Dauphine | PSL
UNIVERSITÉ PARIS

NUMERICAL METHODS FOR DETERMINISTIC AND STOCHASTIC PROBLEMS

Numerical simulation of von Kármán vortex streets

Authors:

Víctor Ballester
Jona Nägerl

Supervisors:

Guillaume Legendre
Gabriel Turinici

M2 Applied and Theoretical Mathematics
MIDO department

March 05, 2023

Contents

1 Introduction

In the study of fluid dynamic systems, the *von Kármán vortex street* phenomenon stands a classical example of pattern formation in flows behind bodies. The pattern is characterized by alternating vortices, and it is not just an example of the complexity of fluid dynamics, but also of great importance in various research fields influencing the design of objects in fluid or aerodynamic systems. For example, the performance of a wing heavily depends on the specific flow, and therefore emergence of vortices can have an impact on the efficiency of the plane and its flight capabilities in various situations. The potential optimization of shape and coatings to achieve desired flight performances led alone in the field of aircraft engineering to a multitude of studies and research activities. Another very interesting occurrence of the phenomenon is shown in Fig. 1, where the interplay of a mountain and wind leads to pattern formation in the clouds and forms alternating vortices.

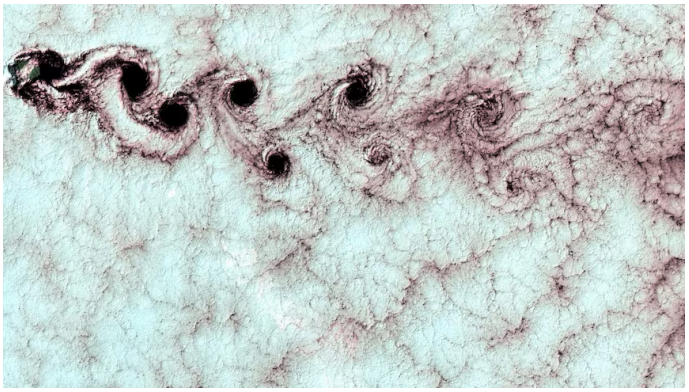


Figure 1: View of a von Kármán vortex street in the atmosphere, showcasing the pattern of swirling vortices caused by airflow around a mountain [Wik].

This project aims to numerically simulate this phenomenon in a two-dimensional flow field. By leveraging the *Chorin projection method* to decouple the velocity and pressure fields and using a *Semi-Langrangian* solver for the advective term in the underlying equations, we aim to simplify the computational complexity inherent in fluid dynamics problems and to ensure stability and accuracy in the representation of vortex shedding and evolution over time. Therefore, the plan for the following report is as follows.

First, we introduce the underlying equations as well as the boundary conditions in Section 2.1 that we deal with, and present and motivate the solver in Section 2.2. In Section 3, we numerically simulate the system and investigate the flow patterns for various setups and different shapes.

2 Methods

2.1 Posing the problem

To simplify our approach we look at a two-dimensional incompressible fluid. In this section we want to pose the problem by introducing the underlying mathematical

equations and discuss the boundary conditions.

Underlying equations

In our approach we simplify the problem to a two-dimensional incompressible viscous fluid in a rectangular domain box flowing from one side of the box to the other, encountering an obstacle in its path. The motion of the fluid is described by the *incompressible Navier-Stokes equation*

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \mu\Delta\mathbf{u} + \nabla p = 0 \qquad (1)$$
$$\nabla\cdot\mathbf{u} = 0$$

where ρ is the mass density (assumed constant), $\mathbf{u} = (u, v)$ is the fluid velocity with horizontal component u and vertical component v, μ is the dynamic viscosity of the fluid and p is the pressure. The first equation encapsulates the momentum balance within the fluid, incorporating the effects of advection ($\mathbf{u}\cdot\nabla\mathbf{u}$, velocity interaction and movement), diffusion ($\mu\Delta\mathbf{u}$, velocity spreading due to viscosity), and the pressure gradient's influence on the fluid motion (∇p), whereas the second equation, often termed the *continuity equation*, asserts the incompressibility of the fluid by ensuring the volume conservation within the flow. The equations can be adimensionalized with the following change of variables

$$\tilde{\mathbf{u}} = \frac{\mathbf{u}}{U}, \quad \tilde{p} = \frac{p}{\rho U^2}, \quad \tilde{\mathbf{x}} = \frac{\mathbf{x}}{L}, \quad \tilde{t} = \frac{U}{L}t \qquad (2)$$

where the characteristic velocity U and characteristic length L are used. These quantities represent the typical velocity of the fluid and typical lengthscale. In practice, they are taken to be the inflow velocity and the length of the obstacle in the transverse direction to the flow, respectively. Applying these change of variables to Eq. (1), we get (dropping the tilde for readability):

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \frac{1}{\mathrm{Re}}\Delta\mathbf{u} + \nabla p = 0 \qquad (3)$$
$$\nabla\cdot\mathbf{u} = 0$$

where the Reynolds number $\mathrm{Re} := \frac{\rho U L}{\mu}$ is a dimensionless parameter that measures the ratio between the inertia of the flow and the viscosity of it.

