



Computational Thermo-Fluid Dynamics
Winter Semester 2020-2021

Laminar Channel Flow Over Obstacles

Philipp Müller¹, Anant Rajeev Talasikar²

Abstract

In the following report, the authors describe their derivation and implementation of a solution scheme for the laminar, in-compressible Navier-Stokes equations. The domain of interest is a 2D channel flow including an obstacle. The main goal is the prediction of vortex formation downstream of the obstacle. A standard incremental pressure-correction scheme is used for time discretization. The spatial domain is discretized by the means of a finite volume formulation. To prevent checker-boarding, a common issue on co-located grids, the Rhie and Chow interpolation is applied. First, a regular channel flow is examined to validate the flow solver. Next, a range of Reynolds numbers and obstacle shapes are simulated. With the current implementation, Reynolds numbers up to 250 can be achieved. The resulting flow field exhibits the expected formation of vortices behind the obstacle. Velocity and pressure contours are presented together with velocity vector and streamline visualizations. Important lessons learned and suggestions for future improvements of the solver are summarized.

Keywords

Navier-Stokes — Channel flow — Obstacle

¹Master Student, TUM, Munich, Germany

²Master Student, TU Berlin, Berlin, Germany

E-Mails: philippjulius.mueller@tum.de, anant.talasikar@campus.tu-berlin.de

Contents

1	Introduction	3
2	Numerical Methodology	4
2.1	Governing Equations	4
2.2	Temporal Discretization	5
2.3	Spatial Discretization	6
2.4	Assembly of the Linear System	8
2.5	Rhie and Chow Interpolation	10
2.6	Boundary and Initial Conditions	12
2.7	Program structure	13
2.8	Solver Algorithm Flowchart	14
3	Results and Discussion	16
3.1	Results	16
	Validation Channel Flow • Sinus Obstacle • Radial Basis Function Obstacle	
3.2	Futher Obstacle Shapes	31
3.3	Lessons Learned and Best Practices	33
3.4	Future Improvements	34
4	Conclusions	35
5	Appendix	36
	Acknowledgments	44
	References	44

1. Introduction

The configuration under study in this project is the fluid flow around an obstacle in a two-dimensional channel. This is a setup heavily studied in literature and therefore a good example for the development of an Navier-Stokes solver. In order to avoid the costly task of turbulence modeling, which itself introduces uncertainty to any flow simulation, the authors refer to laminar flow only.

The transition point between laminar and turbulent flow is determined by the problem setup and the Reynolds number, defined as $Re = \frac{u*L*\rho}{\mu}$ with the velocity u , the length scale L , the density ρ and the kinematic viscosity μ . The choice of the length scale depends on the problem under study. For a channel flow, the height h is an appropriate guess. In general, if $Re < 2300$ holds the flow is said to be laminar. Therefore, Reynolds numbers in the range of 10 – 300 are investigated in this project.

If the geometry of the channel is constant over its length l , sufficiently far from the outlet the flow field becomes independent of the variable x , meaning that $\frac{\partial u}{\partial x} = 0$. This state is called fully developed and for planar channel flow, an analytical solution for the velocity profile of fully developed flow exists. This solution is described by the Hagen–Poiseuille equation [10], which predicts a parabolic profile of the form $u(y) = \frac{G}{2\mu}y(h - y)$. The parameter G depends on the pressure gradient along the channel. Figure 1 shows the general parabolic shape of the horizontal velocity profile for a fully developed, planar channel as described by Hagen–Poiseuille. This analytical solution will later be used to validate the solver. By introducing an obstacle into the flow, the velocity and pressure field is disturbed.

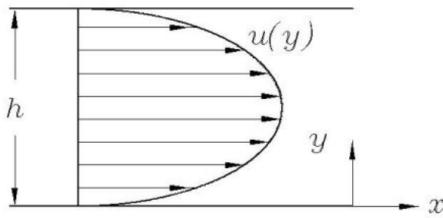


Figure 1. General shape of a laminar velocity profile in a planar channel described by Hagen–Poiseuille [9].

Behind the obstacle, if the Reynolds number is sufficiently large (but still laminar), a recirculation zone forms. Vortices occur and are transported by convection along the channel. They dissipate over time due to the viscosity of the fluid. Figure 2 shows the horizontal velocity u flow field in a planar channel with a half-circle as obstacle in the lower half of the channel. Behind the obstacle, the recirculation zone becomes visible. The main goal of this project is to recreate such flow field behind an obstacle. Channel flow over obstacles is a problem formulation that is relevant in a lot of technical application. Examples range from flow over cooling ribs in a heat exchanger to atmospherical simulations of wind over geographic landscapes.

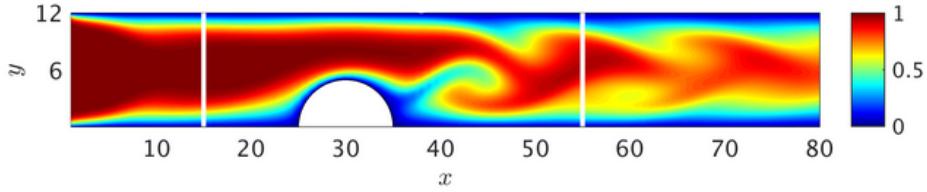


Figure 2. Horizontal velocity u contour of a laminar channel flow with a circular obstacle on the bottom half [5].

In section 2, the numerical methodology of the implemented solver is described. Starting from the very basic governing equations, a temporal and spatial discretization scheme is developed. Additionally, the required boundary conditions are explained. In the following chapter 3, the results for the plain channel flow as well as the obstacle flow are presented. Apart from velocity and pressure plots, important lessons learned are summarized. Finally, in section 4, a conclusion of the project is given and the fulfillment of the main goal is evaluated.

2. Numerical Methodology

2.1 Governing Equations

Fluid flow is described by a set of partial differential equations, the so-called Navier-Stokes equations. For simplicity, we assume in-compressible flow, meaning that the density is constant throughout the domain. The Navier-Stokes equation then take the following form [1]:

$$\nabla \cdot \underline{u} = 0 \quad (1)$$

$$\frac{\partial \underline{u}}{\partial t} + (\underline{u} \cdot \nabla) \underline{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \underline{u} \quad (2)$$

Equation (1) is also called as the continuity equation, describing mass conservation while the equation (2) is the momentum equation. The vector \underline{u} describes the velocity, p the pressure, ρ the density and $\nu = \frac{\mu}{\rho}$ the kinematic viscosity. In the following, we will simplify the problem to be two-dimensional in a cartesian coordinate system. Therefore, the continuity equation can be rewritten as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3)$$

Similarly, the momentum equation is decomposed into:

$$\rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = - \frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (4)$$

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = - \frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (5)$$

In order to solve those equations, temporal as well as spatial discretization is necessary, which will be explained in the next sections.

2.2 Temporal Discretization

In equation (2), a temporal discretization is applied, with the superscript n denoting values at time step n , while $n + 1$ denotes values calculated one step further. Additionally, an intermediate velocity u^* is introduced as follows:

$$\frac{u^{n+1} - \underline{u}^* + \underline{u}^* - \underline{u}^n}{\Delta t} = -\underline{u} \cdot \nabla \underline{u} - \frac{1}{\rho} p + \nu \nabla^2 \underline{u} \quad (6)$$

We split equation (6) up into two parts:

$$\frac{\underline{u}^* - \underline{u}^n}{\Delta t} = -\underline{u}^* \cdot \nabla \underline{u}^* + \nu \nabla^2 \underline{u}^* \quad (7)$$

$$\frac{u^{n+1} - \underline{u}^*}{\Delta t} = -\frac{1}{\rho} \nabla p^{n+1} \quad (8)$$

By rearranging (7), the intermediate velocity field can be calculated from:

$$\underline{u}^* = \underline{u}^n + \Delta t (-\underline{u}^* \cdot \nabla \underline{u}^* + \nu \nabla^2 \underline{u}^*) \quad (9)$$

Equation (8) is multiplied with the divergence operator ∇ . Since we now equation (1) that $\nabla \cdot \underline{u}^{n+1} = 0$ we find a Poisson equation for the pressure p^{n+1} :

$$\nabla^2 p^{n+1} = \frac{\rho}{\Delta t} \nabla \underline{u}^* \quad (10)$$

After solving equation (10) for the pressure in the next time step p^{n+1} we obtain the velocity \underline{u}^{n+1} from equation (8):

$$\underline{u}^{n+1} = \underline{u}^* - \frac{\Delta t}{\rho} \nabla p^{n+1} \quad (11)$$

This predictor-corrector method is called the non-incremental pressure-correction scheme and was first proposed by Chorin [2]. Further improvements were made by Goda [3] by

adding the old pressure p^n to the predictor step in equation (7):

$$\frac{\underline{u}^* - \underline{u}^n}{\Delta t} = -\underline{u}^* \cdot \nabla \underline{u}^* + \nu \nabla^2 \underline{u}^* - \frac{1}{\rho} \nabla p^n \quad (12)$$

The corrector step (8) is adapted accordingly:

$$\frac{\underline{u}^{n+1} - \underline{u}^*}{\Delta t} = -\frac{1}{\rho} \nabla p_c \quad (13)$$

The term $p_c = p^{n+1} - p^n$ is the pressure correction. It can be obtained by applying the divergence as for the original scheme:

$$\nabla^2 p_c = \frac{\rho}{\Delta t} \nabla \underline{u}^* \quad (14)$$

After obtaining the velocity \underline{u}^{n+1} and the pressure correction p_c , the pressure is updated via $p^{n+1} = p^n + p_c$. This predictor-corrector method based on equations (12) and (14) is called the standard incremental pressure-correction scheme. It is widely used in flow simulations and is the method of choice for this project.

2.3 Spatial Discretization

In computational flow simulation, there are two main ways of spatial discretization: co-located grids and staggered grids. For the co-located grid, the pressure and velocity variables are stored in each node. In contrast to that, for staggered grids there is one mesh of nodes for pressure and another mesh for the velocity information, shifted by half a mesh spacing compared to the pressure. Figure 3 shows a schematic for a co-located and a staggered grid. Circles denote positions where the pressure is stored, arrows indicated positions where the horizontal and vertical velocity are stored respectively.

The advantage of a staggered grid is that many derivatives which are needed for the Navier-Stokes equations can be calculated from central difference schemes. On the other hand, for co-located grids interpolation schemes are required to get the variable values at the boundaries between cells. Nonetheless, a co-located grid arrangement is implemented in this project since mesh generation is much easier. Co-located grids suffer from the so-called checkerboard effect and require a suitable mitigation strategy, which will be discussed in chapter 2.5.

In the course of this project, a finite volume (FV) method is implemented for spatial discretization. The main reason to prefer FV over for example finite differences is that FV works well for non-cartesian grids and is therefore optimal for complex geometries. Due to the integration of the obstacles, we expect our mesh to be skewed, which would lead to errors when working with finite differences.

In general, the finite volume method works by integrating the differential equation

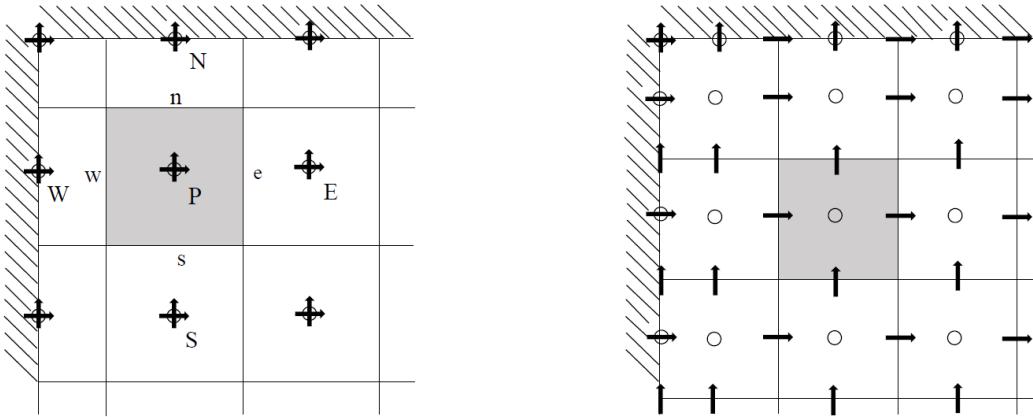


Figure 3. Co-located grid (right) and staggered grid (left). Circles indicate pressure nodes and arrows velocity nodes [1].

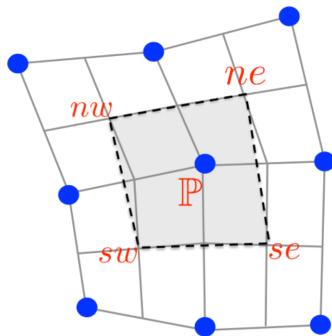


Figure 4. General layout of a finite volume control volume [8].

under consideration over the surface of a control volume around a node P. Figure 4 shows a schematic of this control volume. The lowercase corners (ne, se, nw, sw) are intermediate points defining the edges of the control volumes. The blue dots around P are neighboring nodes. By applying the divergence theorem, the surface integral is transformed to a line integral over the four sides of the control volume. It should be noted that cells are by no means required to be rectangular and can be any shape. Next, the midpoint integration rule is introduced to calculate the line integral based on the distances between the control volume edges and the mid-point value of the variable of interest. The velocity gradient $\frac{\partial u}{\partial x}$ at node P can for example be approximated by the FV method as [1]:

$$\left(\frac{\partial u}{\partial x}\right)_P \approx \frac{1}{S_P} (\Delta y_{sw}^{se} u_s + \Delta y_{se}^{ne} u_e + \Delta y_{ne}^{nw} u_n + \Delta y_{nw}^{sw} u_w) \quad (15)$$

In equation (15), S_P is the area of the control volume around P, Δy geometric properties resulting from the mesh and u_s, u_e etc. the intermediate velocity values at the boundaries. They can be obtained for example by simple averaging between P and its neighboring nodes such as $u_s = \frac{u_p + u_s}{2}$. If the mesh is uniform and rectangular, the method collapses to a second order finite difference scheme.

The derivation of the Laplacian operator ∇^2 based on finite volumes is more complex and will therefore not be described in this report. Instead the interested reader is referred to the lecture script for the course *Computational Thermo-Fluid Dynamics* by Camilo Silva [8].

