `Parts` to `Parts -> Subtract -> Cylinder -> Cylinder Surface`
3. Right click on the newly generated `Report -> Force 1` and select `Create Monitor and Plot from Report` to create a plot

In order to generate force data you will need to run the simulation again.

Step 2.21 – Update the simulation time:

1. Click on the `Stopping Criteria -> Maximum Physical Time` node and change its value 1.0 s

Step 2.22 – Sample cylinder force data:

1. Start the simulation
2. When the simulation is done, export the cylinder force monitor data

The cylinder force monitor log window should now look like in Figure 7.4. With the second simulation completed, the simulation part of CA2 is now done.

7.1.6 Post-processing

In the previous simulation, you got some tips on how to set up scenes for visualization and how to make plots. Use that knowledge now to investigate different aspects of your simulation. Generally, you should not only use post-processing for understanding your physics, but also to make sure that your simulation is of good quality.

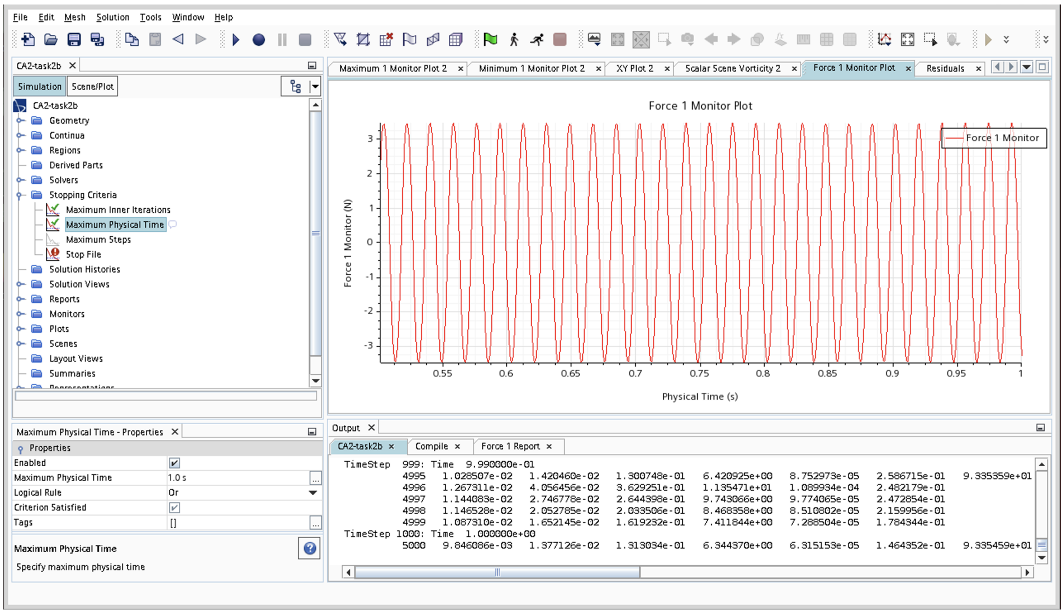


Figure 7.4: Fluctuating cylinder lift force

Besides the post processing set up for you when running the provided script, some other interesting things to visualize might be for example `Velocity Magnitude`, `Static Pressure`, and `Mean of Static Pressure`. Two suggestions for additional post-processing can be found below.

Example 1 – Pressure probe behind the cylinder:

1. Make sure to have a `Mesh Scene` open
2. Right click on the `Derived Parts` node, hover over `New Part -> Probe` and choose `Point`. This will open a new dialog for creating a point probe.
3. Specify the probe coordinates – for example (0.03,0.0,0.0), which will result in a probe located in the wake behind the cylinder – and then press `Create`
4. Right click `Report` and select `Report -> New Report -> User -> Maximum`. This will generate a new `Report` named `Maximum 1` (or some other number number if you already have generated other `Maximum Reports`)
5. Click on `Maximum 1` and update the `Report` settings as appropriate
   (a) Set the the `Field Function` to `Mean of Static Pressure`
   (b) Set `Parts` to `Derived Parts -> Point` (or what your recently generated `Point Probe` is called

Example 2 – Velocity profile in the cylinder boundary layer:

1. Make sure to have a `Mesh Scene` open
2. Right click on the `Derived Parts` node, hover over `New Part -> Probe` and choose `Line`. This will open a new dialog for creating a line probe.
3. Change `Coordinate System` to `Laboratory -> Cylindrical 1`
4. Enter the end coordinates for the line – for example:

r_1	0.015	m	r_2	0.017	m
θ_1	1.5707963267948966	rad	θ_2	1.5707963267948966	rad
z_1	0.0	m	z_2	0.0	m

   which will generate a line probe starting at the top of the cylinder and extending radially out into the boundary layer.
5. Change `Resolution` from the default value (20) to 100
6. Press `Create`
7. Right click the `Plots` node, hover over `New Plot` and choose an `XY plot`. This creates a `XY Plot 1` (or some other number if you have already defined `XY plots`).
8. For the `X Type` properties choose `Scalar` as `Data Type` and make sure that `Scalar` is set for the `Y Types -> Y Type 1` node as well.
9. Expand the `X Type` node and in the `Scalar Function` select `Velocity -> Laboratory -> Cylindrical 1 -> Tangential` as the `Field Function`
10. Select the `Centroid -> Laboratory -> Cylindrical 1 -> theta` as the `Field Function` for the `Y Type 1` scalar function.
11. In the properties window, choose your newly created line probes in the `Parts` menu.

8 Assignments

8.1 Flat-plate boundary layer analysis

This part of CA2 is about the analysis of experimental data measured on a flat plate in a wind tunnel (both laminar and turbulent boundary layer) and analysis of your simulated data. This exercise contains several important steps, including,

- Determination of the displacement thickness δ^*
- Verifications of self-similarity of the velocity profiles
- Estimation of were in the boundary layer the maximum turbulent intensity is reached

8.1.1 Laminar flow

Task 1.1:

From the measured laminar boundary layer velocity profiles provided in the `MTF053_CA2.ipynb` (also in Appendix A), determine the displacement thickness δ^*, momentum thickness θ, and the Reynolds-number ratio $Re_{\delta^*}/\sqrt{Re_x}$ for the three provided measured laminar boundary layer profiles. Compare your result to what can be expected from theory. Are the calculated values inline with theory?

Hints!

- In `Python` you can do the numerical integration using the function `numpy.trapz` in the `numpy` library. When estimated the integral, make sure to only use measurement points that are within the boundary layer using

$$u(\delta) \approx 0.99U_o$$

- The following relation derived from the Blasius solution for laminar boundary-layer flows can be of use

$$\frac{\delta^*}{x} = \frac{1.721}{\sqrt{Re_x}}$$

Write down your answer in `Jupyter Lab`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.

Task 1.2:

The solution to the laminar flat plate velocity field was given by Blasius in 1908. The Blasius profile is provided in MTF053_CA2.ipynb (and also Appendix B). Please note that Blasius solution is not given as U and y, but instead in the variables f' and η, defined as

$$f'(\eta) = \frac{\overline{u}}{U_o} \tag{8.1}$$

$$\eta = y\sqrt{\frac{U_o}{x\nu}} \tag{8.2}$$

Verify that the velocity profiles obtained from the wind tunnel experiment (Task 1.1) are self-similar by transforming the laminar measurement data from the three locations to Blasius variables. Plot them (in one figure) together with the Blasius solution.

8.1.2 Turbulent flow

Task 1.3:
For the measured turbulent boundary-layer data provided in MTF053_CA2.ipynb (and also Appendix C), the friction velocity u^* can be estimated to be 0.4171

a) Plot the mean velocity profile as $u^+ = f(y^+)$

b) Plot the rms-profile as the turbulence intensity $u'/u^* = f(y^+)$

c) Identify the different layers in turbulent boundary flow

d) In which layer is maximum turbulent intensity reached and why?

Note! you should use log-scale for the y^+ axis and a linear scale for the u^+ and u'/u^* axis. Use `semilogx` with y^+ on the x-axis.

Write down your answer in `Jupyter Lab`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.

8.1.3 Comparison between simulations and experiments

You should now compare the CFD results from simulation 1 with the experimental data provided, both for laminar and turbulent flow.

Task 1.4:

a) Plot the three different velocity profiles obtained from the laminar boundary layer simulation (0.35m, 0.60m, 0.90m) compared to the corresponding data provided in `MTF053_CA2.ipynb` (also in Appendix A). The plot should be somewhat like Figure 8.1.

b) Comment on, and try to explain, any significant differences in the profiles extracted from the CFD simulation and the corresponding profiles generated from measured data.

