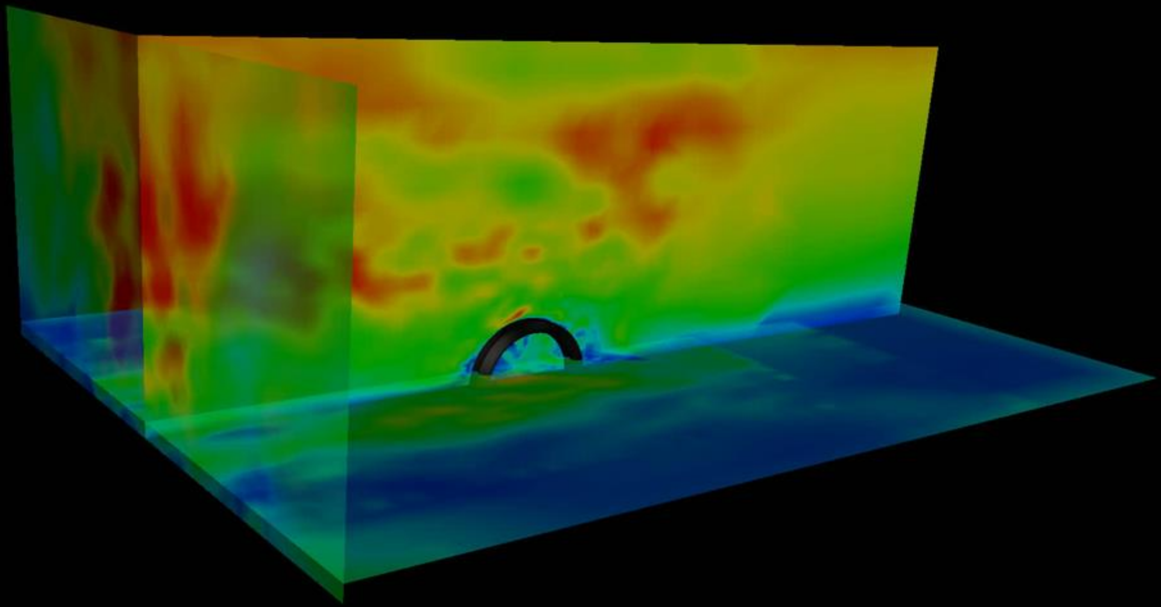
Chair of Structural Analysis
TUM Department of Civil, Geo and Environmental Engineering
Technical University of Munich



STRUCTURAL WIND ENGINEERING

Roland Wüchner

Máté Péntek

Anoop Kodakkal

Presentation material from internal and external sources have been used either directly, modified or adapted to fit the purpose. Effort is continuously being made to accurately reference these. Nonetheless, check referencing in both the script as well as slides for completeness. In case of inconsistencies or mistakes please contact us!

Kratos 2D CFD Tutorial

In this tutorial we will solve another 2D example using GiD and Kratos

Covered topics:

- Predefined example for simulation (aim of the current lecture, do not forget to do the necessary modifications in the setup parameters)

- **Or:** Pre-processing (out of scope for the current lecture)
  - Geometry
  - Input data and conditions

- Post processing of results

- Inlet velocity profiles

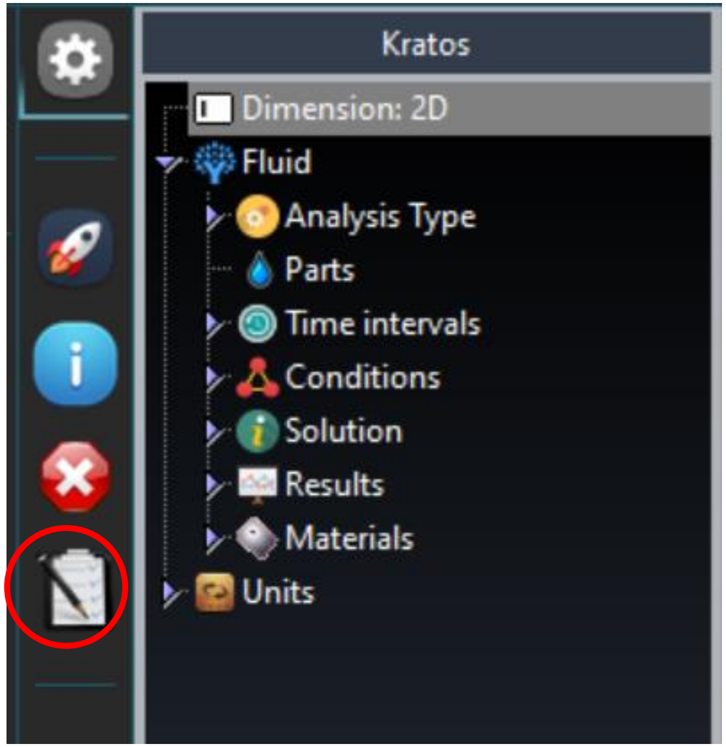
Disclaimer: This example serves the sole educational purpose of demonstrating how to setup a basic 2D CFD problem, run the simulation and do some postprocessing. For any real case in wind engineering a 3D setup should be adopted accompanied with detailed mesh and time step study.