Boundary conditions

The domain Ω is the rectangular box without the obstacle. We call the horizontal direction x-direction and the vertical direction y-direction. In our study, on the left

side a laminar flow is coming into the domain. At the beginning the fluid inside the box is at rest and pressure is null within the domain. At the horizontal walls a slip condition is imposed and a free flow on the right side is assumed. Therefore we obtain as boundary conditions for the walls:

- At time $t = 0$, the fluid is at rest, and both velocity and pressure are zero.
- On the left side, the flow is incoming with a velocity equal to $U\mathbf{e}_x$, and the pressure satisfies the conditions $\frac{\partial p}{\partial x} = 0$.
- On the right side of the domain, the flow is free so that $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = 0$ and $p = 0$.
- On the horizontal sides, the walls are impenetrable, so that $v = 0$. A slip condition is imposed so that $\frac{\partial u}{\partial y} = 0$ and $\frac{\partial p}{\partial y} = 0$.

In order to accurately implement these boundary conditions, we make use of the *ghost cell* method. This technique involves the addition of a layer of cells outside the domain (see Fig. 2), called *ghost cells*. These cells are used to approximate at second order the values of the velocity and pressure at the boundary for both Dirichlet and Neumann boundary conditions.

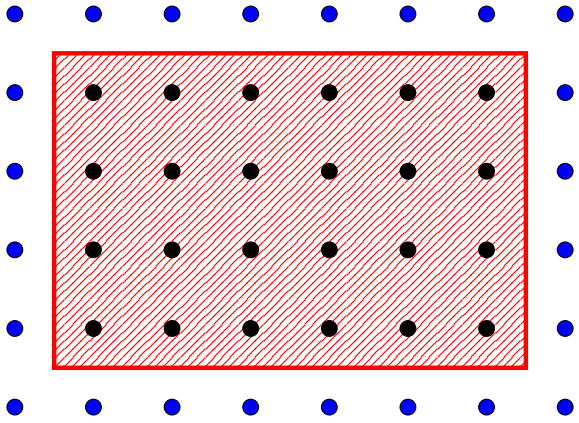


Figure 2: The ghost cell method. The domain Ω is represented in red and the dots represent the grid cells, the black ones being the inner cells and the blue ones, the ghost cells.

As an example, if $u_{0,j}$ denotes the horizontal component of $\mathbf{u}$ at the left boundary and height j (counting from 0), then it is approximated by $2U - u_{1,j}$, where $u_{1,j}$ is the horizontal component of $\mathbf{u}$ at the first cell inside the domain and the same height j. A straightforward Taylor expansion around the boundary point (say $u_{1/2,j}$) shows us that indeed this approximation is of second order. Similarly, we obtain values for $v_{0,j}$ and $p_{0,j}$ as

$$v_{0,j} = -v_{1,j}, \quad p_{0,j} = p_{1,j}$$

The other boundaries are treated in the same way, but exchanging the role between i and j and taking into account the different boundary conditions. It should be noted that when computing discrete derivatives, which involve the use of neighbouring

cells, the computations are carried out only on the inner cells, which in turn use ghost cells. But the derivatives themselves at ghost cells are never computed.

We discuss now the treatment of the boundary conditions on the object. In general dealing with the object is not an easy task. For simplicity, we model the presence of a solid body within a fluid by enforcing the fluid velocity to be zero inside the object, which correspond to imposing a no-slip condition at the boundary. We can think of it as a direct forcing method, which is a common approach in the literature, and it is based on the idea of adding a forcing term to the Navier-Stokes equations to account for the presence of the object (see [Fad+00] for more details). It is very worth-mentioning that the way we integrate the body has an influence on the flow. Especially to study the impacts of coatings or surface structure, our method, is not capable of including these effects, but in our approach we only want to observe the effect, and therefore prioritize simplicity for understanding over complexity for small details in the effect. If the reader is interested in delving deeper into the fluid-structure interactions, including the effects of surface modifications, we recommend consulting works in the field, such as those by Peskin [Pes77] or Schlichting and Gersten [SG00].