2.4 Assembly of the Linear System

After temporal and spatial discretization, the Navier-Stokes equations are now linearized in order to assemble a linear system of equations using matrix operators. First the convective terms are linearized by a Crank-Nicolson formulation. As an example we take [4]:

$$u_* \frac{\partial u^*}{\partial x} = \frac{\partial (u^*)^2}{\partial x} = \frac{1}{2} \frac{\partial}{\partial x} ((u^*)^2 + (u^n)^2) \quad (16)$$

After Taylor series expansion we obtain:

$$u^* \frac{\partial u^*}{\partial x} = u^* \frac{\partial u^n}{\partial x} \quad (17)$$

With this result, we rewrite the momentum equations (4) and (5) as:

$$\frac{u^* - u^n}{\Delta t} + u^* \frac{\partial u^n}{\partial x} + v^* \frac{\partial u^n}{\partial y} = - \frac{1}{\rho} \frac{\partial p^n}{\partial x} + \nu \left(\frac{\partial^2 u^*}{\partial x^2} + \frac{\partial^2 u^*}{\partial y^2} \right) \quad (18)$$

$$\frac{v^* - v^n}{\Delta t} + u^* \frac{\partial v^n}{\partial x} + v^* \frac{\partial v^n}{\partial y} = - \frac{1}{\rho} \frac{\partial p^n}{\partial y} + \nu \left(\frac{\partial^2 v^*}{\partial x^2} + \frac{\partial^2 v^*}{\partial y^2} \right) \quad (19)$$

After the finite volume discretization the analytical derivative operators are replaced by discrete matrix operators:

$$\frac{\partial}{\partial x} \rightarrow Dx \quad (20)$$

$$\frac{\partial}{\partial y} \rightarrow Dy \quad (21)$$

$$\nabla^2 \rightarrow L \quad (22)$$

With the help of those operators, we write down the momentum equations in matrix form, which is the form actually implemented in the code for this project:

$$\underbrace{\left[\frac{1}{\Delta t} I + Dx u^n - \nu L \right]}_{A_U^U} u^* + \underbrace{\left[Dy u^n \right]}_{A_U^V} v^* = \underbrace{\frac{u^n}{\Delta t} - \frac{1}{\rho} D x p^n}_{b_u} \quad (23)$$

$$\underbrace{\left[D x v^n \right]}_{A_V^U} u^* + \underbrace{\left[\frac{1}{\Delta t} I + D x v^n - \nu L \right]}_{A_V^V} v^* = \underbrace{\frac{v^n}{\Delta t} - \frac{1}{\rho} D y p^n}_{b_v} \quad (24)$$

In matrix notation, the linear system can be written in the short form:

$$\begin{bmatrix} A_U^U & A_U^V \\ A_V^U & A_V^V \end{bmatrix} \begin{bmatrix} u^* \\ v^* \end{bmatrix} = \begin{bmatrix} b_u \\ b_v \end{bmatrix} \quad (25)$$

The linear system (25) is solved to obtain the intermediate velocities u^* and v^* (predictor step). Next, the pressure correction term is calculated from equation (14):

$$L p_c = \frac{\rho}{\Delta t} (D x u^* + D y v^*) \quad (26)$$

After solving for the pressure correction term, the velocities in the next time step can be calculated from equation (13):

$$\begin{bmatrix} u^{n+1} \\ v^{n+1} \end{bmatrix} = \begin{bmatrix} u^* \\ v^* \end{bmatrix} - \frac{\Delta t}{\rho} \begin{bmatrix} D x p_c \\ D y p_c \end{bmatrix} \quad (27)$$

The pressure in the next time step is found as:

$$p^{n+1} = p^n + p_c \quad (28)$$

The equations (25), (26), (27) and (28) are solved consecutively for each time-step n until a residual threshold is reached. The residual is hereby defined as the norm of the error of the

discretized continuity equation (1). This error is also called divergence velocity:

$$res = \left\| \underbrace{\left(Dx u^n + Dy v^n \right)}_{\text{Divergence velocity}} \right\| \quad (29)$$

2.5 Rhee and Chow Interpolation

Co-located grids, such as the one used in this project, suffer from an issue called the checkerboard effect. In order to explain its cause, a finite difference scheme is because of its simplicity. However, the checkerboard problem is present in finite volume discretization as well. We start from the pressure correction equation (14) [1]:

$$\frac{\partial}{\partial x} \left(\frac{\partial p_c}{\partial x} \right) = \underbrace{\frac{\rho}{\Delta t} \frac{\partial \underline{u}^*}{\partial x}}_{Q_p} \quad (30)$$

The right hand side will be abbreviated by Q_p for now since it is not of interest. On the left hand side, the outer difference operator is replaced by a central finite difference scheme, leading to:

$$\frac{\left(\frac{\partial p_c}{\partial x} \right)_E - \left(\frac{\partial p_c}{\partial x} \right)_W}{2\Delta x} + \frac{\left(\frac{\partial p_c}{\partial y} \right)_N - \left(\frac{\partial p_c}{\partial y} \right)_S}{2\Delta y} = Q_p \quad (31)$$

The indices N, S, E and W describe the four neighboring nodes of the current node P , where p_c should be evaluated. The pressure gradients can be approximated by another central finite difference scheme, leading to:

$$\frac{\frac{p_{c,EE} - p_{c,P}}{2\Delta x} - \frac{p_{c,P} - p_{c,WW}}{2\Delta x}}{2\Delta x} + \frac{\frac{p_{c,NN} - p_{c,P}}{2\Delta y} - \frac{p_{c,P} - p_{c,SS}}{2\Delta y}}{2\Delta y} = Q_p \quad (32)$$

The resulting expression (32) shows a coupling only between nodes that are $2\Delta x$ apart from each other, meaning that the pressure at the direct neighboring nodes of P does not influence the pressure correction $p_{c,P}$. In other words, the Poisson equation is solved on grid twice as coarse as the original mesh. A pressure distribution which is oscillating with a wavelength $2\Delta x$ would be a valid solution to equation (32). Such a pressure pattern may not be physical and therefore is a wrong representation the actual flow-field. The pressure field shown in 5 is a valid solution as well and illustrates why this issue is often called the checkerboard effect.

In contrast to co-located grids, staggered grids do not suffer from the checkerboard effect. However, since a co-located grid was desired for this project, a method to mitigate checkerboarding has to be implemented. A popular solution that is still widely used is the interpolation scheme of Rhie and Chow [6]. It prevents checkerboarding by using a $1\Delta x$

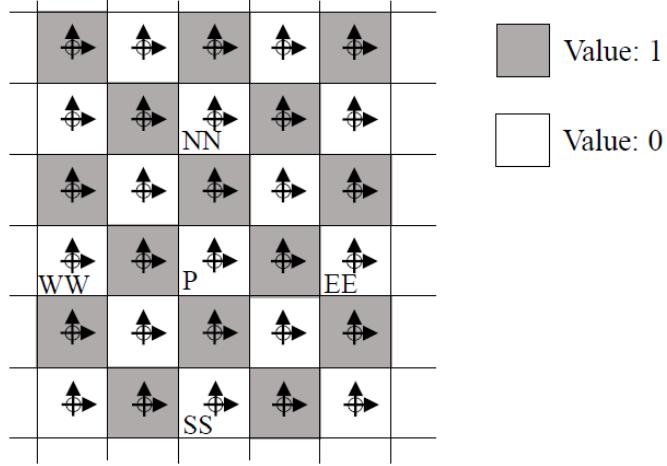


Figure 5. Co-located grid showing the checkerboard effect [1].

scheme for the pressure. Rhie and Chow defined the following interpolation equation for the velocity u_e^* at the east boundary of the control volume:

$$u_{e,RC}^* = \overline{(u^*)}_e - \Delta\Omega_e \overline{\left(\frac{1}{A_P^U}\right)}_e \left[\left(\frac{\partial p^n}{\partial x} \right)_e - \overline{\left(\frac{\partial p^n}{\partial x} \right)}_e \right] \quad (33)$$

In equation (33), overbars indicate interpolated values. They are defined as:

$$\overline{(u^*)}_e = \frac{1}{2} (u_P^* + u_E^*) \quad (34)$$

$$\overline{\left(\frac{\partial p^n}{\partial x} \right)}_e = \frac{1}{2} \left(\left(\frac{\partial p}{\partial x} \right)_P + \left(\frac{\partial p}{\partial x} \right)_E \right) \quad (35)$$

Furthermore, $\Delta\Omega_e$ is the volume of the cell around e . The term A_P^U is the factor coming from the momentum equation discretization. It can be obtained by splitting up the matrices as follows:

$$A_U^U = A_P^U + \sum_{nb} A_{nb}^U \quad (36)$$

Subscript P stands for the main node and the subscript nb its neighbors. For the Rhie and Chow method, this factor is interpolated similar to the other variables as:

$$\overline{(A_P^U)}_e = \frac{1}{2} ((A_P^U)_P + (A_P^U)_E) \quad (37)$$

Plugging all those definitions into Equation (33) finally gives the Rhie Chow interpolation for the velocity u_e^* . This interpolated velocity at the cell face is thus modified by the term

in square brackets. This term is the difference between the pressure gradient evaluated at the cell faces and the interpolated pressure gradient at the same cell face. If the pressure solution is smooth, this term is therefore going to zero. If pressure oscillations are present, this term is non-zero and the velocity modified accordingly to smooth out these oscillations. The velocity interpolation for the other three faces w , n and s is derived similarly. The same approach is used for the vertical velocity v^* , leading to:

$$v_{e,RC}^* = \overline{(v^*)}_e - \Delta\Omega_e \left(\frac{1}{A_P^V} \right)_e \left[\left(\frac{\partial p^n}{\partial y} \right)_e - \overline{\left(\frac{\partial p^n}{\partial y} \right)}_e \right] \quad (38)$$

It uses similar interpolations as the horizontal velocity:

$$\overline{(v^*)}_e = \frac{1}{2} (v_P^* + v_E^*) \quad (39)$$

$$\overline{\left(\frac{\partial p^n}{\partial y} \right)}_e = \frac{1}{2} \left(\left(\frac{\partial p}{\partial y} \right)_P + \left(\frac{\partial p}{\partial y} \right)_E \right) \quad (40)$$

$$\overline{(A_P^V)}_e = \frac{1}{2} ((A_P^V)_P + (A_P^V)_E) \quad (41)$$

The A_P^V stems from the decomposition:

$$A_V^V = A_P^V + \sum_{nb} A_{nb}^V \quad (42)$$

After obtaining u_{RC}^* and v_{RC}^* at all cell faces, these values are used to derive Dxu^* and Dyv^* . These velocity derivates are part of the Poisson equation (26) for the pressure correction. By introducing the Rhie and Chow interpolation, the problem of checkerboarding is prevented in this project. This enables the solver to find a solution even for high Reynolds numbers without exciting pressure oscillation that might make the solution unstable.

2.6 Boundary and Initial Conditions

In the following, the required boundary condition so solve the flow field are outlined. Table 1 summarizes the imposed boundary conditions. At the inlet, the velocities u and v are specified as Dirichlet boundary conditions. The horizontal velocity u is adapted to obtain different Reynolds numbers. Since no information about the inlet pressure is known a priori, a Neumann boundary condition, setting the pressure gradient to zero, is applied. At the walls, the no-slip condition holds, which is implemented as a Dirichlet condition with $u = v = 0$ m/s. As know from boundary layer theory [7], inside the boundary layer $\frac{\partial p}{\partial y} = 0$ holds, which is used to calculate the pressure on the boundary nodes at the wall. The obstacle surface is defined as a wall as well. Finally, at the outlet a Neumann gradient zero boundary condition is defined for the horizontal velocity u . Additionally, since we assume fully developed flow at the end of the domain, $v = 0$ is enforced via Dirichlet. For

the pressure, a Dirichlet condition is used as well, setting the outlet pressure to 0 Pa .

Table 1. Boundary conditions for the flow domain under study.

Variable	u	v	p
Inlet	Dirichlet	Dirichlet	Neumann
Outlet	Neumann	Neumann	Dirichlet
North Wall	Dirichlet	Dirichlet	Neumann
South Wall	Dirichlet	Dirichlet	Neumann

To start the solver, initial values for the fields u, v and p are necessary. Since no information about those fields is available before the first run, they are initialized with zero throughout the domain. However, if large Reynolds numbers are specified, the solver uses a *crescendo* approach, meaning it starts with a Reynolds number of $Re = 50$, calculates the solution, and uses those results to initialize the next step at $Re = Re + 25$ until the final, desired Reynolds number is reached. This method was found to improve the convergence behavior significantly for high Reynolds numbers.

2.7 Program structure

The program was structured based on object oriented programming principles and modularization. This enables fast adaption and adjustment. The structure is visualized in figure 6. The script *Init.m* initializes all the mesh (with or without obstacle), the time variables (iterations, tolerances) as well as the boundary conditions. It is instantiated in the script *main.m*, where all its data can be accessed. This data is passed on as arguments to different functions such as *buildDxDy.m*, which returns the Dx and Dy matrix operators together with geometric properties that required for the Rhie-Chow interpolation.

Further, all data and boundary conditions are passed to the script *buildL.m* which builds the Laplacian operator matrix along with the corresponding right hand side. All the input data and matrix operators are then to script called as *solveNS.m* where the Navier Stokes equations are solved. The obtained solution for velocity is smoothed using the Rhie-Chow interpolation. The obtained solution is returned to *main.m* and is passed on to *visualize.m*, where the plots can be created.

Initially, for the purpose of simplicity, the code was developed serially. However, using the serial approach with nested for loops for the solver as well as the Rhie-Chow interpolation, the code took on average about 10.97 seconds per iteration for mesh of 250×80 nodes. It would have consumed a lot of time to use this implementation for performing numerical experiments such as obstacle inclusion and parametric studies.

Therefore, it was decided to vectorize the code by making use of MATLAB logical arrays. These arrays have *true* values everywhere except on the regions which have been assigned as Dirichlet boundary. This has two advantages: firstly, it eliminates the need of updating the boundary values in every iteration of the time loop (which are lost because of the pressure

velocity coupling). Secondly, it reduces the size of the arrays involved when computing the solution. After the vectorization approach was implemented for the solver and the Rhee-Chow interpolation, each iteration in the time loop took about 0.179 seconds on average on the same 250x80 mesh, thus yielding a speed up factor of about 61.28.