Write down your answer in `Jupyter Lab`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.

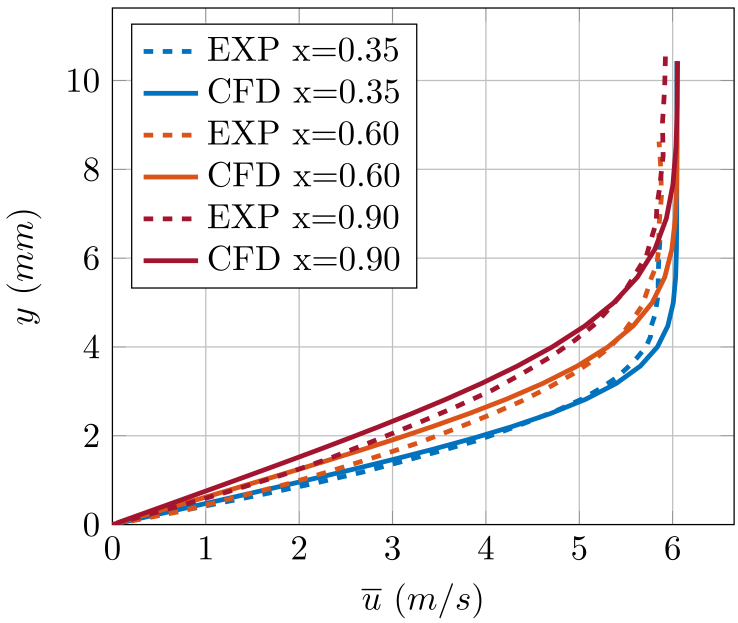


Figure 8.1: Laminar boundary layer velocity profiles – CFD results compared with measured data at three axial positions in a laminar boundary layer.

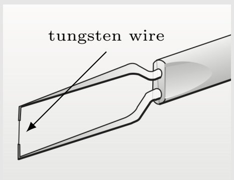
Task 1.5:

a) Plot the turbulent data from the CFD simulation compared to the corresponding data provided in `MTF053_CA2.ipynb` (also in Appendix C). The plot should be somewhat similar to Figure 8.2.

b) Comment on the predicted turbulent flat-plate boundary layer velocity profile in relation to the corresponding measured data

- Does the result look like you expected?
- Do you think that the results could be improved and in that case how?
- Can you think of any significant sources of error?

Note! it might be good to know that the measured data were obtained using hot-wire anemometry, a technique where a thin tungsten wire is heated by an electric current and the cooling effect introduced by the flow is measured, which can be converted to fluid velocity. When a flat-plate boundary layer is measured, the hot-wire probe (see illustration below) is traversed vertically through the boundary layer to measure the velocity at different wall-normal coordinates.



Write down your answer in `Jupyter Lab`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.

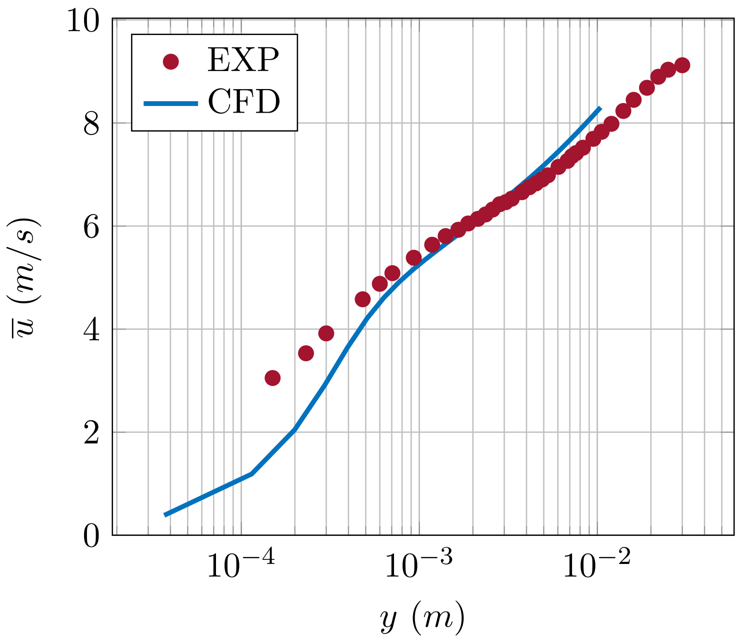


Figure 8.2: Turbulent boundary layer velocity profile – CFD results compared with measured data.