2.2 Difference Scheme

In this section we discuss our numerical solver to integrate Eq. (3) together with the boundary conditions presented in above section. To simplify the equations, we use *Chorin's projection method*. It was originally introduced by Alexandre Chorin in 1967 [Cho67] as an efficient means of solving the incompressible Navier-Stokes equations.

The idea of the method is to first predict the velocities for the next time step by solving the advection and the diffusion term. With the intermediate velocity field we solve the Poisson equation with a finite difference method (described below). Finally, the *projection step* is preformed, and the velocity field is updated with the new pressure field.

This method is based on the Helmholtz-Hodge decomposition of a vector field $\mathbf{F}$, which states that any vector field can be decomposed into a sum of a solenoidal (divergence-free) and an irrotational (curl-free) part. In our of fluid dynamics, this means that the velocity field $\mathbf{u}^*$ can be decomposed into a sum of a solenoidal part $\mathbf{u}^{n+1}$ and an irrotational part ∇p^{n+1}, where p^{n+1} is the pressure field at time t^{n+1}. The projection step is then used to enforce the incompressibility of the velocity field, and to update the velocity field with the new pressure field.

To ensure that the boundary conditions are met, we impose the conditions for the velocity right before solving the Poisson equation and at the end of each iteration. The boundary conditions for the pressure are imposed after solving the Poisson equation and before correcting the velocities in the projection step.

Next, we provide a summary of the steps in our method. Assuming we know the velocity field at time t^n, $\mathbf{u}^n$, we want to compute the velocity field at time t^{n+1}, $\mathbf{u}^{n+1}$.

1. Solve for $\mathbf{u}^a$ in $\dfrac{\mathbf{u}^a - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla \mathbf{u}^n = 0$ using a semi-Lagrangian method.
2. Solve for $\mathbf{u}^*$ in $\dfrac{\mathbf{u}^* - \mathbf{u}^a}{\Delta t} = \dfrac{1}{\mathrm{Re}} \Delta \mathbf{u}^n$.
3. Set the boundary conditions for the intermediate velocity field $\mathbf{u}^*$ and impose the velocity field $\mathbf{u}^*$ inside the object to be zero (forcing term).
4. Solve the Poisson equation for the pressure, $\Delta p^{n+1} = \dfrac{1}{\Delta t} \nabla \cdot \mathbf{u}^*$.
5. Set the boundary conditions for the pressure p^{n+1}.
6. Project the pressure to the intermediate velocity to obtain the new velocity at time t^{n+1}, $\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla p^{n+1}$.
7. Set the boundary conditions for the velocity field $\mathbf{u}^{n+1}$.

From these equations, one can easily check that we have:

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla \mathbf{u}^n - \frac{1}{\mathrm{Re}} \Delta \mathbf{u}^n + \nabla p^{n+1} &= 0 \\ \nabla \cdot \mathbf{u}^{n+1} &= 0 \end{aligned}$$

In the following sections we deepen in how do we compute each of the above steps in our method.

Semi-Lagrangian method

The Semi-Lagrangian (SL) method represents a powerful approach to solving fluid dynamics problems, particularly in the context of advection. Unlike Eulerian methods, which compute changes at fixed points in space, the SL method tracks the motion of fluid parcels. SL methods slightly differ from Lagrangian methods, as the word suggests. The second ones are rarely used in numerical methods because the particle trajectories become chaotic and wildly mixed in a short period of time. However, SL algorithms avoid this problem by *reinitializing* the Lagrangian coordinate system after each time step [Boy01]. There are several ways to implement a SL method, and we describe here the one we used in our method. Our goal in this subsection is on solving:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = 0$$

The idea is to somehow discretize the material derivative. To do that, if we call $\mathbf{u}^n$ the current velocity field and $\mathbf{u}^a$ the velocity field after the advection step, we can write a difference formula as:

$$\frac{\mathbf{u}^a - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n \cdot \nabla \mathbf{u}^n = 0$$

The following discretization for the material derivative is used:

$$\frac{\mathbf{u}(\mathbf{x}_{i,j}, t^{n+1}) - \mathbf{u}(\mathbf{x}_{i,j} - \mathbf{u}(\mathbf{x}_{i,j}, t^n)\Delta t, t^n)}{\Delta t} = 0$$

Here we are approximating the solution along a characteristic line, which generally differs from the actual characteristic associated to the solution at that point, with a constant velocity of $\mathbf{u}(\mathbf{x}_{i,j}, t^n)$ at each grid point $\mathbf{x}_{i,j}$. The reader may rapidly notice that the point $\mathbf{x}_{i,j} - \mathbf{u}(\mathbf{x}_{i,j}, t^n)\Delta t$ is not in general in the grid, and thus, we don't have the respective value of the velocity field. To solve this problem, we make use of bilinear interpolation between the four (in 2D) surrounding grid points to the point $\mathbf{x}_{i,j} - \mathbf{u}(\mathbf{x}_{i,j}, t^n)\Delta t$. This gives us an approximation of the velocity field at the point $\mathbf{x}_{i,j} - \mathbf{u}(\mathbf{x}_{i,j}, t^n)\Delta t$, which in turn provides us with the advected velocity field $\mathbf{u}^a$.