Technical note: Tested on 14.12.2020, works with GiD 15.0.1 and the Kratos problemtype (8.1) on Windows 10 and Ubuntu 18/20 64 bit

Predefined example

- Load the Kratos problem type
  Data → Problem type → Kratos
  Fluid → Fluid → 2D

- Load the predefined example “Fluid High-rise 2D”



- In case you want to set up the problem from scratch on you own (out of the scope of the current lecture)
  → check out slides 5 to 9

- Proceed on page 10 → Check the time and solver settings
- Generate the mesh
- Run the calculation

Defining the Geometry

Geometry

- Create the geometry in the XY-plane using the following points to describe it:

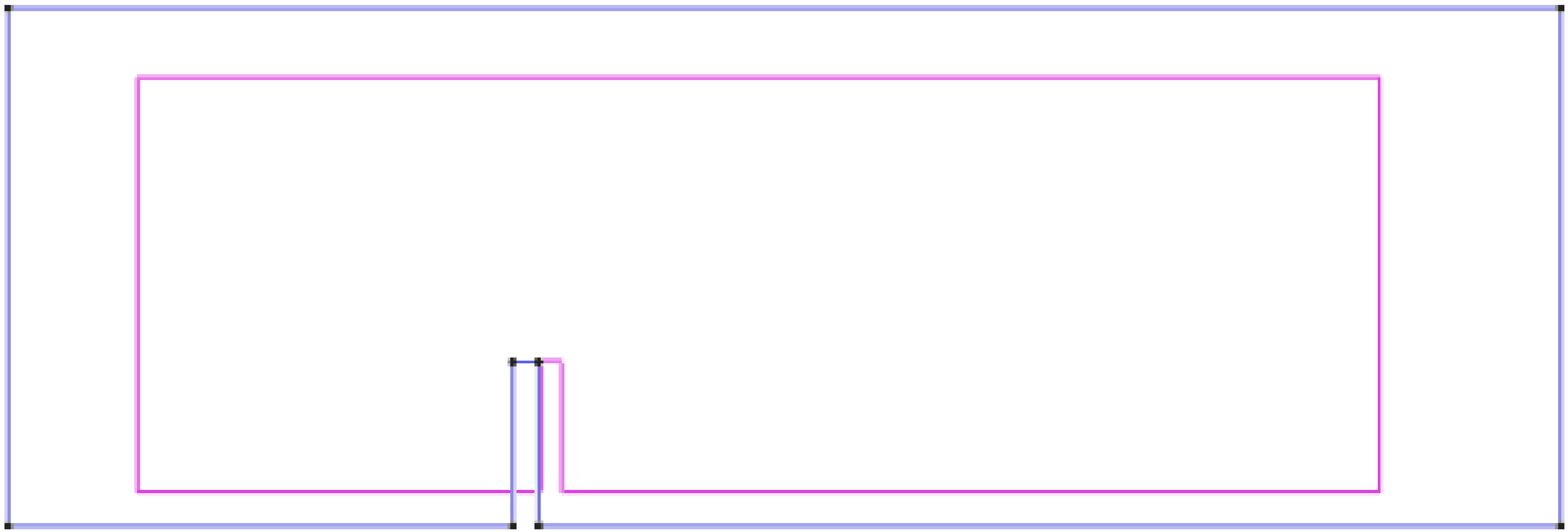
Structure X	Y	Z
15.0	0.0	0.0
15.0	190.0	0.0
-15.0	0.0	0.0
-15.0	190.0	0.0

Boundary X	Y	Z
-600.0	0.0	0.0
-600.0	600.0	0.0
1200.0	0.0	0.0
1200.0	600.0	0.0

- Create the points first, followed by the lines and the surface

Geometry

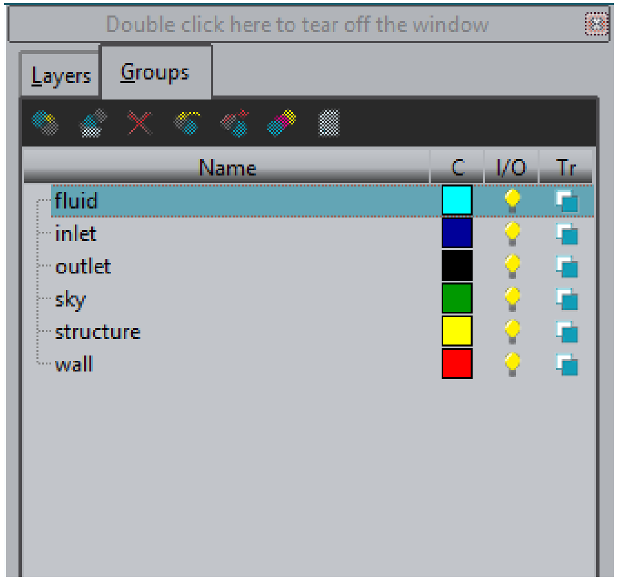
- The final geometry should look like this:

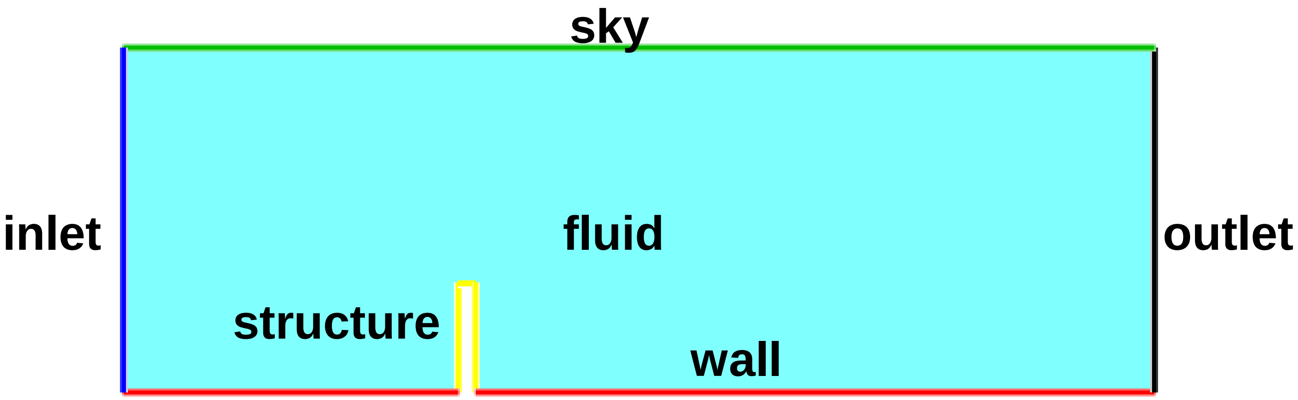


Problem Input

Define the entity groups

- *fluid* group
  - Select surface
- *inlet* group
  - Select left line
- *outlet* group
  - Select right line
- *structure* group
  - Select lines of the structure
- *wall* group
  - Select all bottom line
- *sky group*
  - Select the top line