8.2 Flow around a cylinder

Task 2.1:

Plot the `Mean Static Pressure` profile along the cylinder surface using data extracted from the simulation and the provided (`MTF053_CA2.ipynb` and also in Appendix D) measured pressure distribution over a cylinder in the same figure (if you have done the hands-on lab – *flow around immersed bodies* – you can use the measured data from the lab as well).

Do the numerical results match qualitatively and/or quantitatively with the measured data? E.g., is the separation point the same?

Write down your answer in `Jupyter Lab`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.

Task 2.2:

Show the residuals plot and another plot that justifies that the simulation has run long enough and is converged.

Note: you might need to run it longer than the specified 0.5s.

Write down your answer in `Jupyter Lab`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.

Task 2.3:

The cylinder force monitor signal that you extracted as part of simulation 2 should, if done correctly, be oscillating with a very pronounced sinusoidal shape. Calculate the dominating frequency of the signal. Calculate the corresponding Strouhal number (non-dimensional frequency).

$$St = \frac{fD}{U_\infty}$$

a) What Strouhal number do you get?

b) Compare your result with the data presented in Figure 8.3. Does your result agree with the data presented in the figure?

c) What is the cause of the oscillation of the cylinder force?

Hint: run the cylinder simulation with the `Vorticity Scene` open and observe what happens with the flow over time as the simulation runs.

Write down your answer in `Jupyter Lab`. Derivations made using pen and paper or tablet can be included as photos/images in the `Jupyter NoteBook`.

