PIV-based surface pressure reconstruction using Physics-Informed Neural Networks

Víctor Cayetano Hernández Sánchez

Delft University of Technology

PIV-based surface pressure reconstruction using Physics-Informed Neural Networks

by

Víctor Cayetano Hernández Sánchez

S.N: 6079911

Tutor 1:	Dr. Andrea Sciacchitano
Tutor 2:	Dr. Anh Khoa Doan
Project Duration:	Sep, 2024 - April, 2025
Faculty:	Faculty of Aerospace Engineering, Delft
Department:	Aerodynamics & Wind Energy

Cover: Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Acknowledgements

The present Thesis is dedicated in its entirety to my grandfather, Emilio Sánchez Díaz.

Thank you for being a true role model for our entire family, your strong work ethic and honest values have made us who we are and opened up doors we never imagined possible. You can rest assured we will carry these with pride for the years and generations to come.

Thank you.

Víctor Cayetano Hernández Sánchez
Delft, July 2025

Abstract

Static pressure is a scalar magnitude that expresses the force per unit area exerted by a fluid at rest. As such, it constitutes one of the two mechanisms through which fluid flows generate forces on bodies. Moreover, static pressure is not only relevant in the definition of surface loads, as it plays a key role in a number of fields, such as turbulence research due to its impact on the amplification or damping of turbulent flow instabilities, or medical research, provided its key paper on cardiovascular disease.

Accordingly, different approaches exist that allow to obtain pressure information in Fluid Dynamics applications. Among these, simplified analytical models, Computational Fluid Dynamics and experimental measurements stand out given their extensive use. While each of these comes with its own advantages and limitations, the latter typically offers the advantage of being conducted with real flows, hence providing a reliable source of information if proper similarity parameters and set-up are achieved.

While there exist different techniques to measure pressure experimentally, among which pressure tapping and pressure sensitive paint stand out, these present major limitations, such as the limited spatial resolution that can be achieved without intrusion effects or the challenges encountered during calibration, respectively. Consequently, more recent methods have been developed that allow to reconstruct pressure from velocity fields obtained via Particle Image Velocimetry. An instance of such algorithms is the Poisson Solver, which is based on the application of the incompressible relation to the momentum conservation equations, yielding a boundary-value problem for pressure. Nonetheless, while this approach benefits from the instantaneous and simultaneous nature of PIV measurements, it presents its own challenges, among which the propagation of noise from the velocity field into the reconstructed pressure field stands out.

More recently, with the proliferation of Machine Learning, the number of applications in Fluid Mechanics has grown. In particular, an approach that stands out are Physics-Informed Neural Networks, which optimize Deep NN models minimizing a loss function with contributions from labeled flow data variables and residuals from physical equations, thus learning the flow field variables. While diverse use cases have been reported for *PINNs*, such as the generation of reduced order models or the direct simulation of flows, especial emphasis has been placed in research on their ability to infer unsteady or mean pressure fields from velocity measurements, via the application of the Navier-Stokes equations.

Even if research has shown *PINNs* offer key advantages with respect to traditional pressure reconstruction methods, such as robustness to Gaussian noise or lack of discretization errors, analysis of the research available highlights key areas that require further exploration in the establishment of *PINNs* as a reliable alternative to traditional methods. Specifically, the study of *PINNs* performance with real experimental data and its comparison with solvers as the Poisson against direct experimental measurements is of paramount relevance, provided that the vast majority of publications concern the use of artificial experimental data from CFD simulations.

In accordance, a *PINN* framework has been developed and its accuracy in the reconstruction of surface pressure has been tested using time-averaged data from both CFD simulations and experimental tests of the two-dimensional flow around a cylinder. Particularly, comparison of the *PINN* and the Poisson surface pressure reconstructions with pressure tap data showed superior performance of the former, with respective MSE reductions of -1% and -21% for the flows around a smooth cylinder and one fitted with zig-zag strips at $\theta = \pm 45^\circ$.