Furthermore, to improve the accuracy of the method we use the back-and-forth method [DL07]. This method first advects the velocity field using the method described above, and then advects back with opposite velocity $\mathbf{u} \to -\mathbf{u}$. With this, we can have an estimate of the error we are comitting when comparing the last result with the initial position. Using this information we can correct the initial velocity field and obtain a more accurate result. The steps are reproduced below. In order to make things more clear, we consider the general case:

$$\frac{\mathrm{D}\boldsymbol{\psi}}{\mathrm{D}t} = \frac{\partial \boldsymbol{\psi}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\psi} = 0$$

1. Advect the field $\boldsymbol{\psi}^n$ with velocity $\mathbf{u}$ to obtain $\tilde{\boldsymbol{\psi}}^a$.
2. Advect the field $\tilde{\boldsymbol{\psi}}^a$ with velocity $-\mathbf{u}$ to obtain $\bar{\boldsymbol{\psi}}^a$.
3. Advect the field $\boldsymbol{\psi}^n + \frac{1}{2}(\boldsymbol{\psi}^n - \bar{\boldsymbol{\psi}}^a)$ with velocity $\mathbf{u}$ to obtain $\boldsymbol{\psi}^a$.

In our case, the field $\boldsymbol{\psi}$ exactly corresponds to the same velocity field $\mathbf{u}$. This approach has been proved to be stable and improving considerably the accuracy of the method [DL07].

Last but not least, as in all hyperbolic equations, the Courant-Friedrichs-Lewy (CFL) condition must be satisfied. This condition states that the time step Δt must be chosen such that the real solution propagates at most at the same speed as the numerical solution. Thus, since we are using an explicit scheme we must impose a CFL of

$$|u|_{\max}\frac{\Delta t}{\Delta x} + |v|_{\max}\frac{\Delta t}{\Delta y} \leq 1$$

Diffusion term

As a second step we add diffusion using a second order central differences. The diffusion term is given by:

$$\frac{\mathbf{u}^* - \mathbf{u}^a}{\Delta t} = \frac{1}{\mathrm{Re}}\Delta \mathbf{u}^n$$

Once discretized we get, for each i, j in the grid:

$$\mathbf{u}^*_{i,j} = \mathbf{u}^a_{i,j} + \frac{\Delta t}{\mathrm{Re}} \left(\frac{\mathbf{u}^n_{i+1,j} - 2\mathbf{u}^n_{i,j} + \mathbf{u}^n_{i-1,j}}{(\Delta x)^2} + \frac{\mathbf{u}^n_{i,j+1} - 2\mathbf{u}^n_{i,j} + \mathbf{u}^n_{i,j-1}}{(\Delta y)^2} \right)$$

We write concisely the discretization in vector form, but in practice we do it for each component of the velocity field separately.

Solving for the Poisson equation

There are several efficient ways to compute the solution of the general Poisson equation $\Delta p = f$. In our case, we have to equip this equation with the boundary conditions $\partial_{\mathbf{n}} p = 0$ on the left, top and bottom boundaries, and $p = 0$ on the right boundary. We use a finite difference method to solve this equation with a 5-point stencil to approximate the Laplacian operator. The 5-point stencil is given by:

$$\Delta p_{i,j} = \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{(\Delta x)^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{(\Delta y)^2}$$

If we set $\mathbf{p} := (p_{1,1}, p_{1,2}, \ldots, p_{1,n_y}, p_{2,1}, p_{2,2}, \ldots, p_{2,n_y}, \ldots, p_{n_x,1}, p_{n_x,2}, \ldots p_{n_x,n_y})$, we can write our problem in matrix form as $\mathbf{Ap} = \mathbf{f}$, where $\mathbf{f}$ contains the discrete values at the grid points of the right-hand side of the Poisson equation in the same order as $\mathbf{p}$, and $\mathbf{A}$ is a matrix that contains the coefficients of the discrete Laplacian. Let's describe how we can build the matrix $\mathbf{A}$. Let $\mathbf{A} = \mathbf{X} + \mathbf{Y}$, where $\mathbf{X}$ and $\mathbf{Y}$ denote the matrices that contain the coefficients of the Laplacian operator in the x and y directions, respectively. We can write $\mathbf{X}$ and $\mathbf{Y}$ as:

$$\mathbf{X} = \begin{pmatrix} -\mathbf{I}_{n_y} & \mathbf{I}_{n_y} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{I}_{n_y} & -2\mathbf{I}_{n_y} & \mathbf{I}_{n_y} & \ddots & & \vdots \\ \mathbf{0} & \mathbf{I}_{n_y} & -2\mathbf{I}_{n_y} & \mathbf{I}_{n_y} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & & \ddots & \mathbf{I}_{n_y} & -2\mathbf{I}_{n_y} & \mathbf{I}_{n_y} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{I}_{n_y} & -3\mathbf{I}_{n_y} \end{pmatrix} \quad \mathbf{Y} = \begin{pmatrix} \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B} \end{pmatrix}$$

with

$$\mathbf{B} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -1 \end{pmatrix} \in \mathcal{M}_{n_y}(\mathbb{R})$$

Both $\mathbf{X}$ and $\mathbf{Y}$ are block matrices, composed of n_x matrices in each row and column of size $n_y \times n_y$ each, which results in a matrix of size $n_x n_y \times n_x n_y$. The boundary conditions (using ghost cells) on the pressure are applied at both the first and last block rows of $\mathbf{X}$ and the first and last rows of $\mathbf{B}$.

In order to solve the resulting system, we use the Cholesky decomposition, which requires the matrix to be symmetric and positive definite. The matrix $\mathbf{A}$ is symmetric, but it is not positive definite. In fact, it can be seen that it is negative definite. Thus, in our case, we solve the equivalent system $-\mathbf{Ap} = -\mathbf{f}$.

Updating velocities

As a final step, we compute the gradient of the pressure field and update the velocity field. The gradient of the pressure field is computed using a 2nd order central differences scheme. The update of the velocity field is given by:

$$\mathbf{u}_{i,j}^{n+1} = \mathbf{u}_{i,j}^{*} - \Delta t \left(\frac{p_{i+1,j}^{n} - p_{i,j}^{n}}{\Delta x}, \frac{p_{i,j+1}^{n} - p_{i,j}^{n}}{\Delta y} \right)$$

3 Results

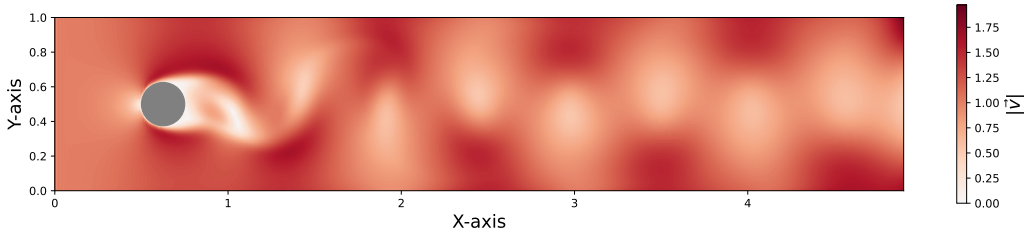
The method was implemented in C++ using the `Eigen` library for linear algebra operations [Eig], particularly for the creation of the sparse matrix $\mathbf{A}$ described above and the resolution of the resulting linear system using the Cholesky decomposition. We opted not to employ parallelization for the sake of simplicity. However, readers may observe that for achieving more accurate results, such as using a denser grid around the boundary of the object and thereby expanding the size of the discrete domain, parallelization becomes essential[1]

Implementing the numerical scheme outlined in Section 2.2 we have the opportunity to explore the dynamics of fluid flow around various objects. First, we want to illustrate the emergence of the phenomenon before delving in different factors on the flow as differences in Reynolds numbers or in the shape of the object.

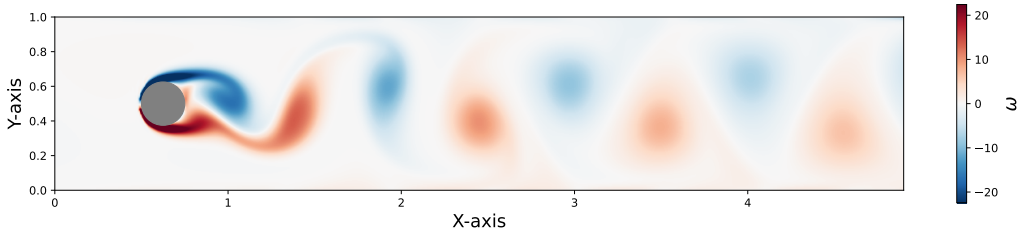
3.1 Observing von Kármán vortex street

To investigate how the phenomenon emerge, we look at an exemplary showcase of a fluid with a Reynolds number $\mathrm{Re} = 500$. We simulate the fluid in a domain of the width $W = 1$ and the length $L = 5$. For most of the simulations we use a homogeneous grid with 500 grid points in the x direction and 100 grid points in the y direction. The time step is taken in an adaptative way, in order to ensure the CFL condition discussed in Section 2.2. On the right side of the domain we place a circle with radius $r = 0.125$ as an object and at the start of the simulation the fluid inside the domain is at rest. Imposing the boundary conditions as presented in Section 2.1,