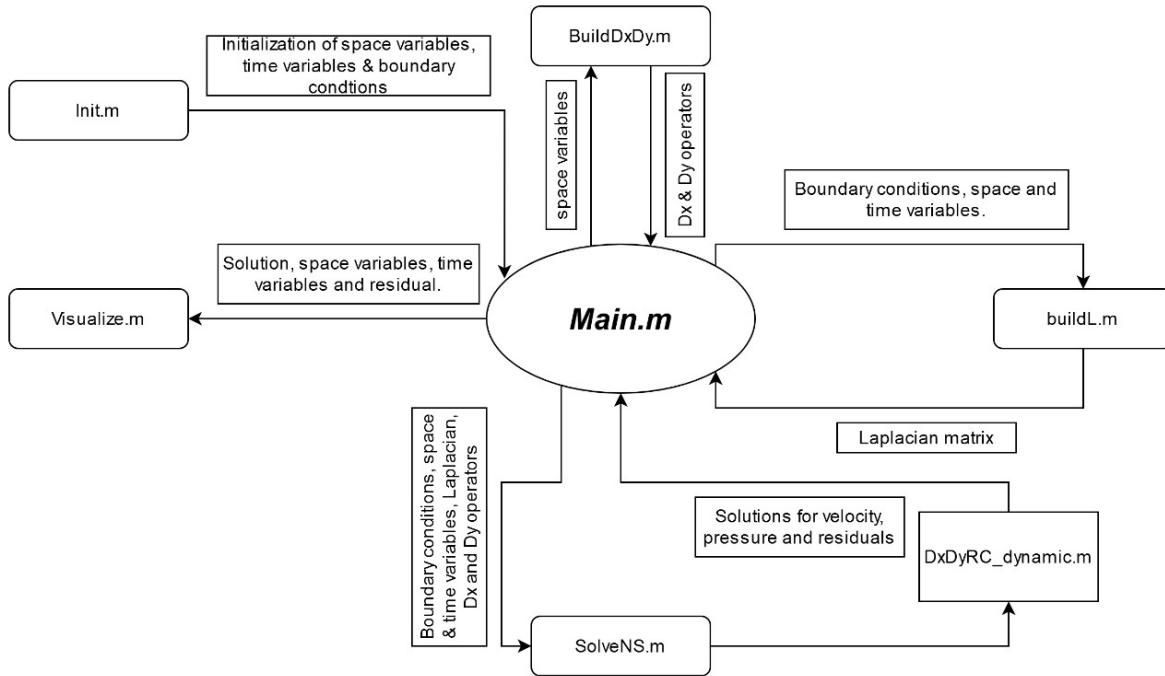
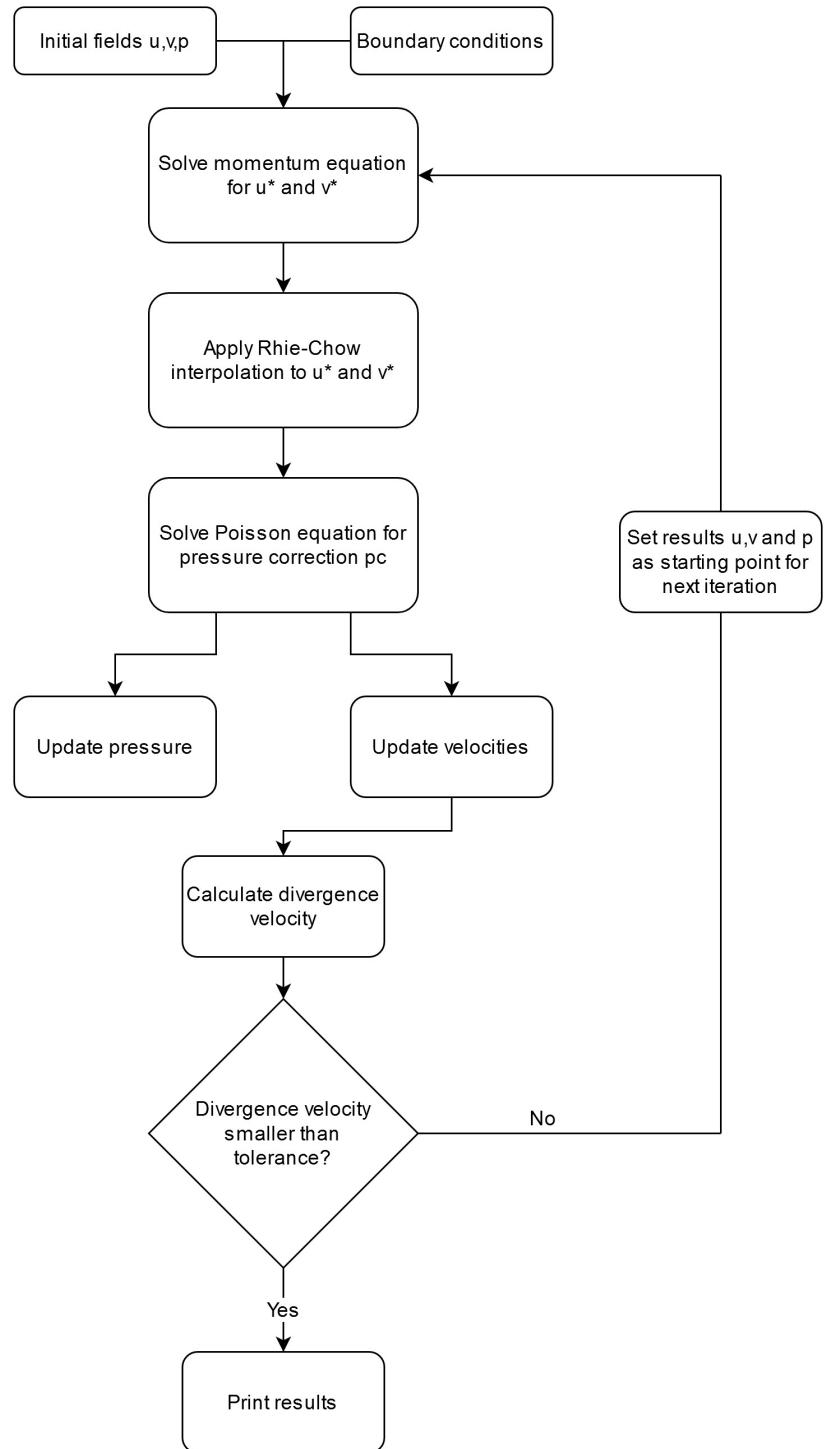


Figure 6. Program structure.

2.8 Solver Algorithm Flowchart

The flowchart in figure 7 describes the general procedure implemented in the current solver version. The corresponding equations that are solved in each step are found in the preceding chapters.

**Figure 7.** Flowchart of the implemented Navier Stokes solver.

3. Results and Discussion

In this section, the results of the Navier-Stokes solver described in chapter 2 are discussed. First, a plain channel flow without obstacles is simulated to validate the solver. Velocity and pressure contour plots are shown. Next, different obstacle shapes and their influence on the flow field are examined. Finally, the last two subsections of this chapter include important lessons learned when working with numerical simulations as well as a list of ideas for future improvements and extensions of the current solver.

3.1 Results

3.1.1 Validation Channel Flow

Low Reynolds Flow: The first setup is a rectangular channel flow with no obstacles. It serves as a validation case to debug the solver. Table 2 shows the respective problem setup. The density corresponds to dry air at room temperature. The viscosity was adapted to get the desired Reynolds number. The time step size was derived from the Courant number $C = \frac{u\Delta t}{\Delta x}$. It was found a Courant number between $C = 0.25$ and $C = 0.5$ lead to the most stable behavior of the solver. A residual of $r = 0.01$ was used as stopping criterion, together

Table 2. Setup of the channel flow validation case.

Length	1 m
Width	0.2 m
Mesh size x	250
Mesh size y	50
Inlet velocity	0.1 m/s
Density	1.117 kg/m ³
Viscosity	9.4140e-04 Pa s
Time step size	0.01 s
Reynolds number	25

with maximum number of iterations $n_{max} = 5000$. Figure 8 shows the residual of this case. The convergence behavior is good, showing no oscillations and a steady decline. Next, figure 9 shows a contour plot of the horizontal velocity u . At the walls, a boundary layer develops. Furthermore, the highest velocity occurs at the channel center near the outlet. The magnitude of $u = 0.15 m/s$ is 1.5 times the inlet velocity. This results matches the analytical solution for a laminar channel flow, in which the maximum velocity equals 1.5 times the average velocity.

The vertical velocity v is shown in figure 10. Near the inlet, two symmetric patterns can be seen that are caused by the formation of the boundary layer. Towards the end of the domain, no vertical velocity is seen. This proofs the assumption that the flow reaches the fully developed state inside the domain.

Finally, the pressure contour plot is shown in figure 11. At the two inlet corner, a pressure peak is obtained which is a boundary condition artifact. Over the rest of the channel, the

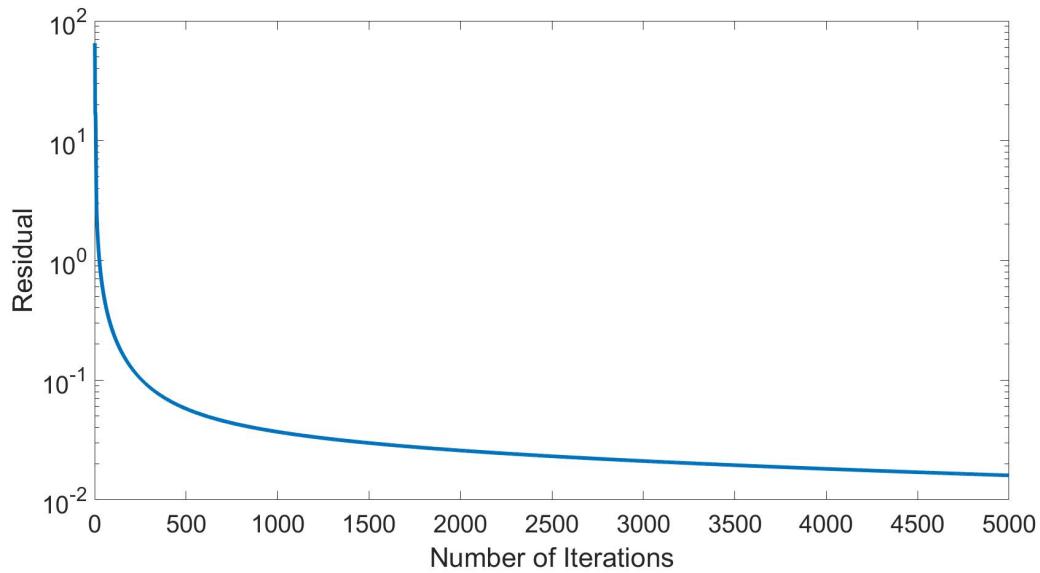


Figure 8. Convergence behavior for the channel flow at $\text{Re} = 25$.

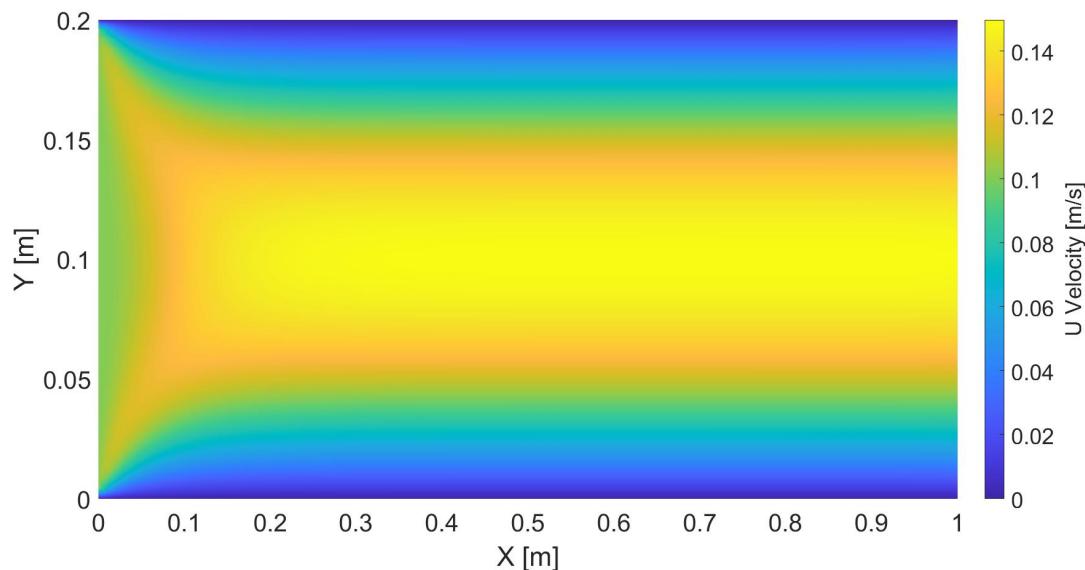


Figure 9. Horizontal velocity u contour for the channel flow at $\text{Re}=25$.

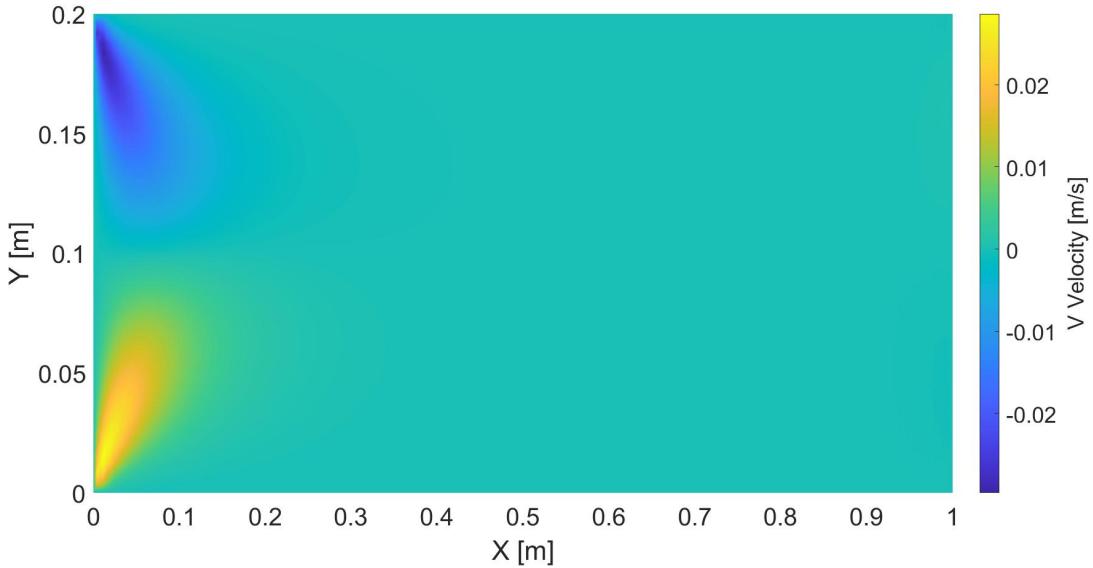


Figure 10. Vertical velocity v contour for the channel flow at $Re=25$.

pressure decreases steadily up to the specified outlet boundary of $0Pa$. Unfortunately, towards the outlet some pressure oscillations along the x direction can be seen. They are damped by the use of Rhie and Chow interpolation, but not completely limited. However, they seem to be small enough in magnitude not to alter the velocity fields significantly

High Reynolds Flow: In order to test the present solver for higher Reynolds numbers, the same setup as the previous case (see table 2) is used, but the viscosity is divided by a factor of 10 to achieve a Reynolds number of $Re = 250$. Also, the *crescendo* approach described in chapter 2.6 was activated. The residual plot (figure 12) shows the superior performance of the *crescendo* increase in Reynolds number. The initial residual is lower than for the low Reynolds case and after only 190 iterations, the residual threshold is reached. This is possible because the domain was initialized with the solution of the preceding step at $Re = 225$. The horizontal velocity plot in figure 13 shows similarities to the low Reynolds setup. As expected, the boundary layer development progresses farther into the domain. Only shortly before the outlet, the velocity profile reaches the fully developed state.

The vertical velocity contour (figure 14) also supports the claim that the flow is fully developed just as it reaches the inlet. In order to achieve even higher Reynolds numbers, the domain length would have to be increased, in order not to violate our assumption that $v = 0$ at the outlet (see chapter 2). The pressure contour, shown in figure 15, shows a similar behavior as the low Reynolds case. The overall pressure drop along the channel is significantly smaller in the high Reynolds case, because for a laminar channel flow, the friction factor λ , which describes the pressure drop, scales with $\lambda \propto \frac{64}{Re}$. Therefore, the pressure drop decreases with increasing Reynolds number. Additionally, the pressure plot

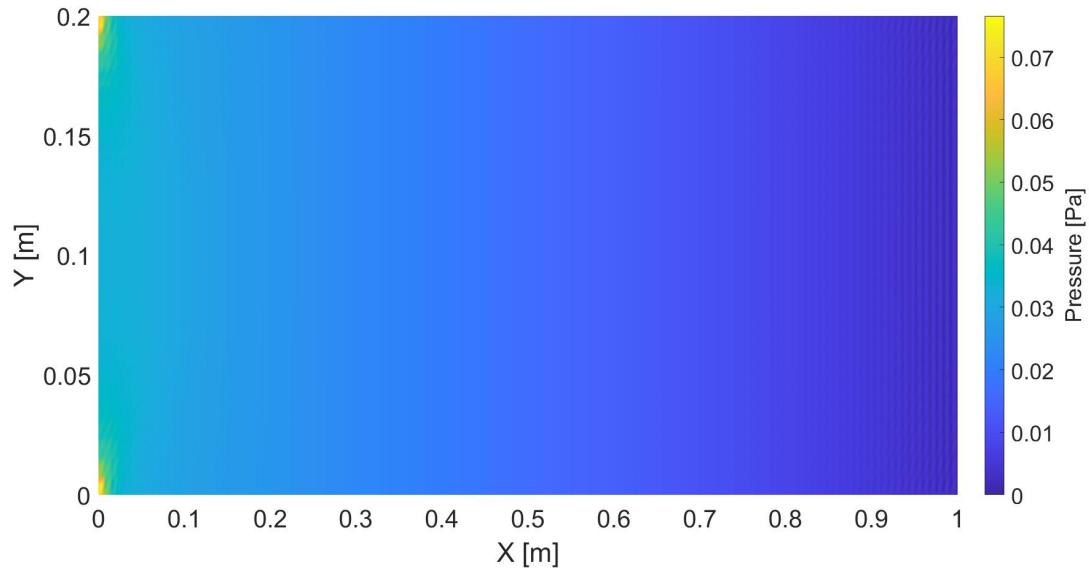


Figure 11. Pressure p contour for the channel flow at $\text{Re}=25$.