Additionally, sensitivity studies to understand the effect of various parameters in the *PINN* training process has resulted in the identification of various trends. Among these, especial attention is required by the ability of *PINNs* to add regularization in areas affected by correlated noise such as reflections via the addition of collocation points, where the PDE loss is evaluated. Further noteworthy findings concern the benefit of using physical boundary conditions at solid surfaces in the form of the no-slip, no-penetration and no-fluctuations constraints. In this study, it has been proven that these allow not

only to bypass non-physical pressure fluctuations that derive from spatially-correlated noise, but also to reduce surface pressure reconstruction error when data gaps exist close to the surface, achieving reductions of up to -92% for a radial data gap of 75% oof r_{cyl} from the cylinder surface. Finally, it has been shown that, provided that the dataset contains points that allow to define a reference pressure value, the provision of sparse pressure tap data to the NN during training results in local, rather than generalized, error reductions.

In conclusion, the sensitivity studies carried out on the smooth cylinder dataset have resulted in pressure MSE reductions as substantial as -50% with respect to the Poisson solver when both are compared with static pressure tap data, supporting the establishment of *PINNs* as an alternative method to conventional pressure reconstruction algorithms as the Poisson solver, despite the time penalty that these can represent given the instantaneous nature of the latter. On the qualitative side, it has been proven that *PINNs* can provide a flexible framework to embed prior knowledge of the solution, such as the positive nature of normal Reynolds Stress components or boundary conditions. Along the same lines, it is shown throughout the thesis how *PINNs* can be used to perform debugging steps that allow to identify sources of error.

Contents

Nomenclature

If a nomenclature is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

Abbreviations

Abbreviation	Definition
SI	International System of Units
CFD	Computational Fluid Dynamics
LES	Large Eddy Simulation
DNS	Direct Numerical Simulation
NN	Neural Network
DNN	Deep Neural Network
PINN	Physics-Informed Neural Network
RMS	Root Mean Square
MSE	Mean Squared Error
PDE	Partial Differential Equation

Symbols

Symbol	Definition	Unit
Re	Reynolds Number	[-]
M	Mach Number	[-]
C_p	Pressure Coefficient	[-]

1 Introduction

1.1. The relevance of pressure in Aerodynamics

Static pressure is a scalar quantity that describes the force per unit area exerted by the flow on a given surface [4]. As such, it is one of the two known mechanisms through which fluids can exert forces over a surface: pressure and shear, with units of force per unit area ($[N/m^2]$ in the SI).

While the latter mechanism derives from the viscous nature of real flows and acts tangentially to the surface, pressure results from the temporal rate of change of the momentum contained in fluid particles. Furthermore, since pressure can be exerted by fluid parcels irrespective of their motion, it follows that it must act normal to surfaces [28].

It then becomes central to reflect on the relevance of surface pressure on aerodynamic phenomena from an analytic and practical standpoints:

- Pressure is one of the two primary mechanisms for generating aerodynamic forces. Consequently, it is a key variable in the design and certification of aerodynamic components across various engineering fields, including aerospace, wind energy, automotive and architecture.

  Additionally, pressure is not only relevant in the reconstruction of local loads but its distribution is also crucial to define aerodynamic moments and force distributions. The latter gains particular relevance in defining flight and vehicle mechanics as well as in the design and optimization of structural components.

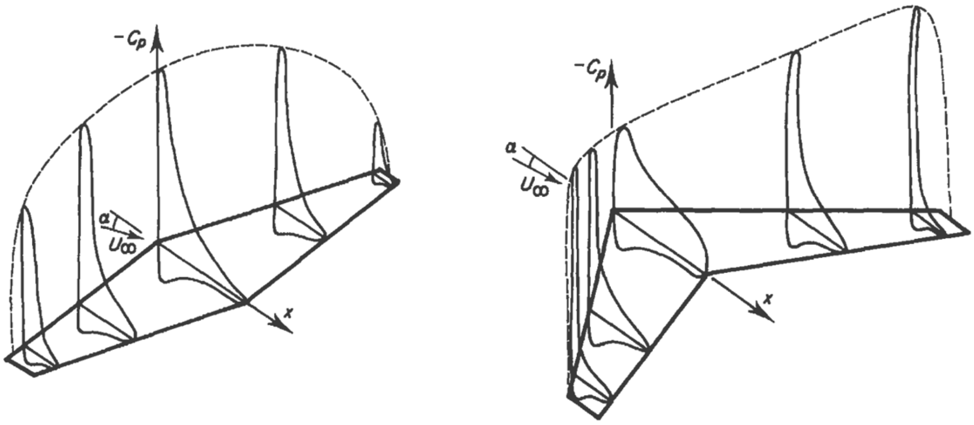


Figure 1.1: C_p distribution along spanwise stations for an unswept (L) and swept-back (R) wings. [28]