[1] For further details on the implementation, we shared our code in this Github repository. There the reader will also find animations of the flow dynamics around different objects to better understand the phenomenon.



(a) Magnitude of the velocity $\sqrt{u^2 + v^2}$



(b) Vorticity ω.

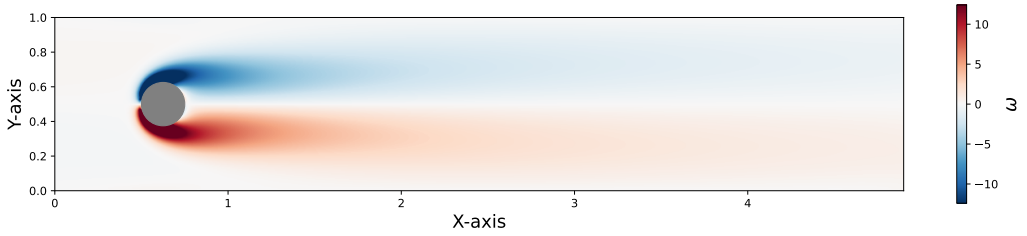
Figure 3: Flow dynamics around a circular obstacle in a fluid with Reynolds number Re = 500 at time $t = 25$. The top image displays the magnitude of the velocity $\sqrt{u^2 + v^2}$, and the bottom image focuses on the vorticity ω.

the fluid starts to move from the left side to the right side of the domain. As the simulation progresses, the flow dynamics evolve to reveal the organized structure known as a von Kármán vortex street. In Fig. 3, we present the outcomes of this prolonged simulation, showcasing the magnitude of the velocity field and the vorticity $\omega = \nabla \times \mathbf{u} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$, to visualize the vortices. The vorticity measures the local rotation of the velocity field. In the figure we can observe the characteristic periodicity between red (counterclockwise rotating vortices) and blue (clockwise rotating vortices) vortices. In front of the body the flow seems to be laminar with an income velocity of approximately $u = 1$ (after normalization), which is consistent with our boundary conditions.

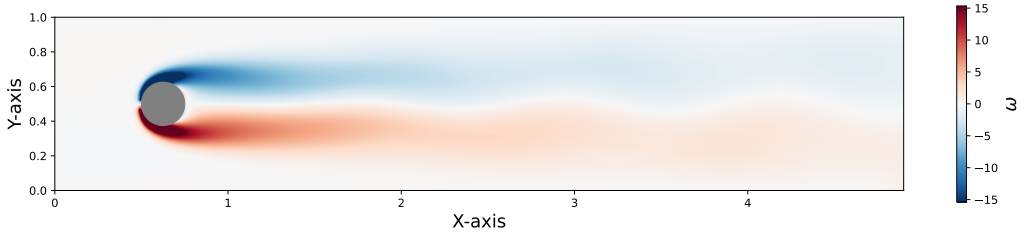
3.2 Influence of Reynolds number

The Reynolds number plays an important role in determining the flow characteristics around bodies in a fluid. It serves as a crucial indicator for the flow regime around bodies immersed in a fluid, significantly influencing the resulting flow patterns. In this section, we look at the same setup as before and vary the Reynolds number. The impact can be categorized in different regimes. In Fig. 4 we plot the vorticity of the flow for different Reynolds numbers.

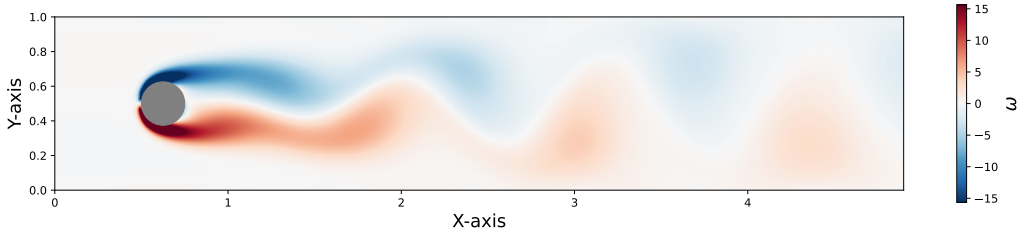
The impact of the Reynolds number is evident. For low Reynolds numbers (Re $\leq$ 150) the flow stays laminar, and pattern formation is not observed. For Re $\approx$ 175 the flow starts to be more complex. We don't observe vortices, but behind the body the vertical velocity periodically changes. Increasing the Reynolds number further we observe that the periodic perturbation become stronger (Re = 190), until finally vortices emerge (Re $\geq$ 230).