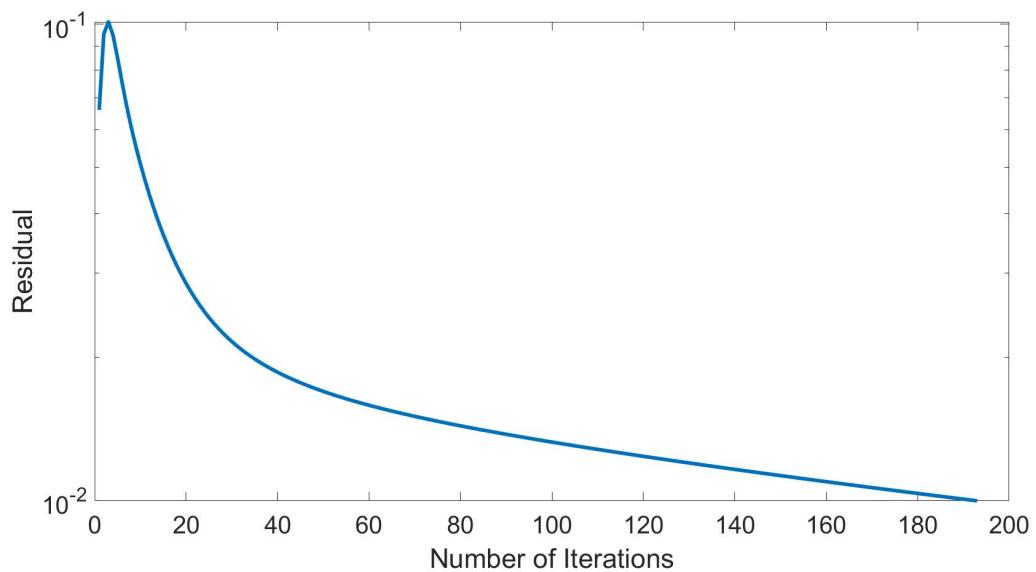


Figure 12. Convergence behavior for the channel flow at $\text{Re} = 250$.

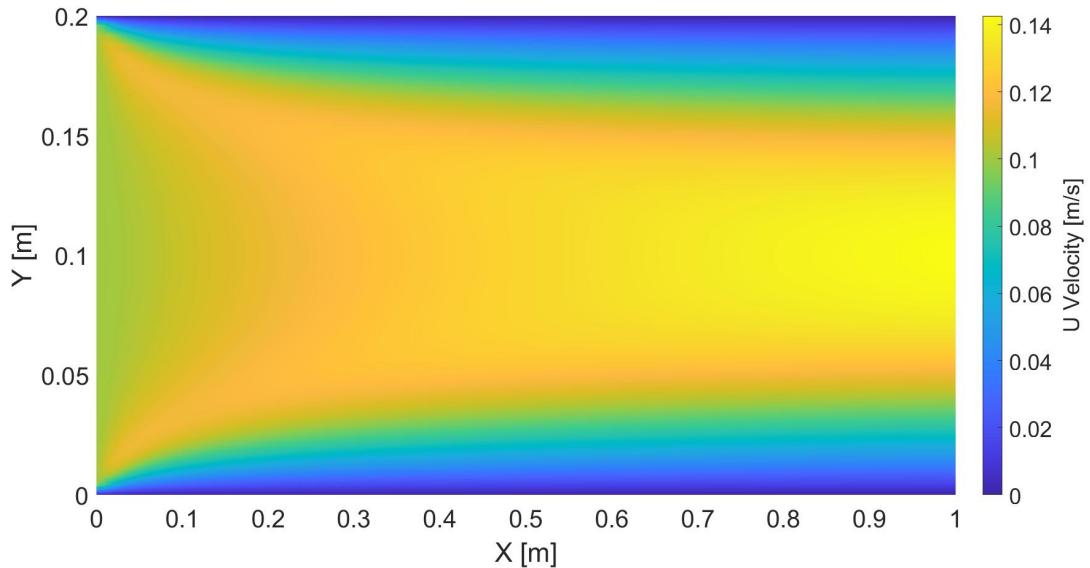


Figure 13. Horizontal velocity u contour for the channel flow at $\text{Re}=250$.

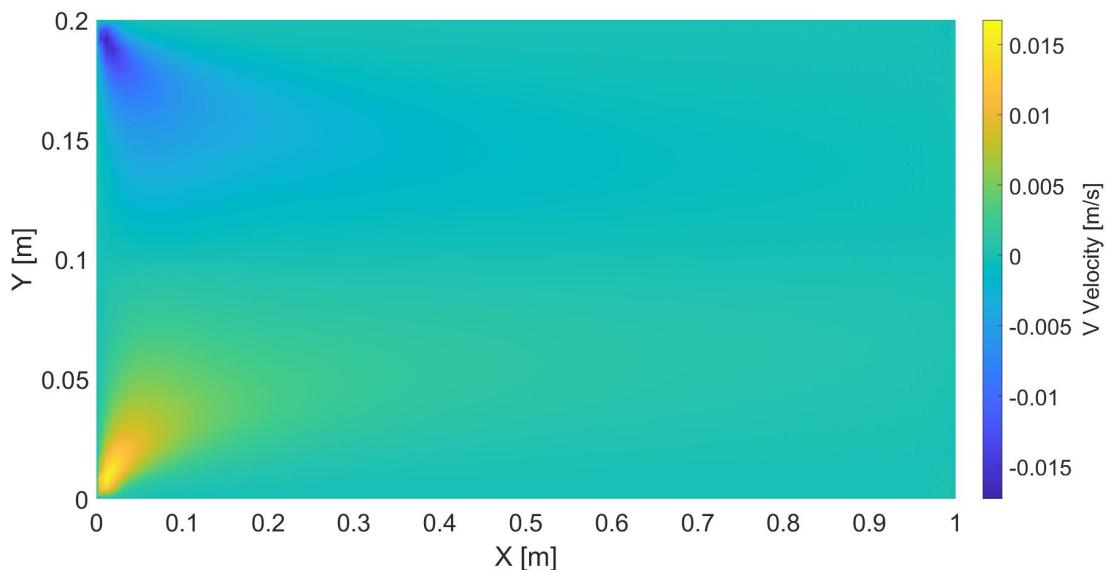


Figure 14. Vertical velocity v contour for the channel flow at $\text{Re}=250$.

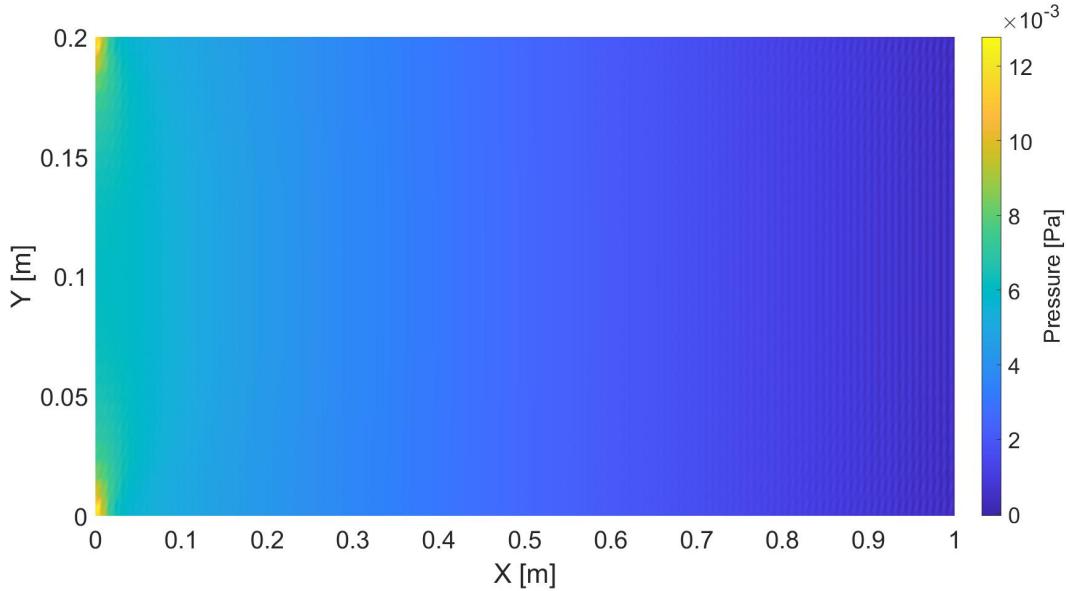


Figure 15. Pressure p contour for the channel flow at $Re=250$.

shows that the oscillating behavior towards domain end is not excited by higher Reynolds numbers, which indicates that the Rhie-Chow interpolation algorithm works correctly.

To sum up, the channel flow validation case proved that the developed Navier-Stokes solver is able to correctly describe in-compressible fluid flow in a planar channel. The pressure and velocity results are physically meaningful and match the expectations from analytical considerations and literature. Furthermore, the high Reynolds case shows that the code is able to produce good results for Reynolds numbers up to 250. When working with such magnitudes, the crescendo approach proved to be very useful in accelerating the convergence behavior.

For the scope of this project, the maximum Reynolds number will be $Re = 250$. Although, by extending the length of the domain, even larger values would be possible using the current code version. During preliminary testing sessions, results for Reynolds numbers up to 500 were achieved. The theoretical limit of laminar channel flow lies in the area of $Re = 2300$, at which the flow turns turbulent. For simulations above that Reynolds number, a turbulence model has to be implemented.

3.1.2 Sinus Obstacle

The first object of interest is implemented as a sinus wave. Therefore, the mesh is fitted to a sinus formfunction between 0 and π on the southern boundary of the channel. The parameters are the amplitude r and the midpoint p . Furthermore, a low Reynolds number and a high Reynolds number case are set up.

Low Reynolds Flow: Table 3 shows the setup of the sinus obstacle at low Reynolds number case. First, the horizontal velocity profile is examined in figure 16. At the highest point

Table 3. Setup of the sinus obstacle low-Re case.

Length	1 m
Width	0.2 m
r parameter	0.1 m
p parameter	0.2 m
Mesh size x	250
Mesh size y	80
Inlet velocity	0.1 m/s
Density	1.117 kg/m ³
Viscosity	9.4140e-04 Pa s
Time step size	0.01 s
Reynolds number	25

of the obstacle, an increase in velocity is seen as the area of the channel is contracted. Since the Reynolds number is rather low, the flow field returns to the one of a undisturbed channel flow rather quickly.

This claim is also supported by the velocity vector plot shown in figure 17. The flow reattaches fast after the obstacle and little to no recirculation zone can be observed. Another method of visualizing a flow field is the computation of streamlines. Streamlines are curves that are tangent to the velocity field in any point. In other words, they show the path that a fluid particle will follow. Since the important aspect is the flow field behind the obstacle, 30 streamlines are created with their starting points being equidistant along y and same x position right behind the obstacle. The streamline plot is shown in figure 18. A small recirculation bubble at around $x = 0.3$ to $x = 0.4$ can be seen.

High Reynolds Flow: For the high Reynolds number case, the same setup as described in table 3 is used, only the viscosity is reduced by a factor of 10 to increase the Reynolds number to $Re = 250$. The horizontal velocity plot in figure 19 indicates that by increasing the Reynolds number, a larger flow pattern of vortices is created behind the obstacle. In the current domain, the flow does not revert to the channel flow velocity distribution. The vector plot (20) reveals that there is a large vortex behind the obstacle and another one in the top right corner of the domain. In the streamline plot, shown in figure 21, the main re-circulation bubble after the obstacle becomes visible. Moreover, it seems to consist of two counter-rotating vortices. The additional, third vortex in the top right corner of the domain is indicated by the bulge in the streamline in the upper half of the channel.

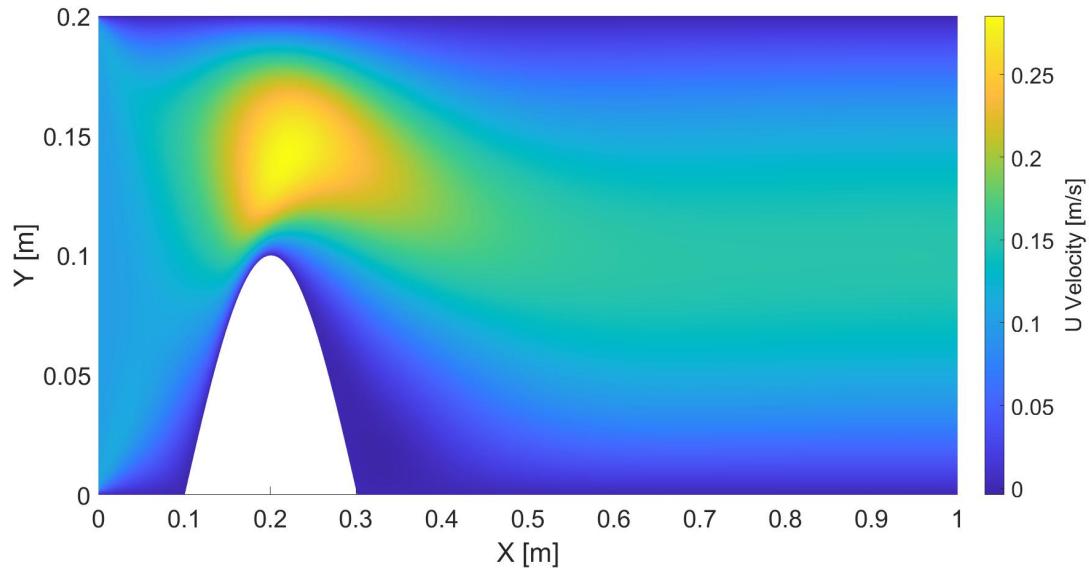


Figure 16. Horizontal velocity u contour for the sinus obstacle at $\text{Re}=25$.