An example of this is shown in Figure 1.1, where the chord-wise C_p distribution of two wings have been plotted at various spanwise locations. As shown, aft sweep induces a non-uniform variation in spanwise loading. This will have major implications for structural design and stall behavior. The peakier C_p distributions on the swept case suggest a higher likelihood of sudden leading edge stall compared to the unswept wing, highlighting the relevance of estimating not only total loads but also their distribution in various engineering applications.

Further support to the relevance of surface pressure in the above case can be found considering the definition of pressure coefficient in Equation 1.1, which expresses a dimensionless pressure difference to the reference value.

$$C_p = \frac{p - p_\infty}{q_\infty} = \frac{p - p_\infty}{\frac{1}{2}\,\rho\,U_\infty^2} \tag{1.1}$$

This expression can be rewritten as a function of velocity via the Bernoulli relation (Eq. 1.3) assuming total pressure is conserved. The resulting expression is included in the term at the center of Eq. 1.2. Additionally, for potential flows, validity of the linearity and superposition principles allow to rewrite its definition as a function of the local super-velocity, $\Delta\, U \;=\; U - U_\infty$, as per the RHS of the same equation.

$$C_p = 1 - \frac{U^2}{U_\infty^2} \approx -2\frac{\Delta U}{U_\infty} \tag{1.2}$$

Accordingly, in transonic aircraft design, span-wise and chord-wise C_p distributions play a crucial role in defining the location and strength of shock-wave formation [75], largely impacting cruise drag and fuel consumption.

- On the other hand, pressure is a key factor both on the surface and off-body in the production and evolution of turbulence, making it relevant in research fields too.

  Specifically, wall pressure gradients are critically influential in the amplification or damping of turbulent instabilities, thus affecting the laminar-to-turbulent transition process. Favorable pressure gradients ($\partial p/\partial x < 0$) delay the formation of inflection points in the wall-normal velocity profile, postponing transition [77].

  As a result, it becomes visible why surface pressure measurements gain importance in flow control applications and other research fields, contributing to the understanding and control the turbulent transition process.

- Another application where surface pressure data can be critical is in the medical field. For instance, [67] examines the equilibrium between blood pressure and wall shear stress in arterial walls as one of the principal mechanisms for cerebral aneurysm growth. This highlights the necessity of pressure data even in conditions where the use of experimental measurement techniques is challenging.

  Beyond hemodynamics, pressure is also a central factor in the study of airway obstruction and resistance. In [68], peak airway pressure, defined as the pressure required for air to move through the lungs while inhalating in mechanical ventilation, is seen as a key performance indicator to analyze the effects of different parameters (inflow velocity, airway diameter, etc.) on airway resistance, as this shows a direct proportionality to peak pressure requirements.

Provided the above reasons that highlight the relevance of pressure in the context of Fluid Mechanics, the present thesis aims to assess the use of Physics-Informed Neural Networks as a tool to obtain time-averaged pressure data from experimental measurements of the mean velocity field, placing special emphasis on surface pressure.

1.2. Traditional techniques to obtain static pressure

With the aim to obtain pressure data in fluid-dynamic applications, different methods exist with their own limitations and levels of fidelity to reality. These include:

1.2.1. Analytical Methods

One of the techniques used in preliminary studies, as well as academia due to its simplicity is the analytical approach. This generally involves the evaluation of simplified physical models that approximate the flow solution under a given set of assumptions.