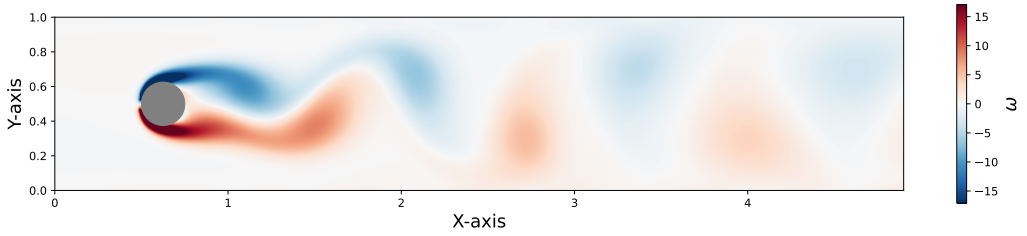
(a) Vorticity ω for Re = 100



(b) Vorticity ω for Re = 175



(c) Vorticity ω for Re = 190



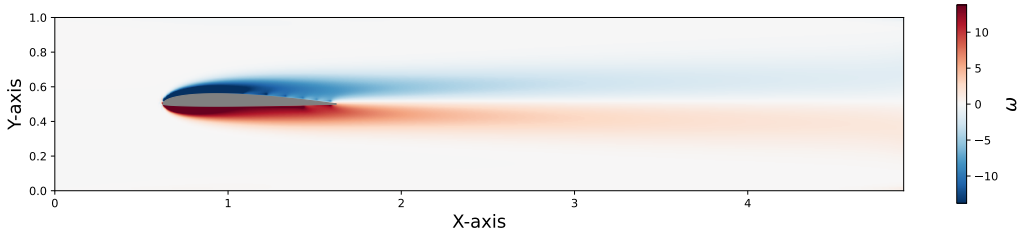
(d) Vorticity ω for Re = 230

Figure 4: Flow dynamics around a circular obstacle for fluid with different Reynolds numbers at time $t = 30$.

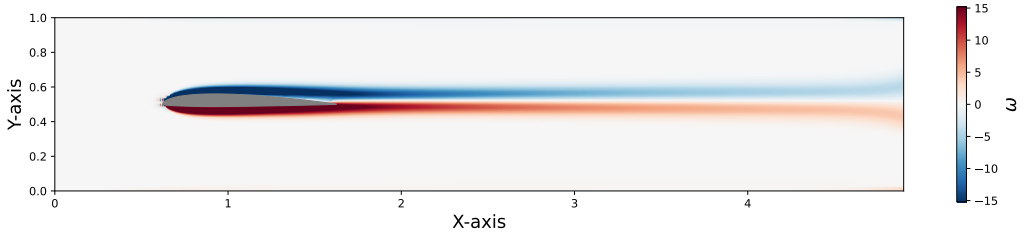
3.3 Influence of the object

We this section we aim to demonstrate the impact of object geometry on the flow characteristics. To show that the geometry of objects immersed in a fluid significantly influences the flow patterns that develop around them, we look at a more advanced shape: an airfoil. This shape is not just designed to create lift for a plane, but also to ensure that the flow patterns are stable and manageable. This stability is essential for effective maneuverability during flight.

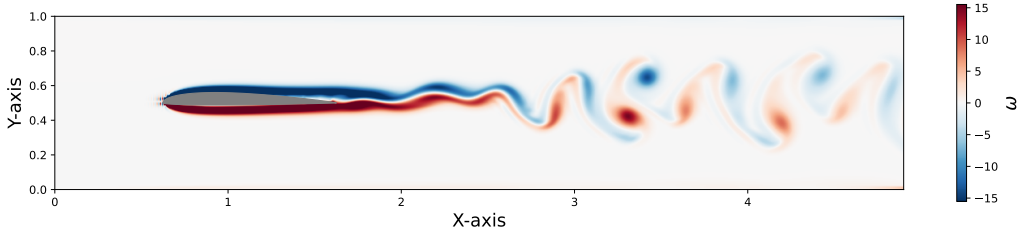
In Fig. 5a, we present our findings for the airfoil subjected to a Reynolds number Re = 500. In contrast with the observations made in Fig. 3a for the circle, the flow surrounding the airfoil is remarkably devoid of vortices or significant flow perturbations. The elongated shape reduces the region in which the side eddies can interact



(a) Vorticity ω for Re = 500



(b) Vorticity ω for Re = 3500



(c) Vorticity ω for Re = 5500

Figure 5: Flow dynamics around an airfoil in a fluid with different Reynolds numbers at time $t = 30$.

and delays the onset of mixing to a position much farther behind the object. This difference demonstrates the influence of object geometry on flow dynamics. The streamlined shape of the airfoil leads to smoother flow, reducing the tendency for vortex formation.