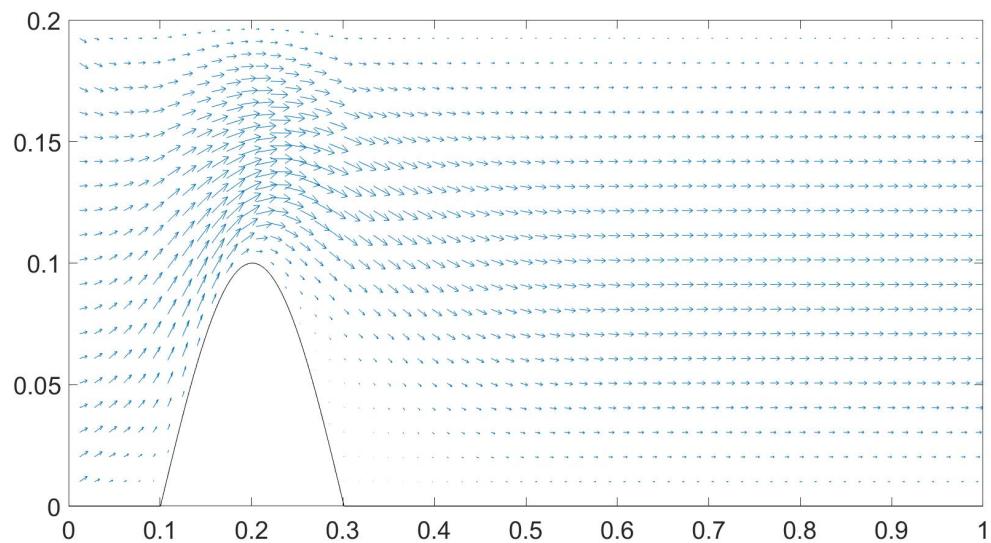


Figure 17. Vector plot for the sinus obstacle at $\text{Re}=25$.

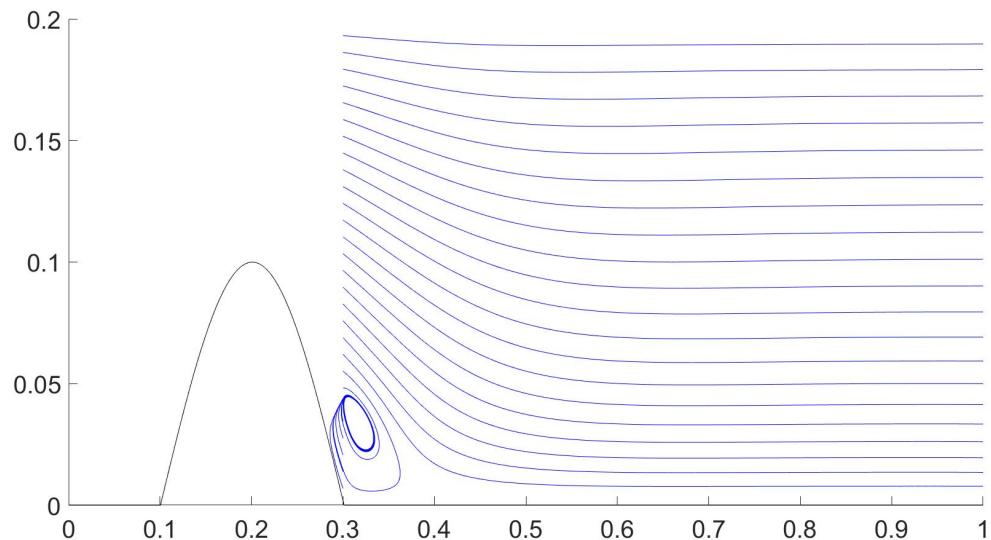


Figure 18. Streamline plot for the sinus obstacle at $\text{Re}=25$.

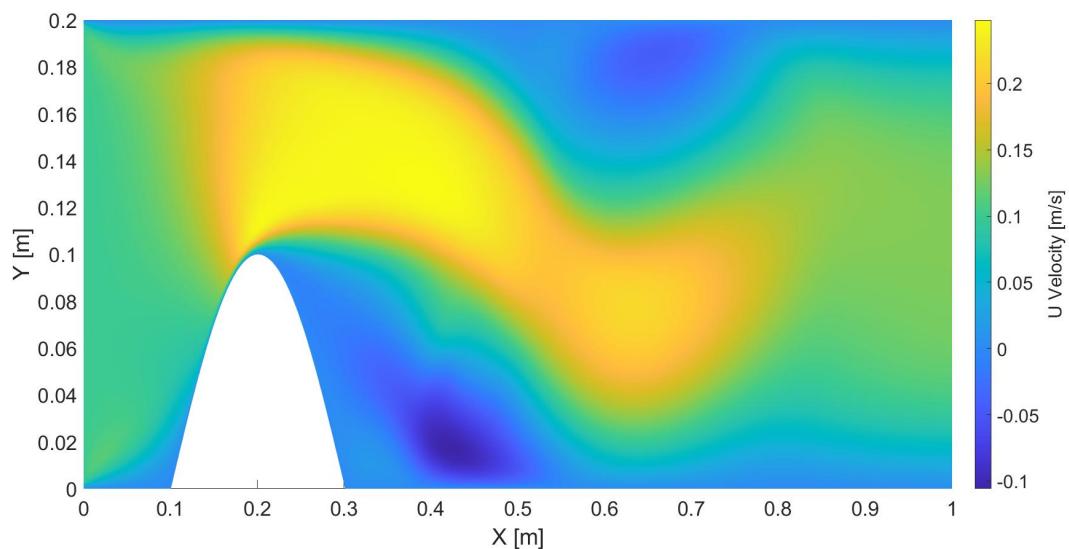


Figure 19. Horizontal velocity u contour for the sinus obstacle at $\text{Re}=250$.

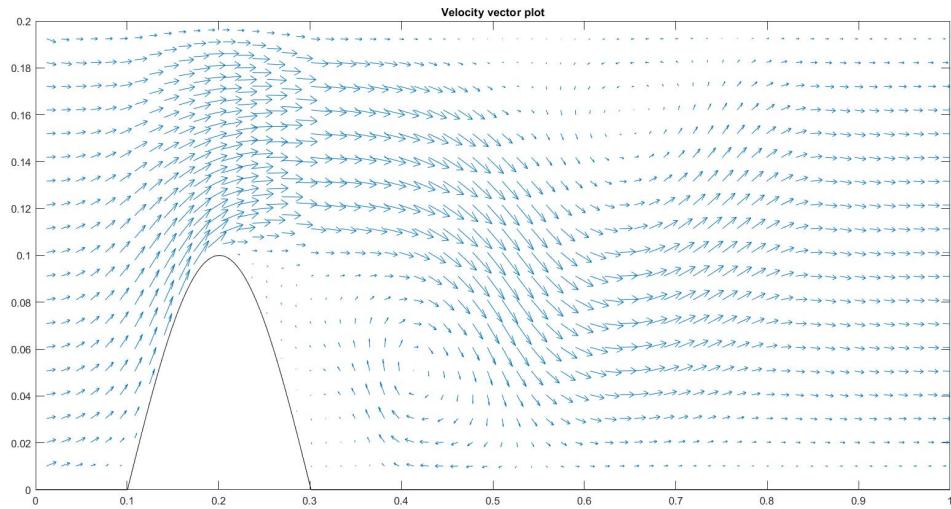


Figure 20. Vector plot for the sinus at $\text{Re}=250$.

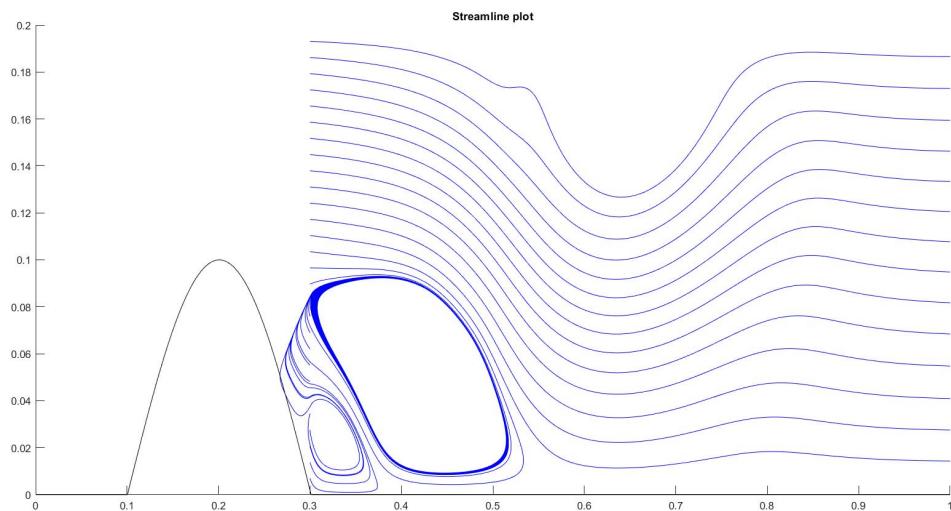


Figure 21. Streamline plot for the sinus obstacle at $\text{Re}=250$.

3.1.3 Radial Basis Function Obstacle

For the radial basis function obstacle, an exponential function is implemented that describes the southern wall shape. Its definition is:

$$y(x) = m \cdot \exp\left(\left(-\epsilon(x - p)^2\right)\right) \quad (43)$$

The height of the obstacle is controlled by parameter m , its sharpness by ϵ and the position by p . The advantage over the sinus obstacle is that the sharpness of the obstacle can be controlled and that the mesh is less disturbed at the intersection between the horizontal wall and the obstacle walls. Again, two cases are setup just like for the sinus obstacle: one with a low Reynolds number, one with a high Reynolds number.

Low Reynolds Flow: Table 4 shows the problem setup for the low Reynolds number setup of the exponential obstacle.

Table 4. Setup of the exponential obstacle low-Re case.

Length	1 m
Width	0.2 m
m parameter	0.1 m
p parameter	0.3 m
ϵ parameter	500
Mesh size x	250
Mesh size y	50
Inlet velocity	0.1 m/s
Density	1.117 kg/m ³
Viscosity	9.4140e-04 Pa s
Time step size	0.01 s
Reynolds number	25

Figure 22 shows the horizontal velocity contour for the exponential obstacle at $Re = 25$. At top of the obstacle, the channel area is the smallest and therefore the incompressible fluid is accelerated by area contraction. After the flow passed the obstacle, it decelerates again. At the right, descending side of the obstacle, a region of near-zero horizontal velocity can be seen. This is the re-circulation zone. In order to make this zone visible, a vector plot is shown in figure 23. The arrows indicate the flow direction and their length is scaled by the velocity magnitude. Therefore, the re-circulation zone is easily noticeable.

The resulting plot is shown in figure 24. For more results of this case, like pressure and convergence behavior, the reader is referred to the appendix.

High Reynolds Flow: The same obstacle geometry was also simulated for a high Reynolds number of $Re = 150$. The viscosity of the fluid is therefore decreased by a factor of 6. To achieve stable convergence, the number of nodes in y direction has to be doubled from

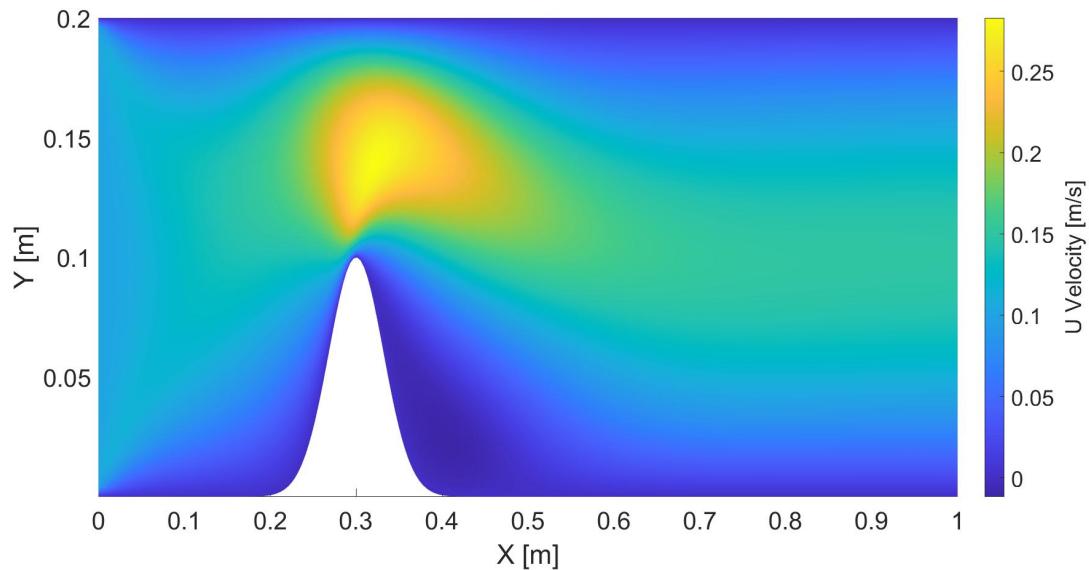


Figure 22. Horizontal velocity u contour for the exponential obstacle at $\text{Re}=25$.

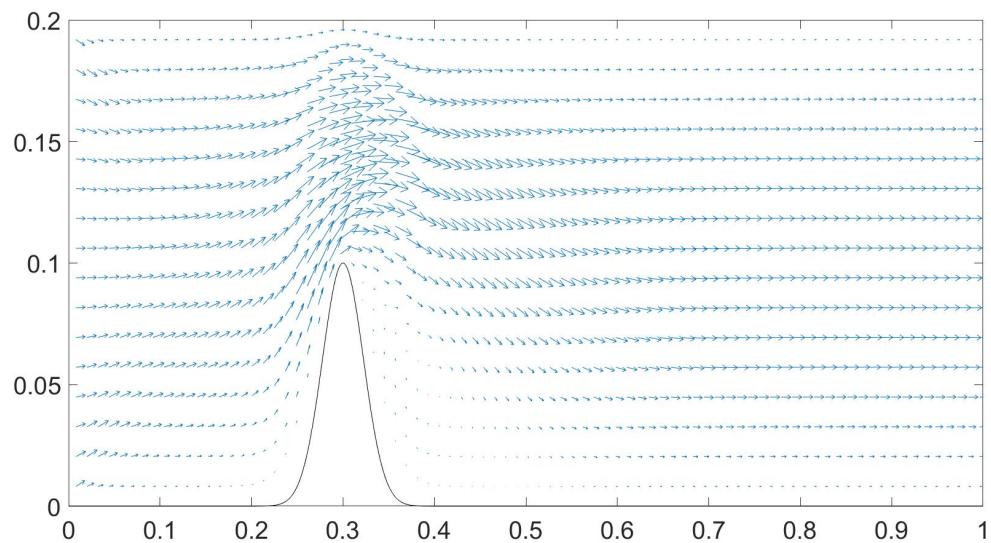


Figure 23. Vector plot for the exponential obstacle at $\text{Re}=25$.

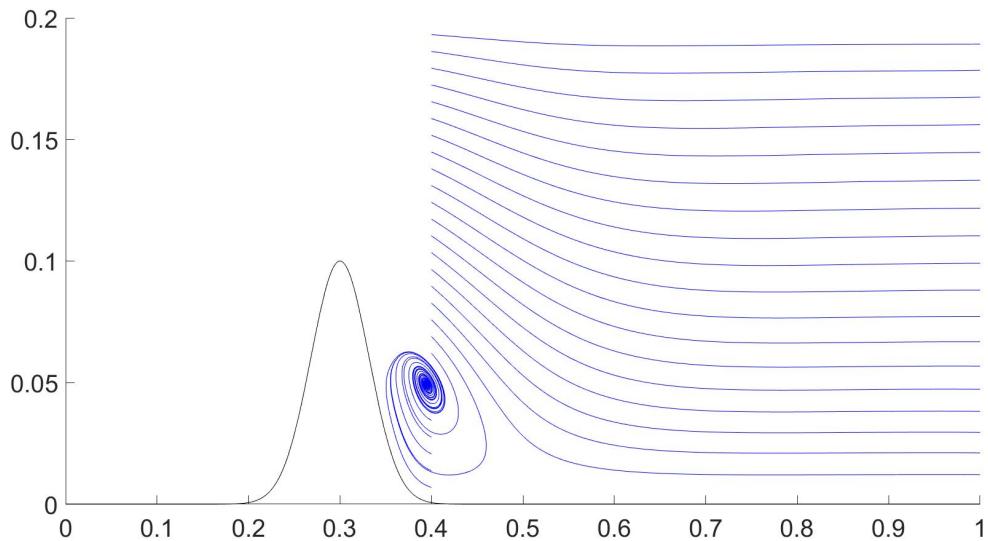


Figure 24. Streamline plot for the exponential obstacle at $\text{Re}=25$.