As a result of these simplifications, the complexity of the expressions to be evaluated is reduced, which results in an efficient way to compute approximate flow solutions. On the contrary, such assumptions can limit their applicability to very specific flows or even only in certain regions. Consequently, one of the core challenges is to understand the application envelope of such techniques.

An example of the above are cases which respond to inviscid flows [6], where the effect of shear stresses is assumed to be negligible. This simplification is theoretically feasible as the *Reynolds number* tends to infinity (i.e. $Re \to \infty$).

Furthermore, inviscid flows that are incompressible ($M \leq 0.3$ [23]) and irrotational are especially attractive, since their velocity fields can be expressed through a single scalar quantity, the velocity potential [5]. This facilitates the construction of analytical expressions for the velocity field of such *potential* flows as exposed in [4], where the four canonical potential flows, namely the uniform, the source (sink), the doublet and the vortex flow solutions are developed.

Although potential flow models fail to accurately predict drag forces due to obviating shear stresses, they allow point pressure to be inferred from local speed and total pressure at infinity by assuming conservation of the latter as per Bernoulli's equation [34], offering a simple method to yield the pressure field.

$$p_{\infty} + \frac{1}{2}\,\rho\,U_{\infty}^{2} = p_{p} + \frac{1}{2}\,\rho\,u_{p}^{2} \tag{1.3}$$

While these assumptions appear to limit the applicability of potential flow models, certain properties make them extensively used. On the one hand, dimensional analysis of the wall normal momentum conservation equation for a Boundary Layer Flow results in the conclusion that pressure remains virtually constant across an attached boundary layer [64]. Furthermore, the linearity and superposition characteristics of the velocity potential permit complex flow solutions to be produced as the sum of their simpler forming parts. As a result, potential flow models such as panel methods [35] are useful to define approximate surface pressure distributions for real attached flows.

However, this family of models is not as extensively used in more advanced engineering or research fields because of the difficulty it presents in adding specific boundary or design conditions, such as geometric modifications, but it can still prove useful in providing reference results for more advanced studies.

An example of this is the potential flow solution around a non-lifting cylinder, which can be found from the expression for the doublet flow in a uniform flow. Notably, the doublet flow arises from the addition of the canonical flow solutions of a source and a sink at an infinitesimally small distance [4]. The resulting expression for the velocity potential distribution is given by Equation 1.4, where R stands for the cylinder radius, r is the radial coordinate and θ is the angular coordinate with origin at the back of the cylinder.

$$\phi\,(r,\,\theta)\, = U_{\infty}\,\cdot\,r\,\cdot\,cos\,\theta\left(1 + \frac{R^{2}}{r^{2}}\right) \tag{1.4}$$

Furthermore, this can be evaluated at the cylinder surface where $r = R$ to yield the expression for the angular velocity, as the no-through flow condition causes the radial velocity, U_r, to be null:

$$U_\theta|_{r=R} = \frac{1}{r}\left.\frac{\partial \phi}{\partial \theta}\right|_{r=R} = -2\,U_\infty sin\,\theta \tag{1.5}$$

Finally, one can find the expression for the pressure coefficient on the surface using Bernoulli's relation with $u_p = U_\theta$, which will be used in later sections and has been represented in Figure 1.2.

$$C_p|_{r=R} = 1 - 4\,sin^2\,\theta \tag{1.6}$$

To provide further information on the applicability of the above equation, experimental results from [1], conducted in a pressurized wind tunnel featuring a $0.5 \times 0.9\,m$ closed test section and with $Re = 3.6 \times 10^6\,[-]$ have also been represented in Figure 1.2.