Model properties & boundary conditions (1)

- Load the Kratos problem type
  Data → Problem type → Kratos
  Fluid → Fluid → 2D

- Analysis Type to *Fractional Step*

- Choose the *fluid* for *Group*

- Set the same *fluid properties and elements* as in the first tutorial

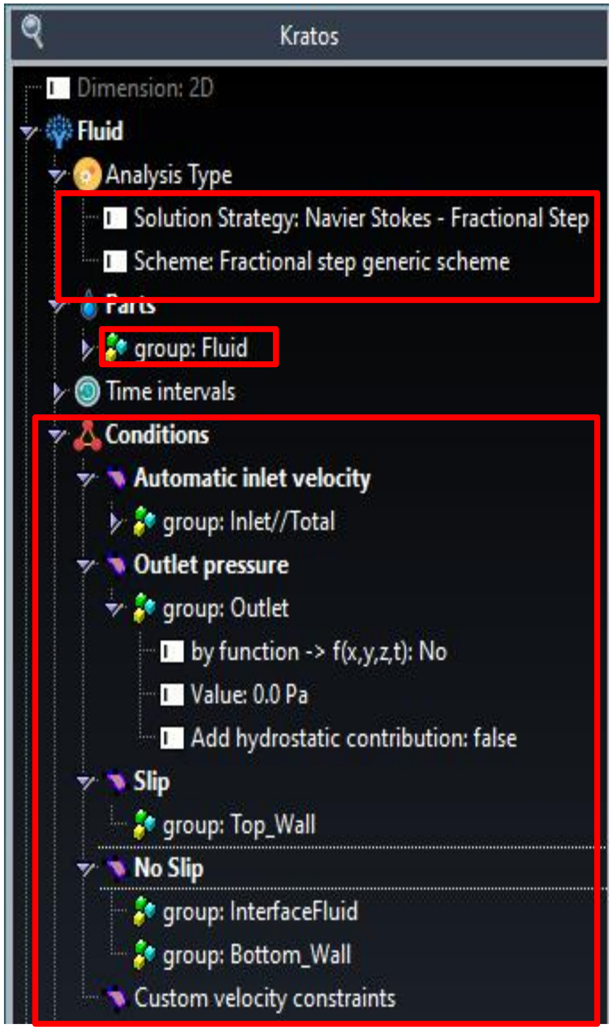
- In *Solution* Set *End time of the analysis to 100* and *Delta time to 0.05*

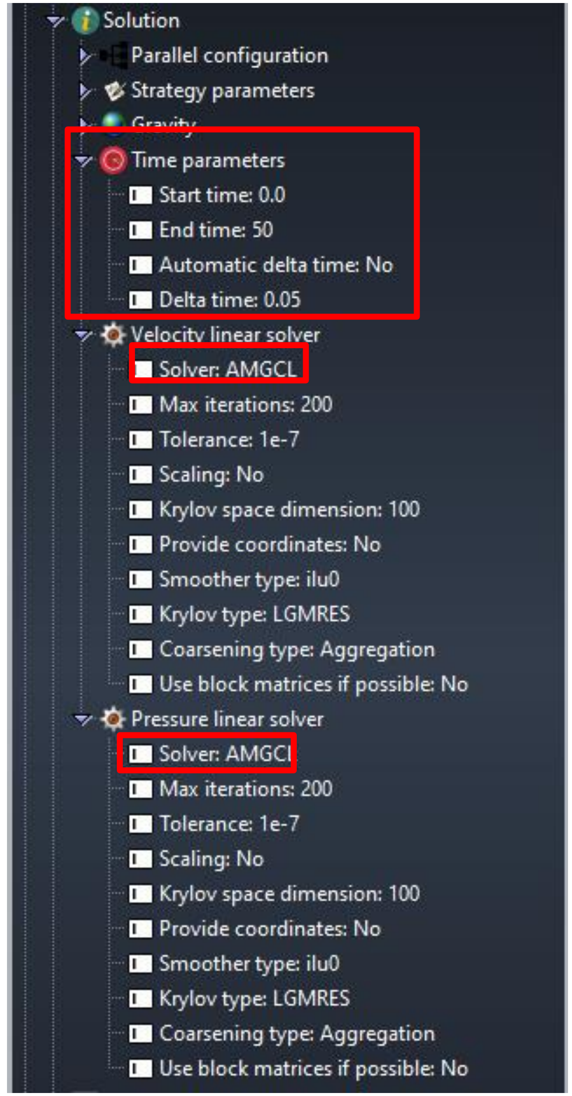
- Select an Iterative *Solver for velocity and pressure -> AMGCL*

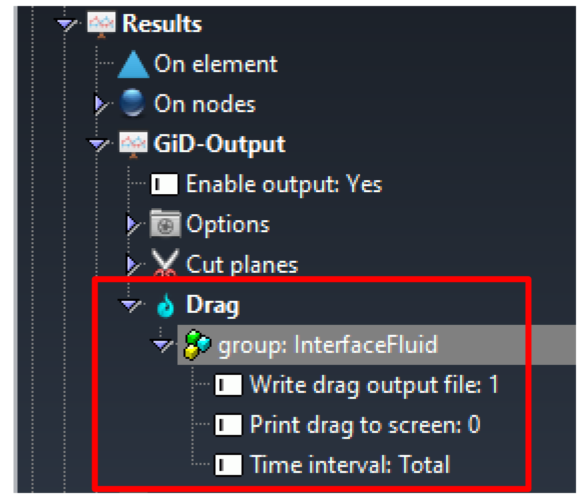
- Assign the following *boundary conditions*:
  - inlet: Inlet velocity (X = 25.0)
  - outlet: Outlet pressure 0.0
  - sky: Slip
  - wall: No-slip
  - structure: No-slip