50 to 100. Besides that, the problem setup in table 4 holds. In figure 25, the horizontal velocity contour is shown. The maximum magnitude is similar to figure 22, but the area of increased velocity protrudes further into the domain. As expected, the re-circulation zone grows with the Reynolds number as well. This is illustrated by the vector plot in figure 26 and in the streamline plot 27. The re-circulation bubble extends until $x = 0.8$ for the high Reynolds case. For the low Reynolds case, the flow already reattached to the wall at around $x = 0.6$. Again, more plots illustrating this case can be found in the appendix. []

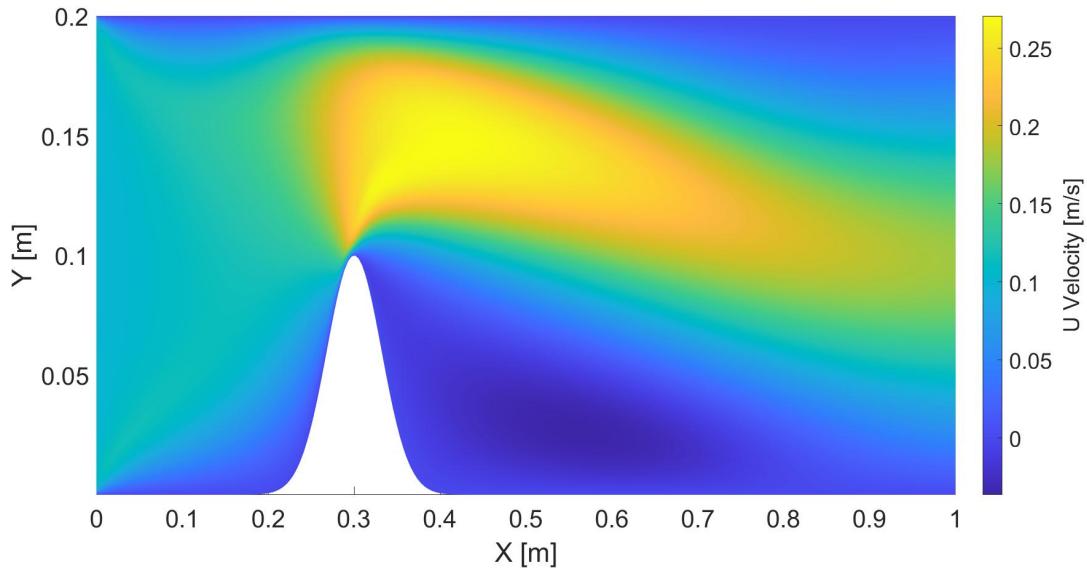


Figure 25. Horizontal velocity u contour for the exponential obstacle at $\text{Re}=150$.

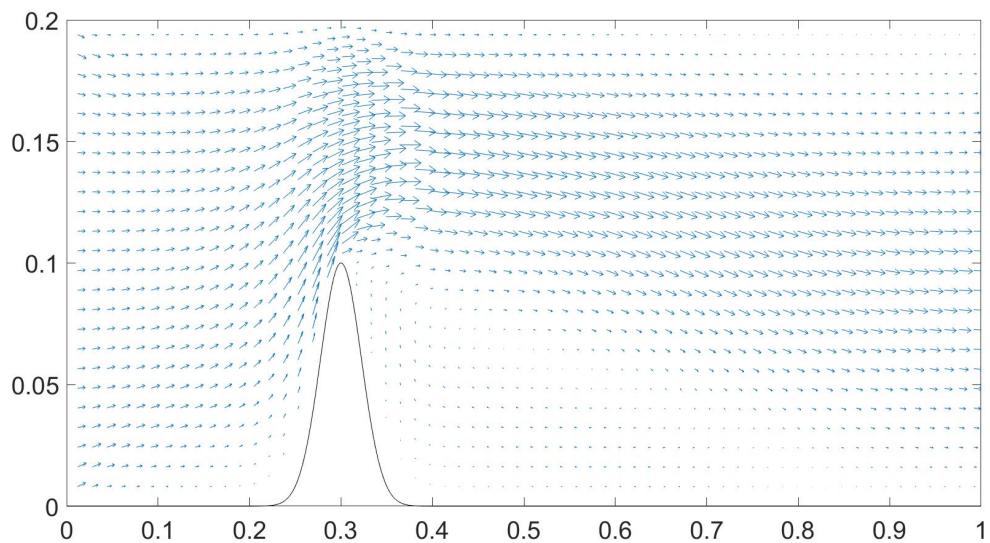


Figure 26. Vector plot for the exponential obstacle at $\text{Re}=150$.

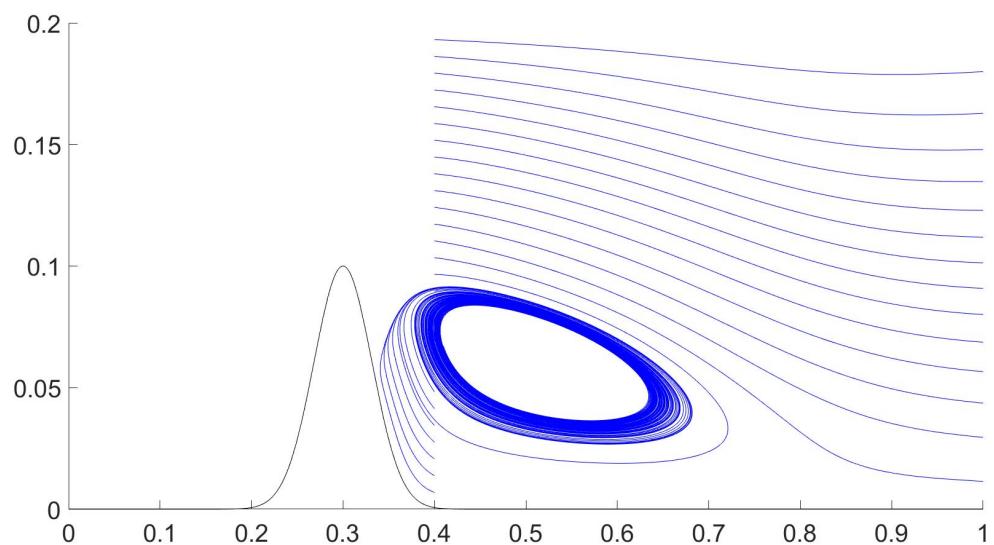


Figure 27. Streamline plot for the exponential obstacle at $\text{Re}=150$.

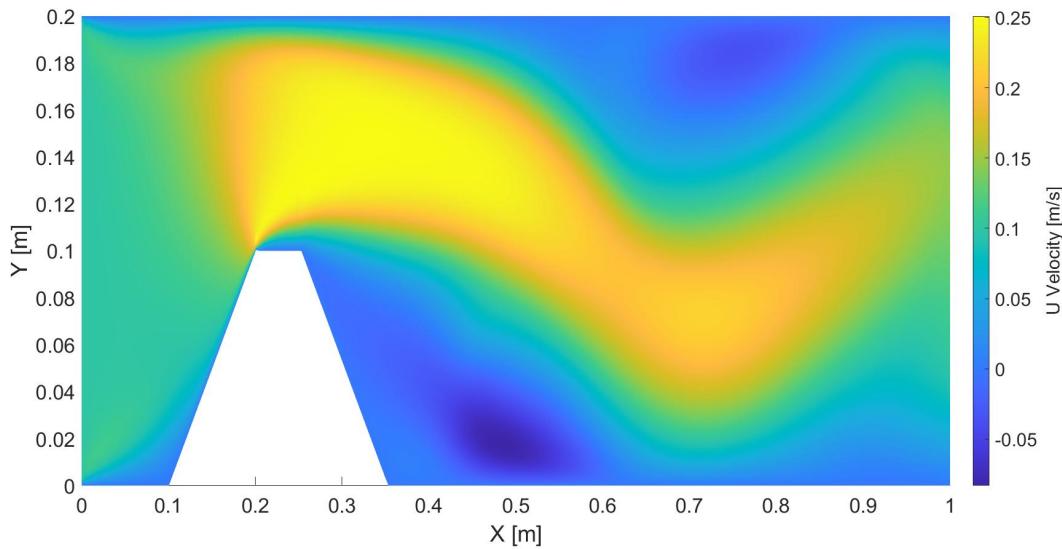


Figure 28. Horizontal velocity contour for a rib-shaped obstacle at $Re=200$.

3.2 Futher Obstacle Shapes

Other obstacle shapes than the two describes above were also tested. However, a lot proved to give convergences issues. As an example, figure 28 shows the horizontal flow velocity u over a rib-shaped obstacle at a Reynolds number of $Re = 200$. Although the formation of vortices is clearly visible, the algorithm was not able to reach the residual threshold. Therefore, no extensive investigation into this obstacle shape was conducted.

In order to recreate figure 2 from literature, a semi-circle object was implemented as well. Figure 29 shows the corresponding horizontal velocity distribution at $Re = 25$. Due to the axis not being scales equally, the circle is shaped like an ellipse in the plots. Nonetheless, it is a circle when using 1:1 axis scaling.

From the vector plot (figure 30) it can be seen that for low Reynolds numbers, the flow attaches rather quickly and no significant recirculation zone is observed. For larger Reynolds numbers, the formation of vortices is expected. However, there are no converged results available for the circular object at high Reynolds numbers as of the writing of this report.

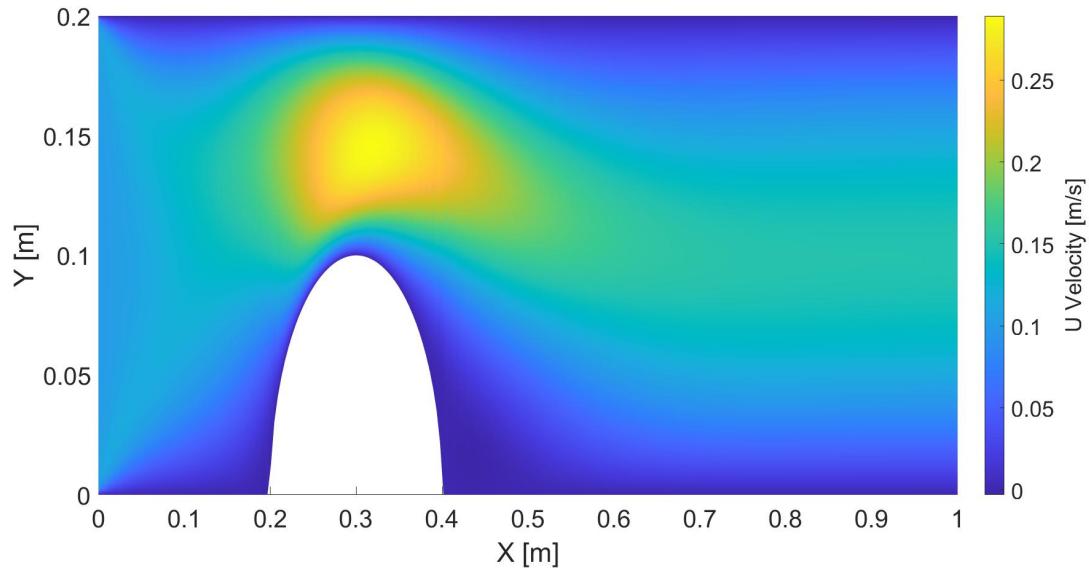


Figure 29. Horizontal velocity contour for a circular obstacle at $Re=25$.

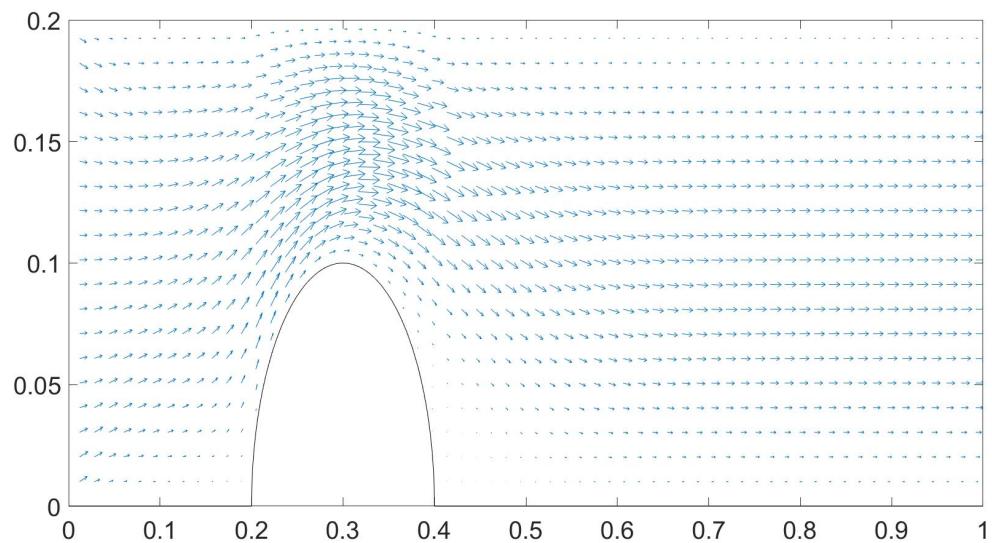


Figure 30. Velocity vector plot for a circular obstacle at $Re=25$.

3.3 Lessons Learned and Best Practices

In this section, the authors want to note down their lessons learned to help the next generations of students to improve their projects and get even better results faster:

- Instead of trying to implement all features (e.g. obstacles, Rhie-Chow interpolation, turbulence etc.) during the first try, start with something simple. A regular channel flow is easier to validate against literature and easier to debug.
- Always check your boundary conditions twice. Make sure that they are physical.
- A messy, slow code that produces reasonable results is better than a fast code producing rubbish. Cleaning and improving the code should be done if there is time left at the end.
- Don't aim too high. A fully turbulent flow inside a rocket engine with combustion flame modeling, heat transfer, supersonic expansion and fluid structure interaction sounds awesome, but is a little out of what is achievable in the given time-frame.
- Communicate with other groups. They may have similar problems or better approaches to tackle an issue. Learn and give back advice.
- Gitlab is your friend for code sharing and tracking, use it.
- Specific tips for teams working on Navier-Stokes solvers:
 - The current setup of boundary conditions (velocity specified at inlet, pressure at outlet) is easy to implement and converges fast and stable.
 - If even the channel flow case does not work out, go one step back to Stokes flow which omits the non-linear terms in the Navier-Stokes equations
 - Start with low flow velocities / Reynolds numbers and work your way up
 - A CFL number between 0.25 and 0.5 proved to work best for the discretization applied in this report
 - For higher Reynolds numbers, you need a finer mesh resolution as well
 - If the solution diverges, plot the velocity and pressure fields, but also their derivatives after every time step. Carefully investigate in which region of the domain the solution behaves badly and think about the reason.
 - For post-processing, the Matlab commands *quiver* and *streamline* are hidden gems

3.4 Future Improvements

Starting from the current state of the project, there is of course a lot of aspects that could be improved or extended. Some of the most important ideas are summarized in the following:

- Extend the achievable Reynolds range up to the border of the laminar regime
- Include a turbulence model to simulate turbulent channel flow
- Implement obstacles that are fully immersed in the fluid, for example a cylinder or a square
- Examine the time-depended behavior of the vortices behind the obstacle
- Conduct extensive parametric studies to find a correlation between Reynolds number, obstacle shape and recirculation zone properties
- From the pressure distribution along the obstacle, include a structure mechanical formulation of the obstacle domain. Final goal is a coupled fluid-structure interaction model
- Include the energy equation in the solver implementation to enable thermal calculations
- Extend the solver capabilities to three-dimensional or compressible flow

4. Conclusions

Over the course of the project, a solver for the two-dimensional, in-compressible Navier Stokes equations was derived. Therefore, temporal as well as spatial discretization schemes were applied. For the boundary and initial conditions, reasonable assumptions were made. The problem of checker-boarding was addressed by adding Rhie and Chow interpolation. By examining the flow field in a planar channel flow, the solver was found to give good results in a Reynolds range up to 250. The predicted velocity and pressure fields matched physical expectations. The correct development of a boundary layer was observed. The velocity fields obtained from the solver are smooth, while the pressure field showed oscillating behavior towards the outlet boundary. This issue requires additional investigation, but fortunately did not prove to be problematic over the course of this project.