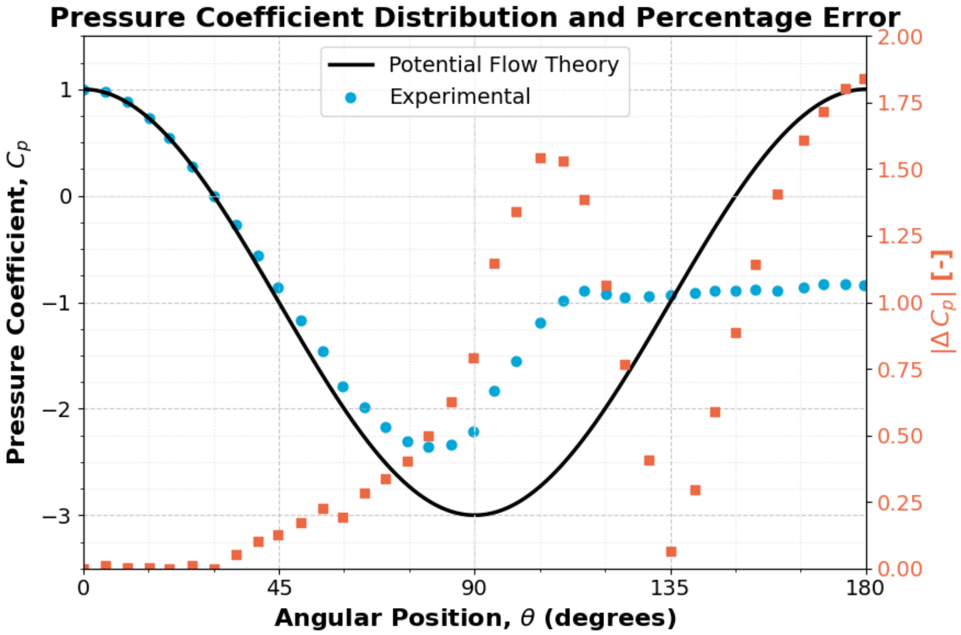


Figure 1.2: C_p distribution around a smooth 2D cylinder - potential flow vs experimental data from [1]

Examining the C_p distribution around a smooth 2D cylinder, which will be discussed throughout the thesis, the stagnation point can be identified by $C_p = 1\,[-]$ at $\theta = 0°$. This point is characterized by having zero dynamic pressure, causing static pressure to equal total pressure. At this location, the experimental and potential flow theory results match perfectly, since the high-Re regime of the experimental test causes viscous effects to be negligible locally. As θ grows, the flow is accelerated due to surface curvature, causing pressure to decrease through the relation in Equation 1.3, in this region correlation between both results is great, with a slight but progressive deviation from $\theta = 45°$ due to minimal loss of total pressure through viscous effects.

Regarding suction peak magnitude and location, the potential flow continues to accelerate to $\theta = 90°$ with a $C_{p,min} = -3\,[-]$. On the other hand, the experimental measurements show a $C_{p,min} \approx -2.4\,[-]$ at $\theta \approx 80°$, reflecting a modeling error of around $\Delta C_p \approx 20\%$ and $\approx 10\%$ on suction peak location when neglecting viscous effects.

Beyond the suction peak, the potential flow solution is perfectly symmetric as it assumes the flow stays attached throughout the cylinder with no total pressure loss, as given by $C_p = 1.0\,[-]$ at $\theta = 180°$. In contrast, the real flow exhibits signs of separation at $\theta \approx 115°$ as shown by the inflection point on C_p and virtually constant C_p beyond that location.

Overall, observing the absolute deviation between real and model results, one could argue that an acceptance criterion with $\Delta C_p \leq 0.1 \times C_{p,min,pot}$ would make the model valid up to $\theta \approx 70°$, showing its limited suitability for the prediction of suction peaks and separation.

1.2.2. CFD Simulations

Going one step further in complexity and freedom to implement case-specific features come CFD simulations. While meshless CFD models such as those based on Smoothed Particle Hydrodynamics exist [43], the most extended CFD tools consist of the application of the governing equations to the elements of the discretized domain. Subsequently, iterative numerical schemes are utilized to minimize the residuals of such governing equations (typically mass, momentum and energy conservation) until the specified convergence criteria are met.

Furthermore, several steps exist in complexity for turbulent flow simulations, which can provide enhanced correlation to reality and further insight into the production and evolution of turbulent structures.

Highest on the modeling versus computational effort scale are RANS models, which do not represent but rather simulate the effects of all turbulent scales through the Reynolds-Averaged conservation equations. This simplification assumes that flow variables can be decomposed into an averaged (e.g. $\overline{u}$) and fluctuating (e.g. u') components, leading to the reformulation of the conservation equations in their Reynolds-Averaged form, where fluctuating components are the modeled ones, as they derive from turbulence.