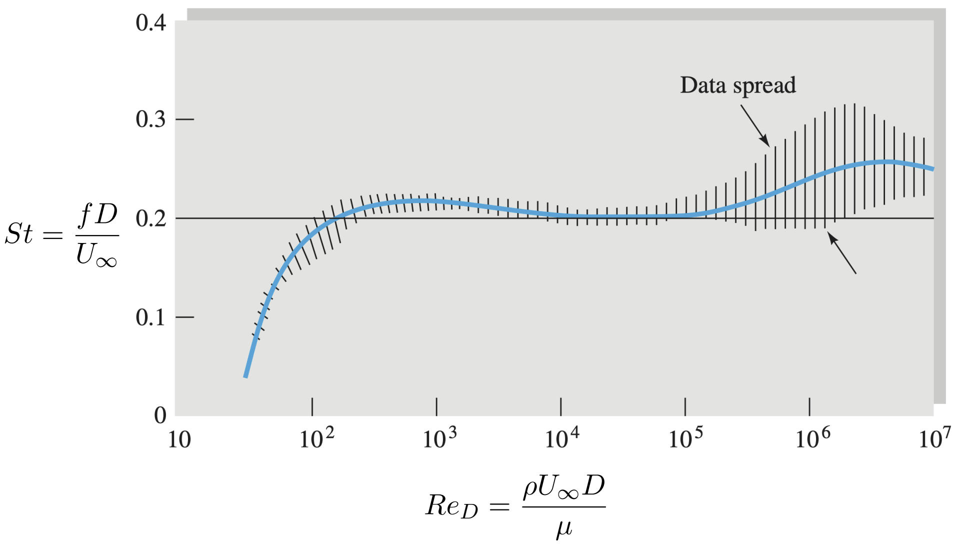


Figure 8.3: Cylinder Strouhal number versus Reynolds number

References

F. M. White. *Fluid Mechanics.* McGraw-Hill, New York, 8:th edition, 2016.

A Measured Laminar Boundary Layer Data

$x = 0.35m$		$x = 0.60m$		$x = 0.90m$	
u $[m/s]$	y $[m]$	u $[m/s]$	y $[m]$	u $[m/s]$	y $[m]$
0.000	0.000	0.000	0.000	0.000	0.000
0.560	0.231	0.737	0.300	0.684	0.400
1.139	0.481	1.344	0.625	1.253	0.750
1.669	0.706	1.891	0.925	1.830	1.125
2.167	0.931	2.367	1.225	2.335	1.500
2.678	1.181	2.859	1.550	2.789	1.875
3.078	1.406	3.291	1.850	3.259	2.275
3.509	1.656	3.662	2.150	3.677	2.650
3.886	1.881	4.049	2.475	4.066	3.025
4.222	2.131	4.376	2.775	4.388	3.400
4.505	2.356	4.661	3.075	4.692	3.775
4.771	2.581	4.938	3.400	4.952	4.150
5.032	2.831	5.149	3.700	5.175	4.525
5.218	3.056	5.331	4.000	5.336	4.900
5.396	3.306	5.473	4.325	5.493	5.275
5.514	3.531	5.574	4.625	5.602	5.650
5.618	3.756	5.672	4.925	5.719	6.050
5.716	4.006	5.726	5.250	5.769	6.425
5.762	4.231	5.771	5.550	5.824	6.800
5.772	4.481	5.830	5.850	5.831	7.175
5.808	4.706	5.849	6.175	5.858	7.550
5.845	5.181	5.855	6.775	5.898	8.300
5.845	5.656	5.880	7.400	5.891	9.050
5.855	6.131	5.873	8.025	5.914	9.825
5.869	6.581	5.854	8.625	5.923	10.575

B Blasius Laminar Profile

η	$\frac{u}{U_o}$	η	$\frac{u}{U_o}$	η	$\frac{u}{U_o}$	η	$\frac{u}{U_o}$	η	$\frac{u}{U_o}$	η	$\frac{u}{U_o}$
0.0	0.00000	1.0	0.32979	2.0	0.62977	3.0	0.84605	4.0	0.95552	5.0	0.99155
0.2	0.06641	1.2	0.39378	2.2	0.68132	3.2	0.87609	4.2	0.96696		
0.4	0.13277	1.4	0.45627	2.4	0.72899	3.4	0.90177	4.4	0.97587		
0.6	0.19894	1.6	0.51676	2.6	0.77246	3.6	0.92333	4.6	0.98269		
0.8	0.26471	1.8	0.57477	2.8	0.81152	3.8	0.94112	4.8	0.98779		

C Measured Turbulent Boundary Layer Data

y $[m]$	$\overline{u}$ $[m/s]$	u_{rms} $[m/s]$	y $[m]$	$\overline{u}$ $[m/s]$	u_{rms} $[m/s]$
0.00015000	3.05100000	0.77100000	0.00452500	6.82800000	0.65400000
0.00023100	3.53100000	0.82600000	0.00490000	6.90500000	0.64800000
0.00030000	3.91700000	0.85800000	0.00527500	6.98500000	0.63200000
0.00048100	4.57900000	0.87400000	0.00605000	7.14800000	0.62600000
0.00060000	4.87900000	0.85500000	0.00680000	7.26500000	0.61600000
0.00070600	5.08600000	0.83600000	0.00760000	7.41700000	0.59900000
0.00093100	5.38500000	0.82600000	0.00830000	7.52000000	0.58600000
0.00118100	5.63700000	0.77200000	0.00950000	7.69300000	0.56400000
0.00140600	5.80500000	0.75900000	0.01057000	7.82900000	0.54700000
0.00165600	5.92800000	0.73000000	0.01200000	7.98400000	0.52200000
0.00188100	6.05000000	0.71900000	0.01400000	8.23300000	0.48400000
0.00213100	6.13900000	0.70400000	0.01600000	8.44900000	0.44000000
0.00235600	6.22500000	0.69900000	0.01900000	8.68300000	0.37800000
0.00258100	6.32000000	0.69600000	0.02200000	8.89700000	0.30800000
0.00283100	6.42400000	0.68400000	0.02500000	9.03400000	0.22200000
0.00305600	6.46300000	0.67500000	0.03000000	9.12100000	0.11500000
0.00330600	6.53100000	0.67900000			
0.00377500	6.65700000	0.66300000			
0.00415000	6.75300000	0.66200000			

D Cylinder Pressure

θ [°]	Δp $[Pa]$	θ [°]	Δp $[Pa]$
0	56.898	80	-34.335
10	52.974	90	-30.411
20	39.240	100	-30.411
30	19.620	120	-32.373
40	-1.962	140	-30.411
50	-23.544	160	-30.411
60	-37.278	180	-30.411
70	-42.183		