Further increase in Reynolds number would require the computational domain to be extended since the boundary conditions require fully developed flow at the outlet. Nonetheless, there is reason to believe that the code is capable of finding a solution for higher Reynolds number as well.

After validation, several obstacle shapes and Reynolds numbers were investigated. The first object of interest was an obstacle shaped by a trigonometric function. The obstacle height was set to half the channel height. For low Reynolds numbers, a small recirculation bubble after the obstacle was observed. When increasing the Reynolds number of the flow, a more complex pattern of vortices developed. This matches the expectations of flow over blunt bodies from experiments.

Next, an exponential obstacle was implemented, which mainly differs from the sinus ones by its dramatically increased sharpness. Therefore, the observed re-circulation zone was found to be significantly bigger for $Re = 25$ when compared to the sinus case. Unfortunately, no convergence was reached for $Re = 250$, preventing a direct comparison of the two obstacles at that Reynolds number. As a surrogate, $Re = 150$ was examined for the exponential obstacle. The observed vortex was found to be bigger than the ones for the sinus obstacle, although the Reynolds number is significantly lower. In order to resolve this obstacle at $Re = 250$, an extended domain would be necessary since the flow will not reattached before reaching the current outlet. This may also explain the convergence issues.

Finally, based on the current results, suggestions were made to improve the current solver. The most important ones are the inclusion of more complex obstacles and the increase of Reynolds number up to the laminar-turbulent transition point. In the far future, turbulent or compressible flow may be interesting topics to study.

To sum up, the Navier Stokes solver developed over the course of this project successfully obtained solutions channel flow as well as flow over obstacles of different shapes under several flow conditions. The resulting flow field is physically reasonable and matches the authors expectations. The current code-base has some smaller shortcomings and limitations, but over all the the goal of the project was successfully met.

5. Appendix

Sinus Obstacle

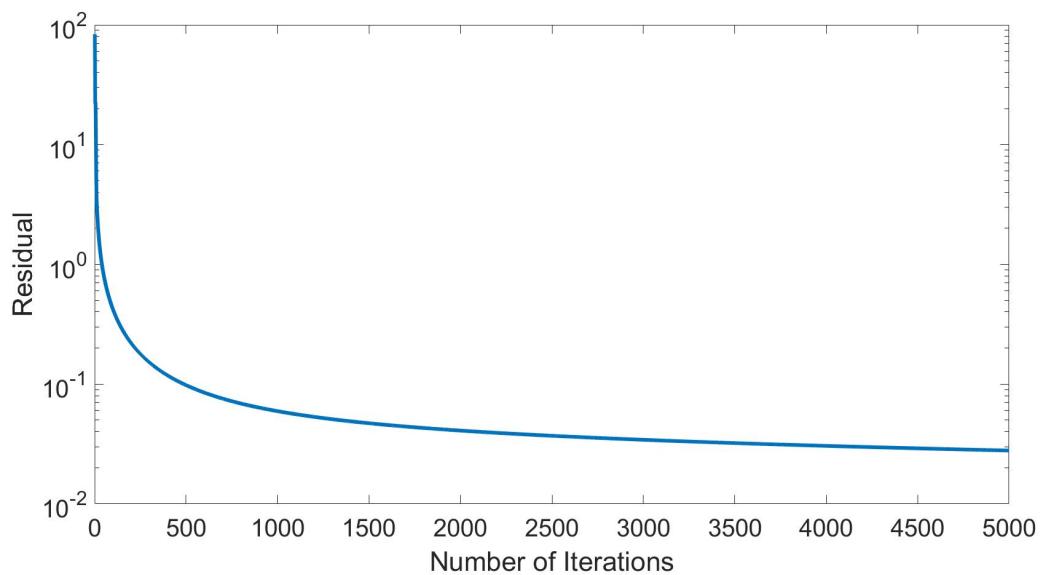


Figure 31. Convergence behavior for the sinus obstacle at $\text{Re} = 25$.

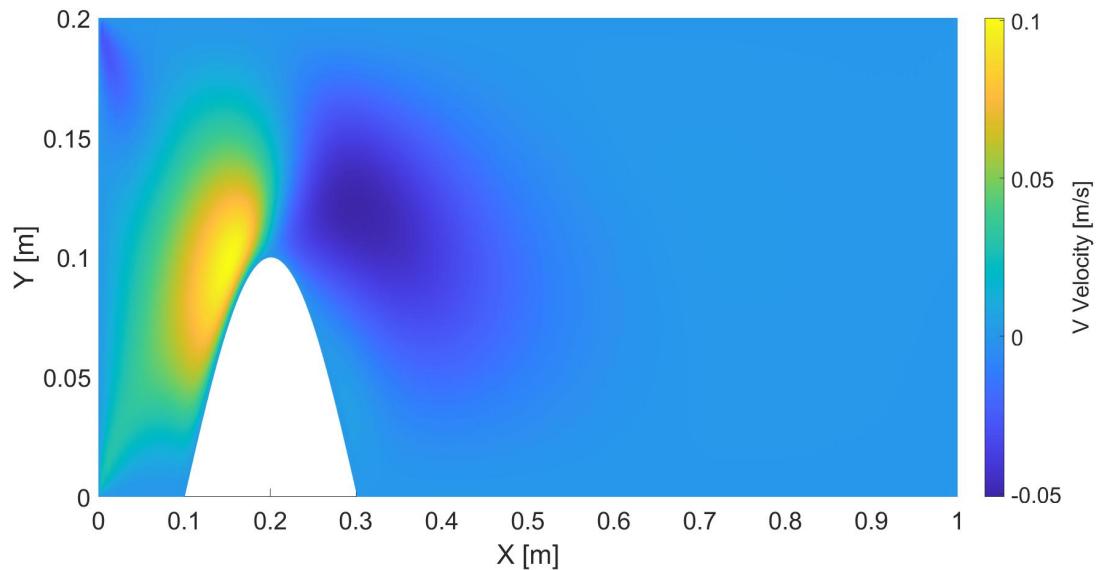


Figure 32. Vertical velocity v contour for the sinus obstacle at $\text{Re}=25$.

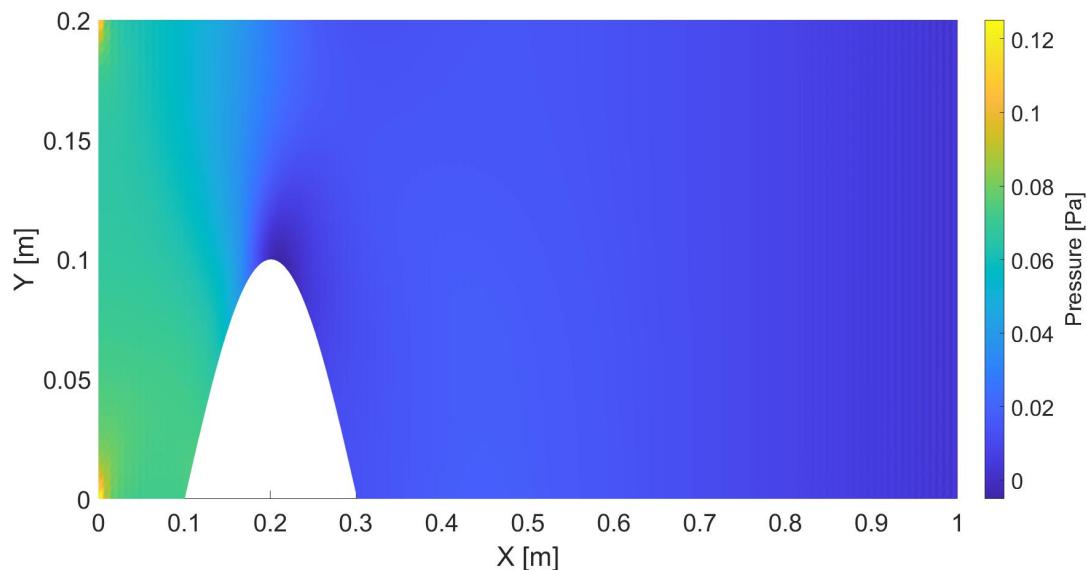


Figure 33. Pressure p contour for the sinus obstacle at $\text{Re}=25$.

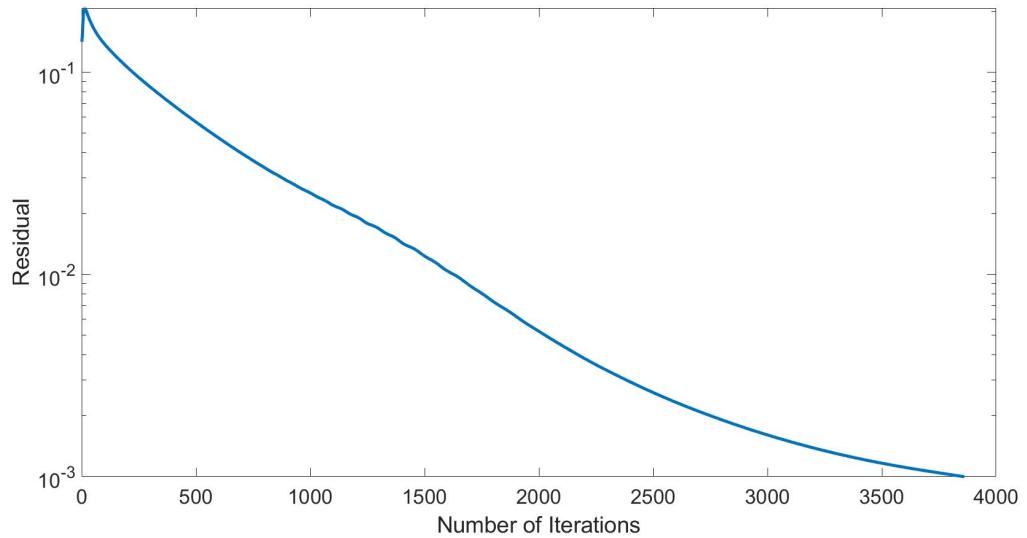


Figure 34. Convergence behavior for the sinus obstacle at $\text{Re} = 250$.

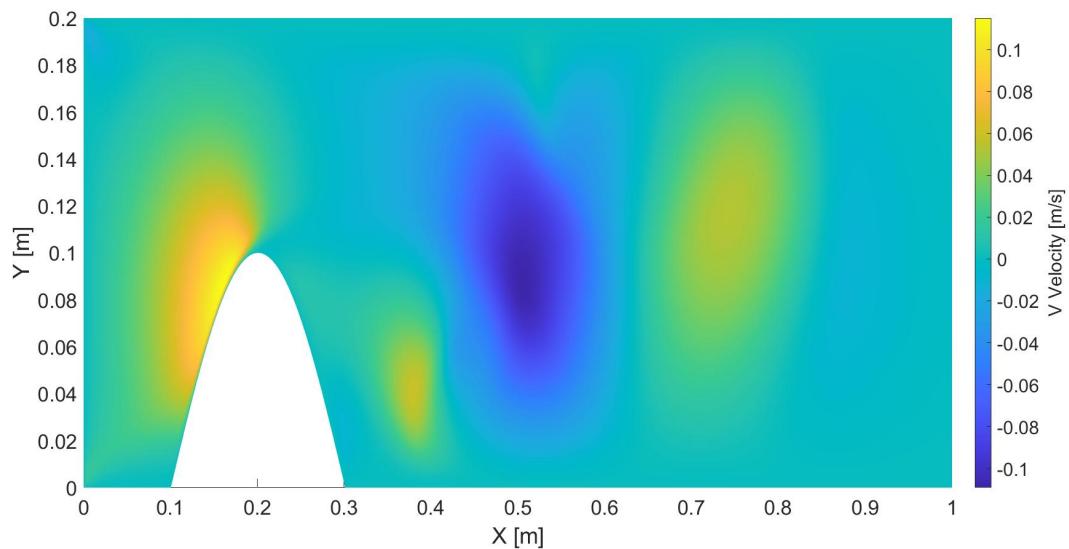


Figure 35. Vertical velocity v contour for the sinus obstacle at $\text{Re}=250$.

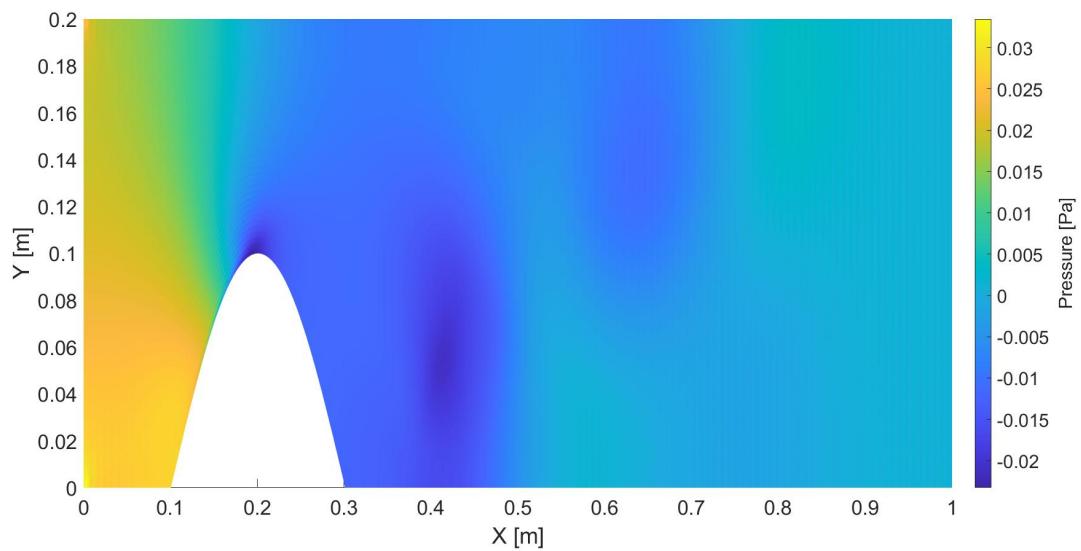


Figure 36. Pressure p contour for the sinus obstacle at $\text{Re}=250$.