Applying Reynolds decomposition to the momentum conservation equation and temporally averaging it yields Equation 1.7, where the only surviving fluctuating component terms are the so-called Reynolds Stresses, which reflect the rate of energy drain from the mean flow into the turbulent scales. The need to model them has given rise to a number of approaches, from eddy-viscosity models (EVMs) to Reynolds Stress Models (RSMs).

$$\overline{u_j}\,\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_i' u_j'}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu\left(\frac{\partial^2 \overline{u_i}}{\partial x_j^2}\right) \tag{1.7}$$

On the one hand, EVMs model the effect of turbulent scales through analogy to a viscous-dissipation term, contained in the eddy viscosity - ν_T. Besides, a variety of models exist depending on the number of additional transport equations – ranging from Prandtl's zero-equation mixing length model, that yields a single approximate value for ν_T, to one and two-equation models, these adding further terms like the turbulent kinetic energy, k, and turbulent ($k-\epsilon$) and specific turbulent ($k-\omega$) dissipation rates.

On the other hand, Reynolds Stress Models attempt to add one conservation equation for each of the Reynolds Stress Components and the dissipation rate. Therefore, they require four or seven additional transport equations in two and three-dimensional problems respectively, performing better in anisotropic flows at a higher computational cost [29].

While RANS turbulence treatment is the most extended in engineering applications due to its simplicity and computational efficiency, it becomes apparent that its time-averaged nature and approach to modeling limit its ability to accurately represent flows with strong transient dynamics or anisotropy and make it challenging to represent turbulent structures [37]. As a result of this limitation, alternative approaches have emerged such as Large Eddy Simulations and Direct Numerical Simulations, which respectively solve the largest scales or the entirety of the turbulence spectrum [56].

1.2.3. Experimental Simulations

Finally, experimental simulations are another alternative to obtain information about the flow. In this case, real flows are generally used in controlled environments such as wind tunnels to analyze the flow around scaled down or real-sized models of the geometry under study. Due to the ability to capture real flow behavior as well as the ability to conduct parametric studies in an efficient manner, experimental flow analysis is at the core of every research or engineering design process.

However, physical modeling errors are still present in wind tunnel testing, such as wall interference or the challenge of matching the most representative dimensionless numbers that characterize the flow regime across many engineering fields. Take, for example, the flow around an A-320 wing, which has a full-scale chord length of $3.6\,m$ and a flight speed of around $250\,m/s$ [3]. In this case, the two most relevant dimensionless numbers for aerodynamic performance are the Reynolds number,

$$Re = \frac{\rho\,U\,L}{\mu} = \frac{0.320\,kg/m^3 \cdot 250\,m/s \cdot 3.60\,m}{1.42 \cdot 10^{-5} N\,s/m^2} \approx 20 \cdot 10^6\,[-], \tag{1.8}$$

which will govern the turbulent flow behavior and depends on the reference speed, U, and length, L, as well as the fluid properties, ρ and μ, and the Mach number

$$M = \frac{U}{a} = \frac{250\,m/s}{290.6\,m/s} = 0.86\,[-], \tag{1.9}$$

which depends on the reference speed and the speed of sound, a, affecting compressibility effects such as shock wave pattern formation and wave drag. Attempting to match both of these conditions would be impossible even in one of the most capable atmospheric wind tunnel facilities in Europe, such as the *DNW*. This wind tunnel features a test section of 6 by 6 m^2 and a speed capability up to 145 m/s [66]. For example, using a test model with a scale of 1:7 to ensure the 34 m wingspan fits the test section would yield a maximum Re and M of:

$$Re|_{WT} = \frac{1.225\,kg/m^3 \cdot 145\,m/s \cdot 0.514\,m}{1.8 \cdot 10^{-5} N\,s/m^2} \approx 5 \cdot 10^6\,[-] \tag{1.10}$$

$$M|_{WT} = \frac{145\,m/s}{340\,m/s} = 0.43\,[-] \tag{1.11}$$

According to the above, it is observed that neither of the two full-scale dimensionless parameters can be attained with such test conditions and tunnel specifications. In the case of the Reynolds number, while the flow regime might not be extremely different due to both values being in the turbulent regime, a four-fold factor will certainly cause variations in the boundary layer thickness and location of the separation points. These two phenomena are of major relevance in aircraft design, not only in the prediction of accurate total force and moment coefficients but also in properly capturing interactions between different aircraft components.