- Select the *structure* entity group in order to compute the *drag* force

Model properties & boundary conditions (2)

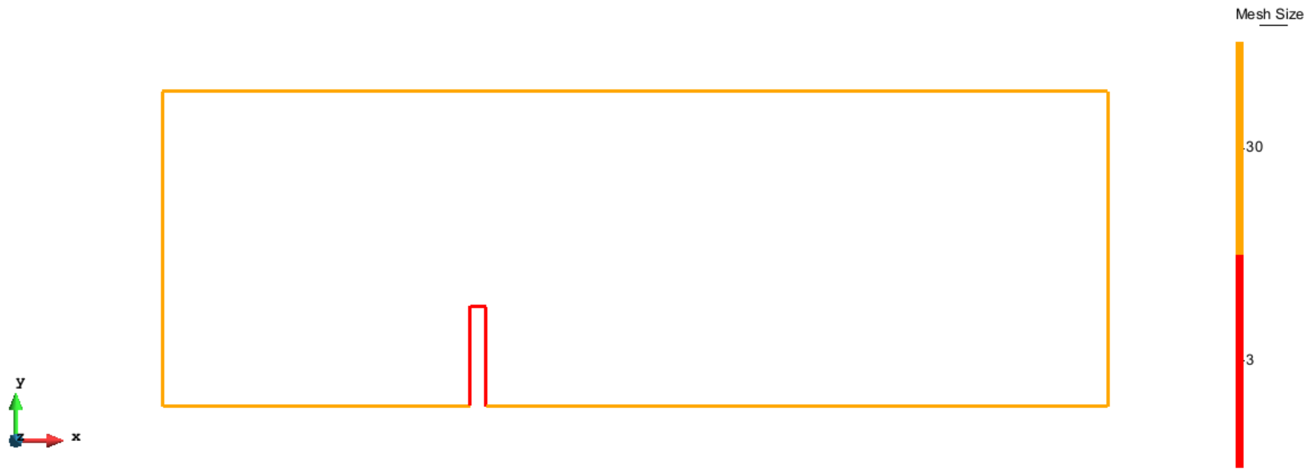




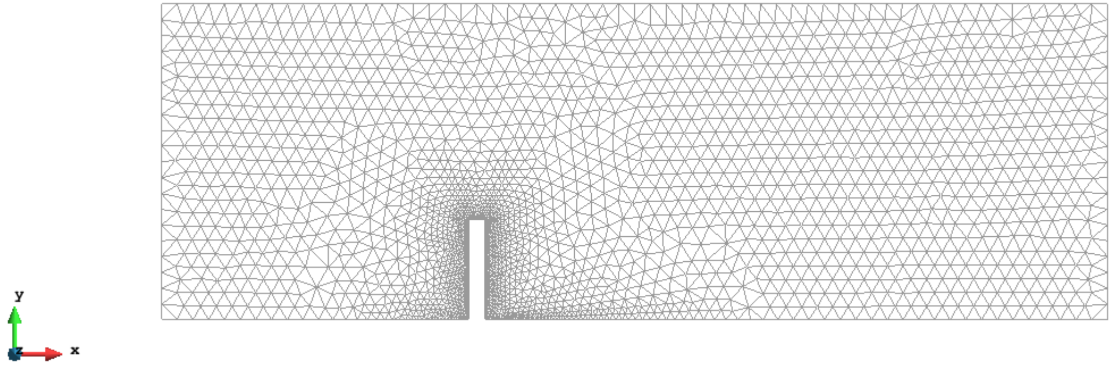


Mesh the domain

- Set the following sizes in the domain (*mesh → unstructured → assign sizes on lines*)



- Set *30 as element size* and mesh the domain
- The generated mesh should have ~ *2640 nodes* and ~ *4900 elements*



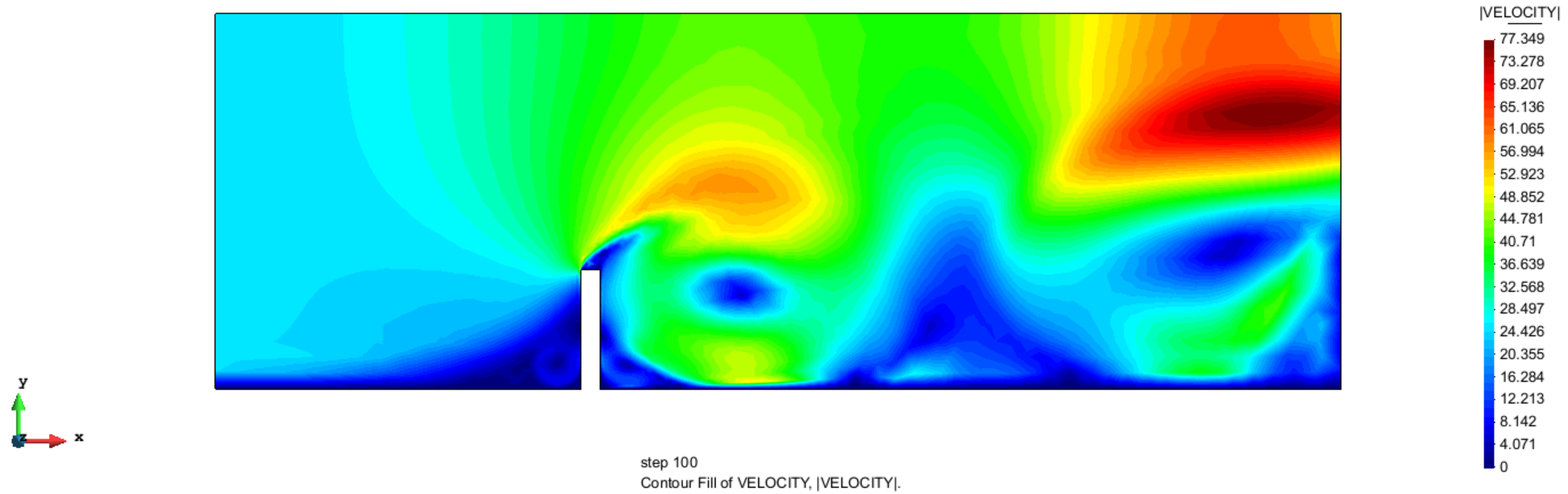
Solve the problem

- Save the model
- Run the analysis
- The input data will be checked for errors
- The calculation should not take more than 5 minutes
- Afterwards, switch to post process