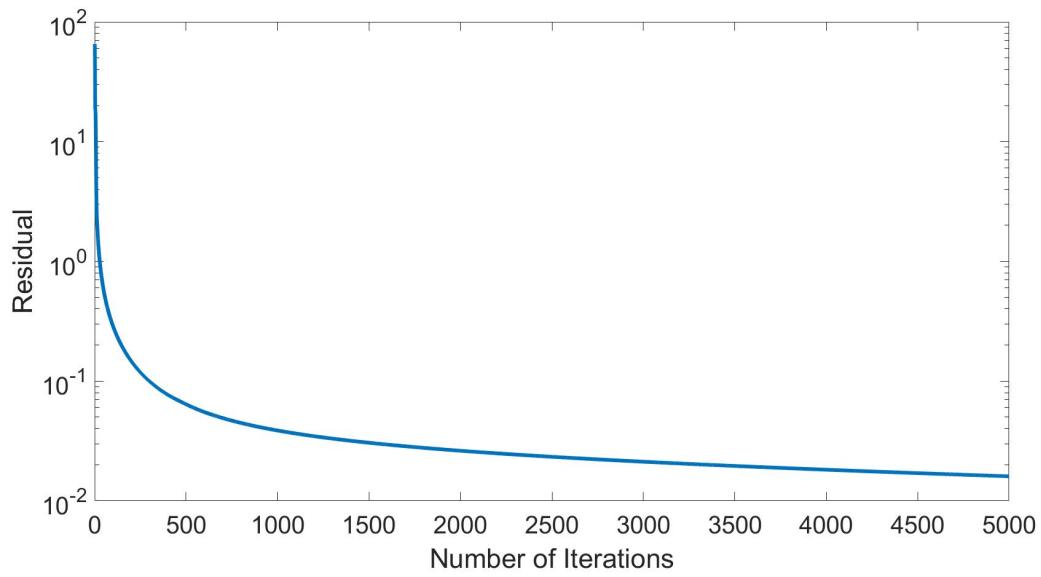
Radial Basis Function Obstacle

Figure 37. Convergence behavior for the exponential obstacle at $\text{Re} = 25$.

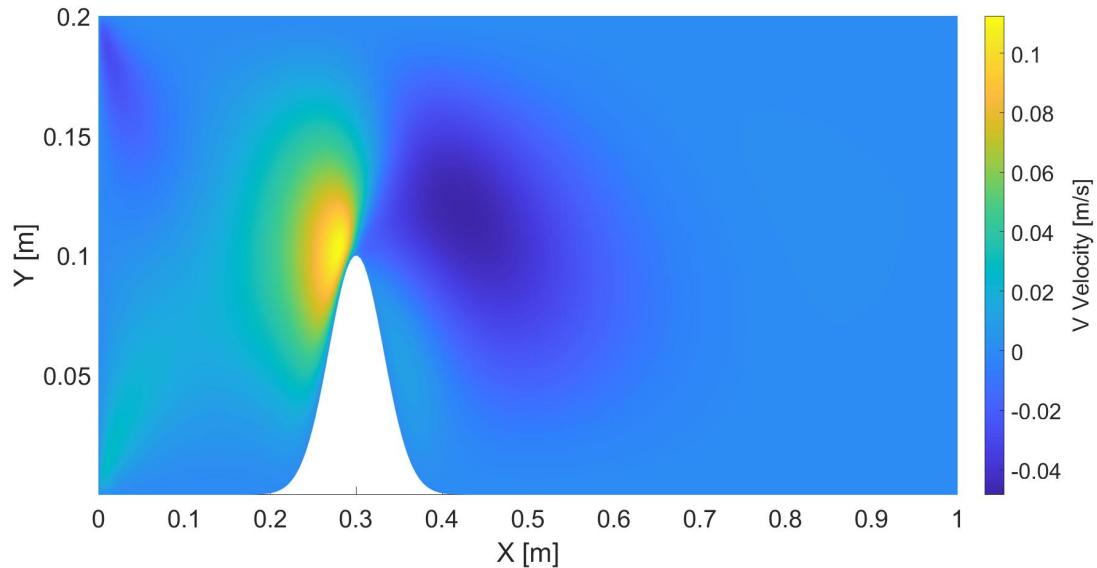


Figure 38. Vertical velocity v contour for the exponential obstacle at $\text{Re}=25$.

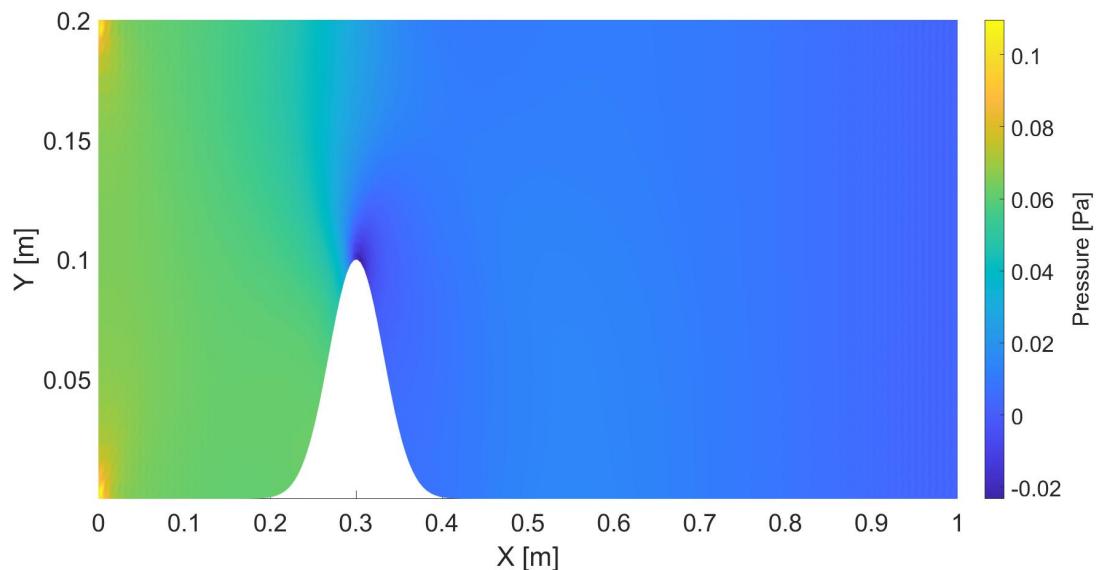


Figure 39. Pressure p contour for the exponential obstacle at $\text{Re}=25$.

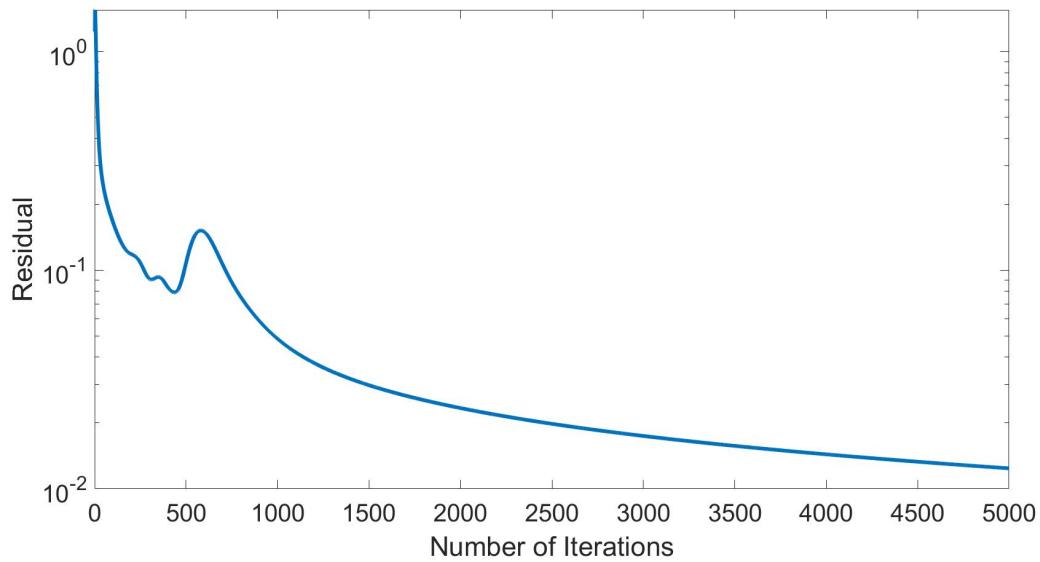


Figure 40. Convergence behavior for the exponential obstacle at $\text{Re} = 150$.

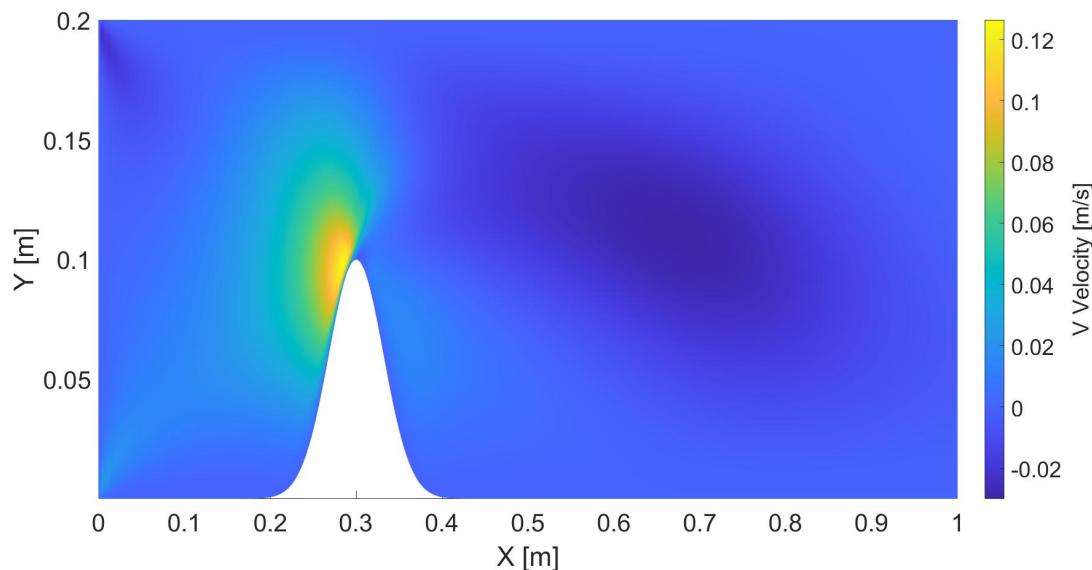


Figure 41. Vertical velocity v contour for the exponential obstacle at $\text{Re}=150$.

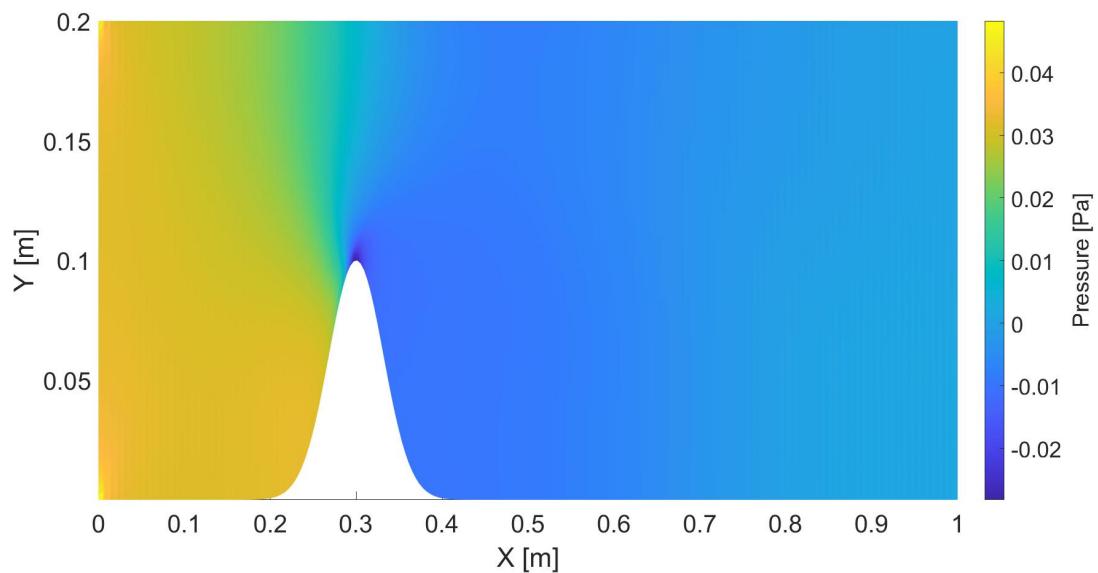


Figure 42. Pressure p contour for the exponential obstacle at $\text{Re}=150$.

Acknowledgments

Philipp: First and foremost, I want to thank Dr. Camilo Silva for putting together this fun and valuable course. We did learn a lot about numerical simulation, finite volume implementation, coding and debugging. Moreover, without his patience to answer our quadrillion messages and calls, this project would not have been possible. I also want to thank the guy giving me money (aka my dad) for allowing me to study what I like. Additionally, I have to thank Mathworks for the incredible MATLAB help page that has answers for even the most stupid questions I could come up with. Finally, I sincerely thank my six year old notebook for not dying while I was torturing it with bad-conditioned code for hours on maximum workload.

Anant: The word “thanks” is totally incapable of expressing my gratitude for the constant support offered by Prof. Dr. Silva and Mr. Akshay Kulkarni, without whom my participation in this course would not have been possible. I am also extremely thankful to Mr. Philipp Müller, who is one of the most articulate and organized people I have met, for his patience and promptness which immensely helped me cope up with my academics at TU Berlin as well as TU Munich.

References

- [1] Philip Bonnaire. “Development of a 2D stochastic, incompressible Navier-Stokes solver based on Finite Volumes and intrusive Polynomial Chaos Expansion”. Lehrstuhl für Thermodynamik. Semester Thesis. München: Technische Universität München, 2018.
- [2] Alexandre Joel Chorin. “Numerical Solution of the Navier-Stokes Equations”. In: *Mathematics of Computation* 22.104 (1968), pp. 745–762.
- [3] Katuhiko Goda. “A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows”. In: *Journal of Computational Physics* 30.1 (1979), pp. 76–95.
- [4] César Páez. “Descretisation of the Navier-Stokes equations on a co-located grid arrangement. Description of Methodology”. Lehrstuhl für Thermodynamik. Project Report. München: Technische Universität München, 2019.
- [5] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. *Hidden Fluid Mechanics: A Navier-Stokes Informed Deep Learning Framework for Assimilating Flow Visualization Data*. 2018. URL: <https://maziarraissi.github.io/HFM/> (visited on 02/14/2021).
- [6] C. M. Rhie and W. L. Chow. “Numerical study of the turbulent flow past an airfoil with trailing edge separation”. In: *AIAA Journal* 21.11 (1983), pp. 1525–1532.

- [7] Hermann Schlichting. *Boundary-layer theory*. Ninth edition. Berlin and Heidelberg: Springer, 2017.
- [8] Camilo Silva, Kilian Förner, and Wolfgang Polifke. “Notes on Computational Thermo-Fluid Dynamics”. Lehrstuhl für Thermodynamik. Lecture Notes. München: Technische Universität München, 2020.
- [9] study.com, ed. *Two dimensional Poiseuille flow between two infinite parallel plates*. 2021. URL: <https://study.com/academy/answer/consider-fully-developed-two-dimensional-poiseuille-flow-between-two-infinite-parallel-plates-separated-by-distance-h-with-both-the-top-plate-and-bottom-plate-stationary-and-a-forced-pressure-gradient.html> (visited on 02/14/2021).
- [10] S. P. Sutera and R. Skalak. “The History of Poiseuille’s Law”. In: *Annual Review of Fluid Mechanics* 25.1 (1993), pp. 1–20.