On the other hand, the mismatch in Mach number is likely to impact performance to a larger extent. While $M = 0.43$ is very close to the $M = 0.3$ value, generally considered as a boundary below which the effects of compressibility are limited, $M = 0.86$ is much closer to the sonic regime. Accordingly, due to curvature-induced flow acceleration, shock waves are likely to appear, and so is the transonic buffetting phenomenon, of paramount relevance in terms of aerodynamic and structural aircraft certification.

The above exemplifies the challenges associated with dimensionless parameter matching. Among the ways to reduce them are pressurized wind tunnels, to increase air density through pressurization, and cryogenic wind tunnels, using liquid nitrogen to reduce air temperature, and hence air viscosity as well as its density, to a lower extent. However, each experiment and application usually finds its sweet spot at a different compromise between scale, speed, blockage factor, and other parameters depending on the research goals, always accounting for some level of uncertainty and error with respect to full-scale physics.

Finally, the controlled nature of the wind tunnel environment, with turbulence intensity levels as low as 0.1% like in the Low-Speed Lab facility at TU Delft, make experimental simulations extremely suitable for flow characterization using sensors of various types. This is because flow-induced phenomena can be discerned properly from environmental effects, as opposed to free-flight tests.

Experimental Measurement Techniques

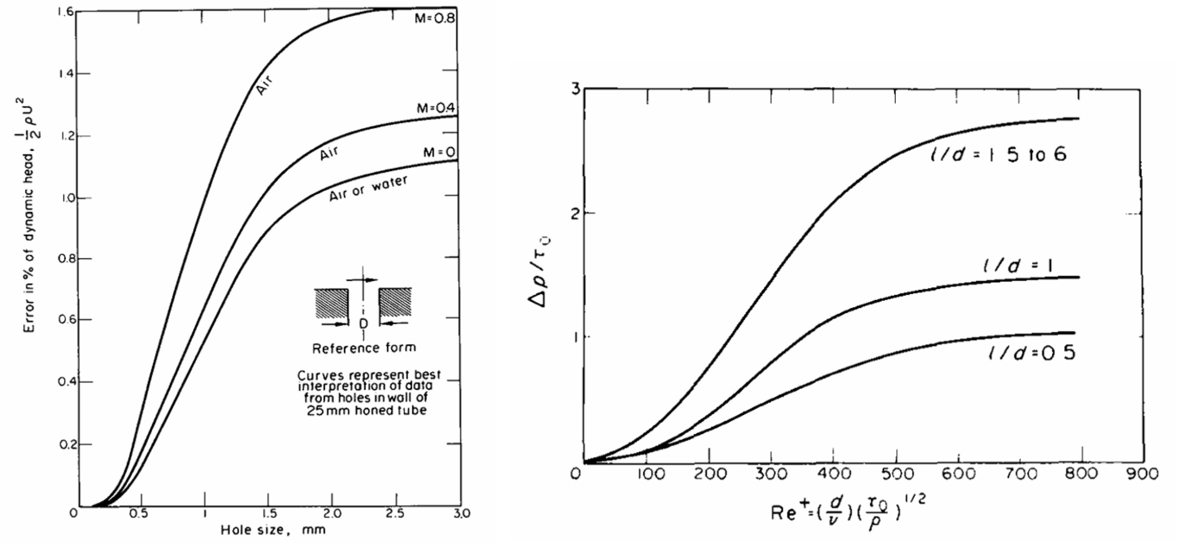
As a result of this suitability of experimental testing to capture flow behavior, a number of techniques have been developed to measure static pressure around testing objects. Focusing on the measurement of surface pressure due to its relevance in this thesis, the two most widespread techniques are pressure tapping and pressure sensitive paint.