Solution Postprocessing

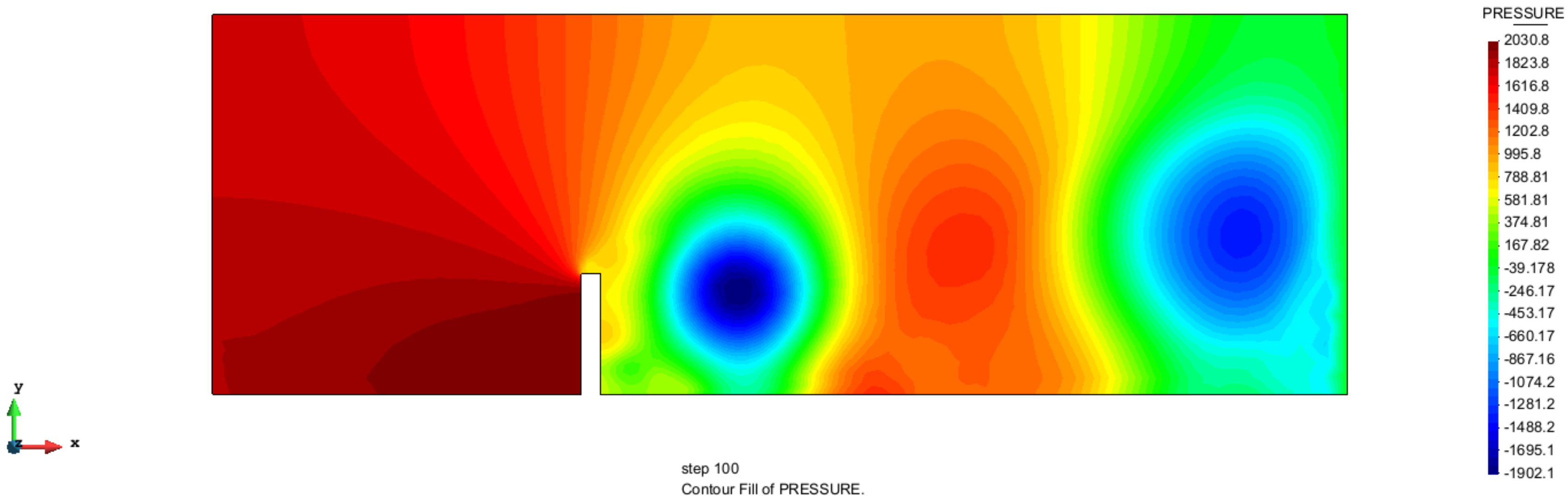
Post processing

- Play around with the results and the visualization
- Plot and animate the results for the velocity and the pressure and compare them
- Results for *magnitude of velocity* in the last timestep:



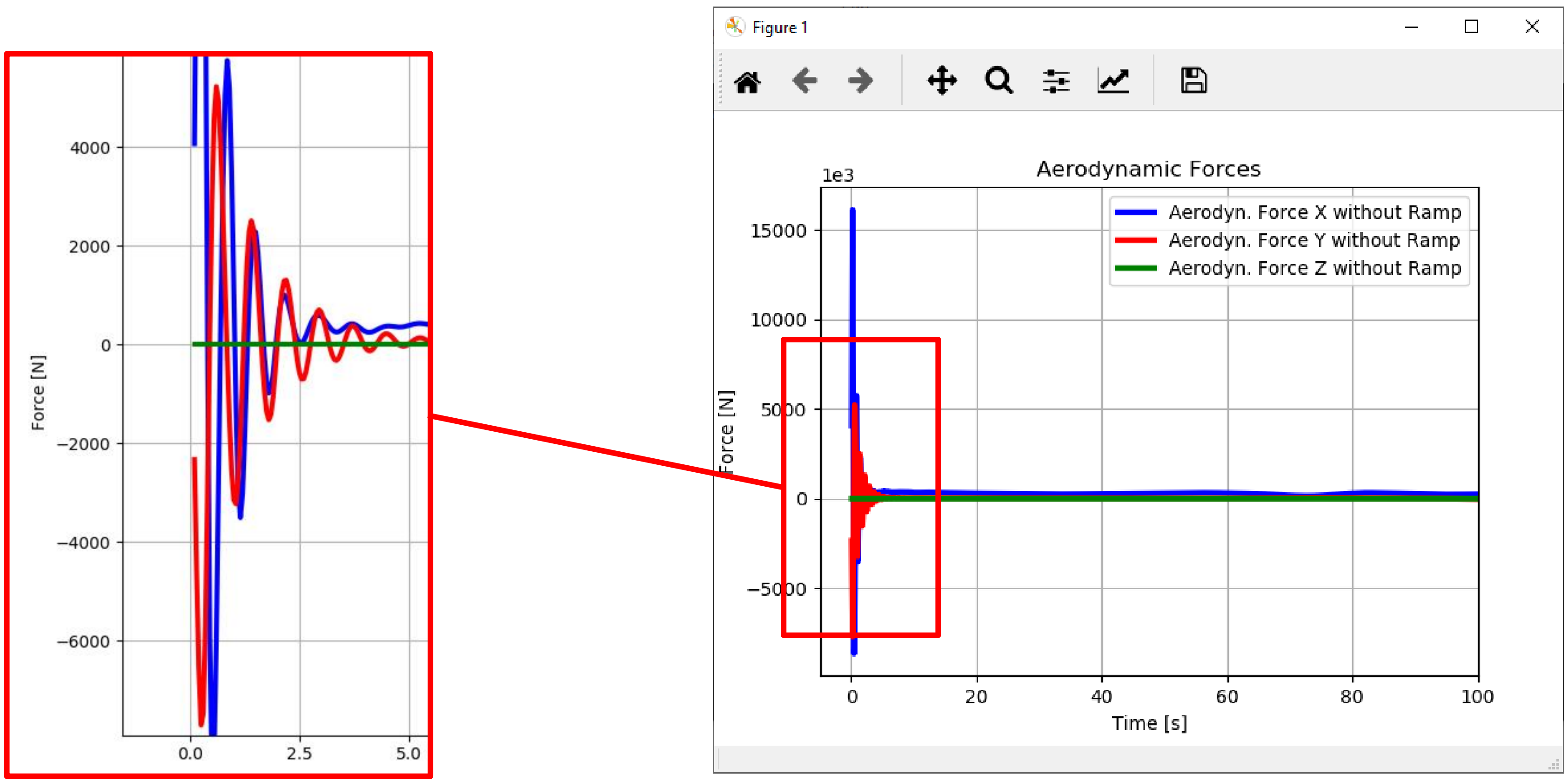
Post processing

- Results for *pressure* in the last timestep



Aerodynamic results

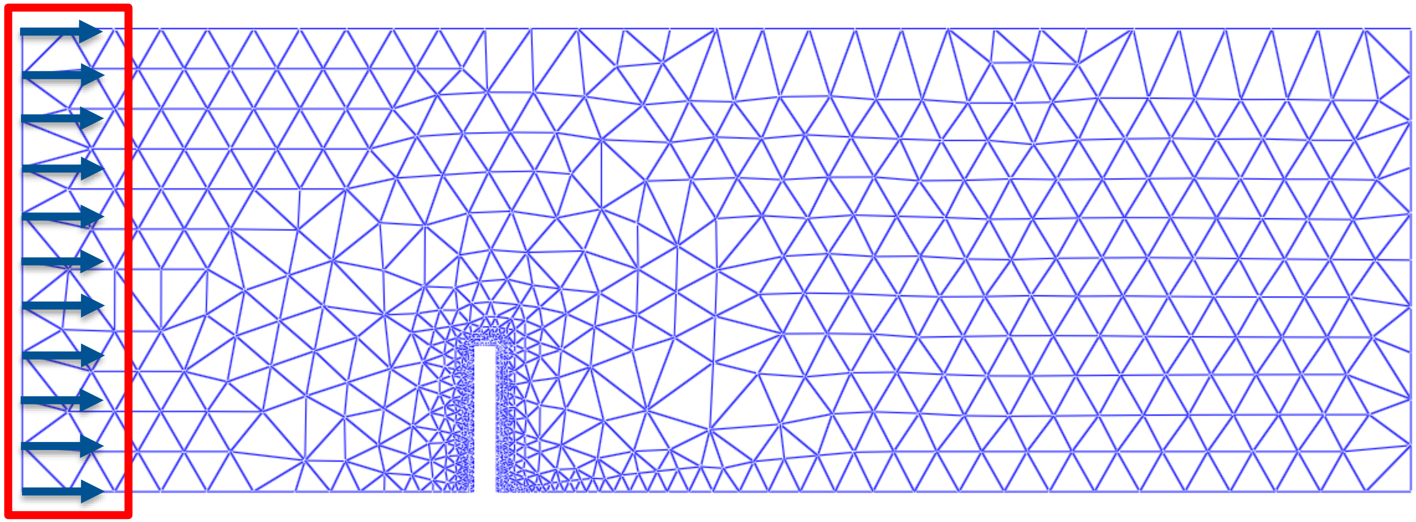
- In order to compute the aerodynamic load follow the same procedure as previous tutorial.
- Plot the results of the drag using the “plot_aerodynamic_force_results.py“ (available in *AdditionalFiles*)
- The results should look like this:



Inlet Velocity Profiles

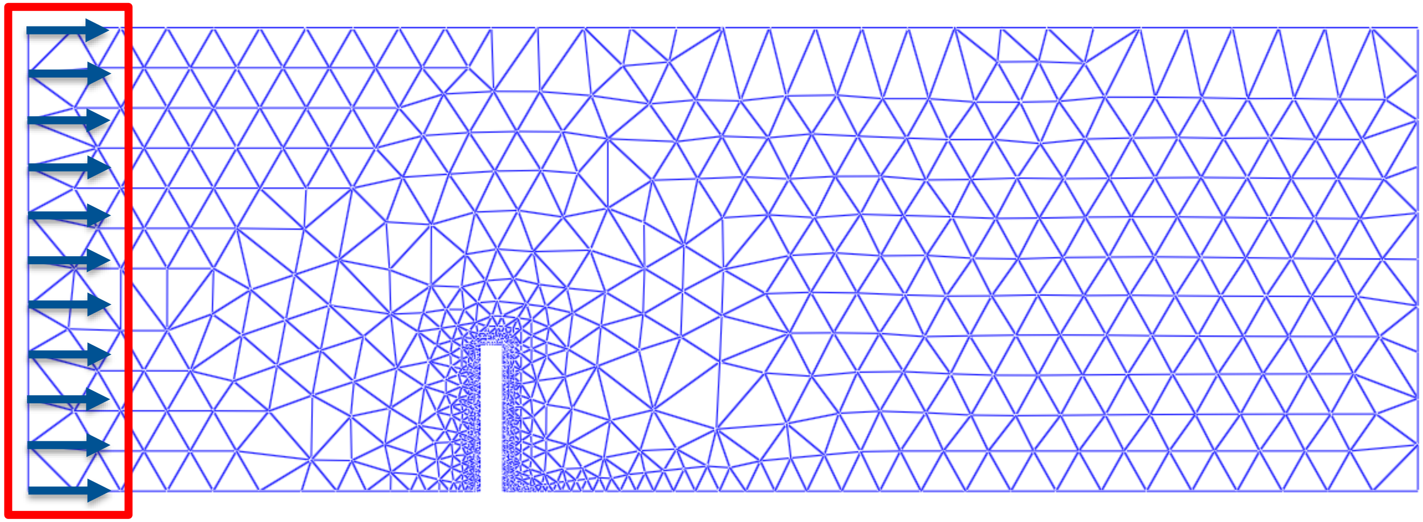
Inlet velocity profiles

- Remember Tutorial 2D CFD problem:
  Constant inlet velocity V_x = 25.0 has been applied

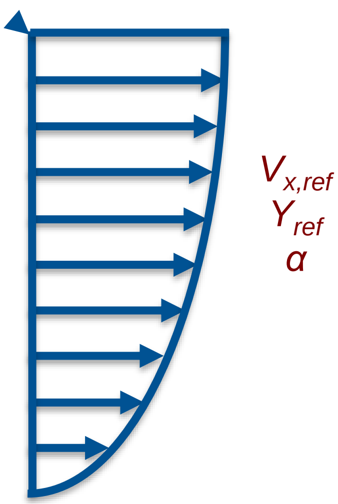


Custom boundary condition, i.e. self defined inlet, is necessary

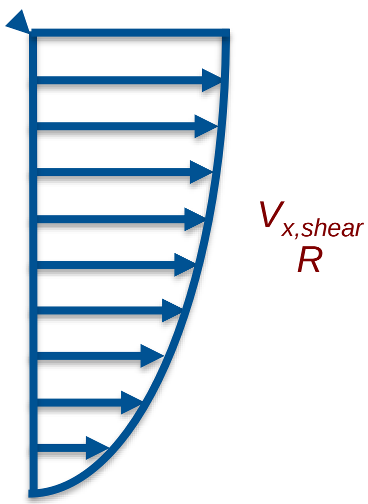
Inlet velocity profiles



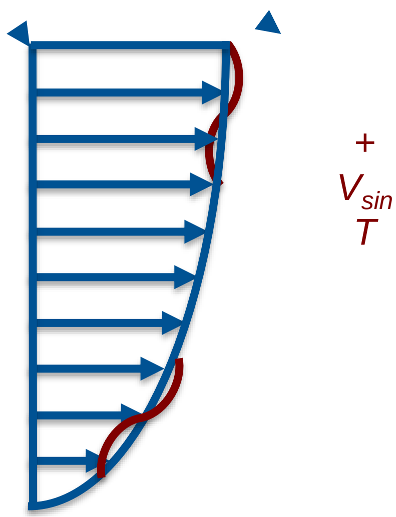
Proposed inlets:



Power law



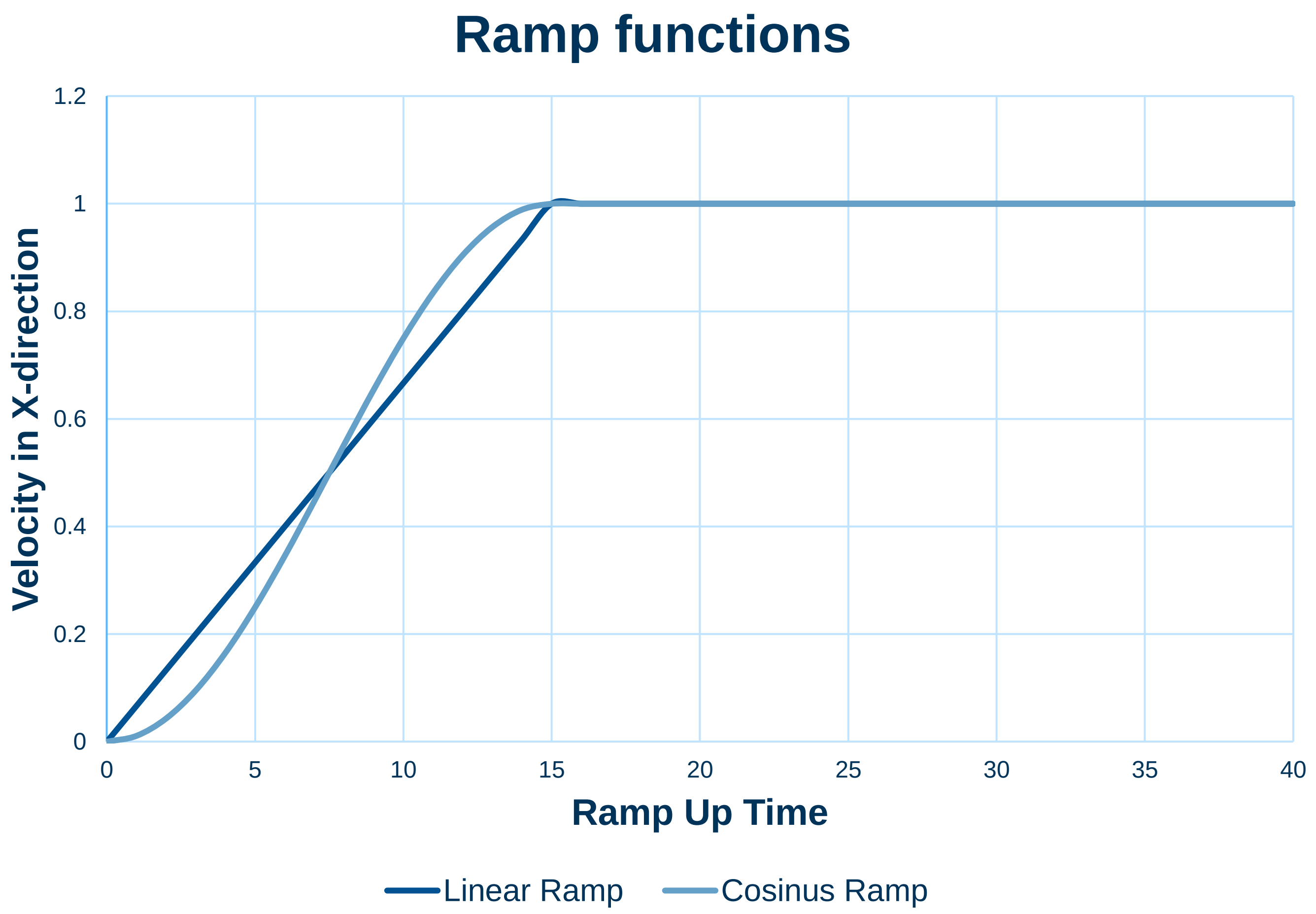
Logarithmic law



Convoluted log/power law

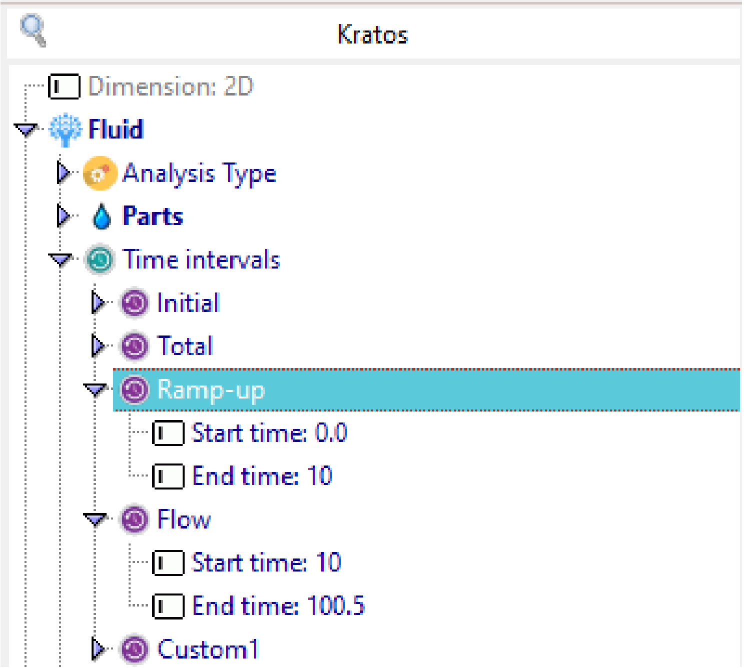
Inlet velocity profiles

All of the inlet velocity profiles can be ramped with respect to time



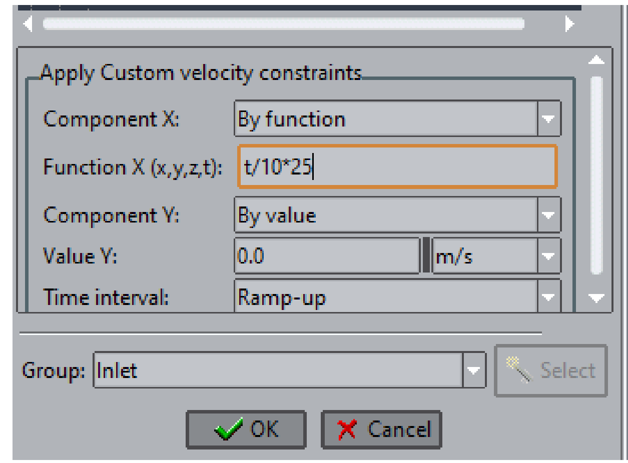
Inlet velocity profiles

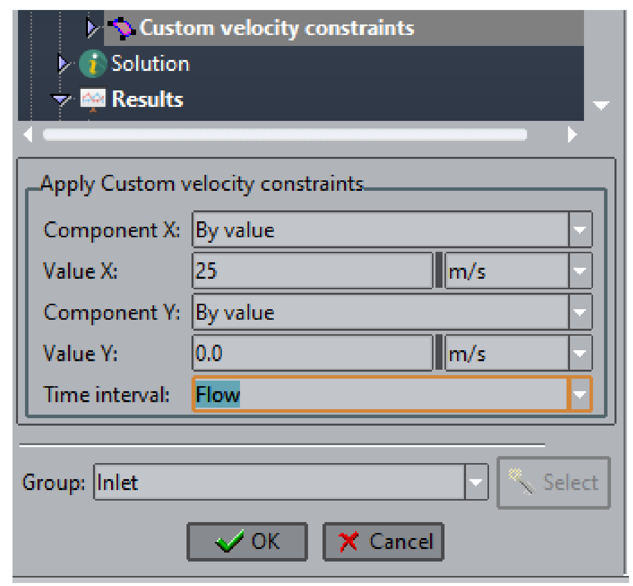
- Set up two *Time intervals*:
  - Copy *Custom1* (right click: copy)
  - Rename(F2) them in Ramp-up and Flow
  - Set time periods



Inlet velocity profiles

- Unassign the inlet group from the Automatic inlet velocity
- Set up two Custom velocity constraints, one for the ramp-up time the other on for the normal flow
- Change for the ramp-up:
  - By function to yes
  - X function: t/10 * 25
- Calculate the model again





Inlet velocity profiles

- Change the velocity function for both time intervals to an oscillating velocity profile → *by function: Yes*

- Add to the *X function* the oscillating sin() function
  - f(t) = Sin(0.02*t*2.0*3.41)

- Task: Use the velocity profile from tutorial 1
  Hint x^2 ⟶ **2