It is important to note that comparing flows around bodies of dissimilar shapes at equivalent Reynolds numbers is inherently challenging. This challenge arises due to the variability in characteristic lengths and velocities, which fundamentally alters the flow's nature around each object. But even increasing the Reynolds number to Re = 3500, does not lead to turbulences in the flow. As we can see in Fig. 5b the flow around the airfoil maintains a predominantly laminar characteristic, a testament to its aerodynamically efficient design.

Only when we further increase the Reynolds number to Re = 5500, the flow around the airfoil begins to exhibit clear signs of turbulence, marked by the formation of vortices. In Fig. 5c we plot the results and we can clearly see vortices arising behind the foil. The circle, with its inherent propensity to induce turbulence, resulted in flow conditions too chaotic for our solver to accurately simulate over extended durations.

In the case of the airfoil we see that the influence of the shape on flow dynamics

is very important. This underscores the significance of the complexity in designing aircraft that must operate effectively across a broad range of speeds. As a result, the design of wings is not only highly optimized but, in certain cases, incorporates the ability to change shape dynamically to adapt optimally to the aircraft's velocity.

4 Conclusion

In this report we numerically analysed the von Kármán vortex street. We successfully built a solver which leverages Chorin's method to simplify the integration step and breaks it down in smaller steps, which are easier to solve. The method then simplified to treating the advection with a second order Semi-Lagrangian scheme, the diffusion with a second order central difference scheme and the Poisson equation for the pressure with a 5-point stencil method. To achieve great flexibility for the possible shapes inside the domain, we impose the fluid at rest in the form of the shape. Basically, this corresponds to imposing a no-slip condition at the boundary of the shape.

In the second part of our approach we numerically investigated the emergent flow patterns around different objects. First, we exemplary showed the emergence of the von Kármán vortex street in case of a circle placed on the left side of the domain for fluid with intermediate Reynolds numbers. As expected we observe the alternating vortices behind the body. We identified a critical Reynolds number which indicates the dynamics of the flow. For smaller Reynolds numbers the flow is laminar and no vortices emerge. For higher numbers periodic formations emerge, which finally lead to mixing and vortices shredding. Finally, we demonstrated the impact of the shape on the flow. By looking at an airfoil we showed that with intelligent design of the shape the dynamics of the flow can be greatly influenced. With this new shape only with much higher Reynolds numbers we observed vortex shredding and thereby showcased the importance of fluid- and aero-dynamic favorable design. Especially for planes, ships or turbines the shape has a great impact on performance and efficiency, among other essential performance metrics.

References

[Boy01] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. 2nd edition. Dover Publications, 2001.

[Cho67] A. J. Chorin. "The numerical solution of the Navier-Stokes equations for an incompressible fluid". In: *Bull. Am. Math. Soc.* 73.6 (1967), pp. 928–931. DOI: 10.1090/S0002-9904-1967-11853-6.

[DL07] T. Dupont and Y. Liu. "Back and forth error compensation and correction methods for semi-Lagrangian schemes with application to level set interface computations". In: *Math. Comput.* 76 (Apr. 2007), pp. 647–668. DOI: 10.1090/S0025-5718-06-01898-9.

[Eig] Eigen. *Eigen: C++ Template Library for Linear Algebra*. Accessed: 04-03-2024. URL: ↗.

[Fad+00] E. A. Fadlun et al. "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations". In: *Journal of Computational Physics* 161.1 (2000), pp. 35–60. DOI: 10.1006/jcph.2000.6484.

[Pes77] C. S. Peskin. "The Immersed Boundary Method: Numerical Solutions of PDEs Involving Fluid Flow and Interacting Boundaries". In: *Journal of Computational Physics* (1977).

[SG00] H. Schlichting and K. Gersten. *Boundary-Layer Theory*. 9th. Springer, 2000. DOI: 10.1007/978-3-662-52919-5.

[Wik] Wikipedia contributors. *Kármán vortex street*. Accessed: 04-03-2024. URL: ↗.