Pressure Taps

The most commonly used method to obtain pressure information on the surface is through pressure tapping, which consists of drilling a cavity perpendicular to the surface and placing a pressure transducer locally or connecting it to a pressure scanner by means of a soft piping system. The $90°$ angle of the cavity to the local flow is of paramount importance as it aids to avoid the mis-classification of dynamic pressure as static pressure.

Furthermore, provided that the intent is to measure the pressure around the object being analyzed, research has been conducted to weigh and minimize the impact of each step in the measurement process. As a result, variables such as pressure tap size, geometry and finish, as well as the associated Reynolds number have been found to be determinant [48]. While the intent is not to provide a detailed insight into these, it is worth exposing the magnitude of the errors that can be incurred in by some of these variables, as this technique will be employed in the present work.

For simplicity and coherence with the thesis, the focus is placed on the very low-speed regime effects, and thus attention is brought to the $M = 0$ line in the graph on Figure 1.3a, which shows the effect of both compressibility and absolute hole diameter on the relative static pressure error. As observed, error remains negligible up to 0.5 mm hole diameter, having a seemingly constant slope up to hole sizes of 1.5 mm and plateauing at about 1.1% error above.



(a) Error as a function of compressibility and hole diameter.

(b) Error due to relative cavity length.

Figure 1.3: Pressure tap measurement errors from [17].

In addition, Figure 1.3b displays the effect of the relative cavity length as a function of the hole diameter Re, measuring the pressure error relative to the reference shear stress. As may be witnessed, larger relative pipe lengths have a detrimental effect, magnifying the S-shaped response with increasing Re.

While these errors are kept within certain boundaries, especially when appropriate manufacturing practices are followed, pressure taps find a major limitation in the spatial resolution they can provide. This is not only because of the need to have simultaneous acquisition, but also because increasing the spatial resolution comes at the expense of intrusion errors, thus possibly affecting the measured data.

This has also been studied in [38], where the effect of pressure tap diameter as well as that of single-chord versus multiple spanwise locations were studied for a pseudo-infinite DU96-W-80 airfoil. The results showed that while small for overall lift coefficient values, the local effects of pressure taps in regions of high curvature can lead to the formation of separation wedges. These have been highlighted in the flow-vis pattern of Figure 1.4, where one of the pressure taps closer to the leading edge appears to have caused a separation triangle.

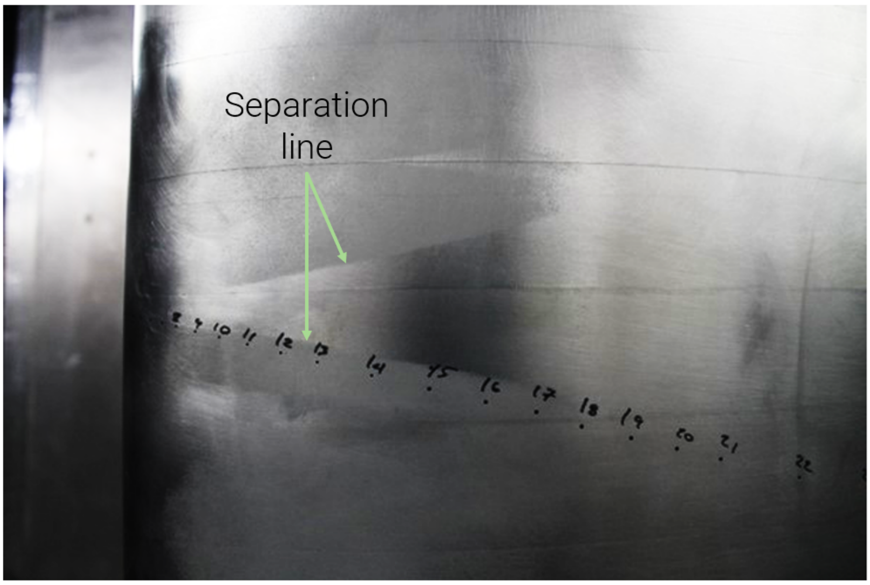


Figure 1.4: separation wedge in DU96-W-80 due to LE pressure tap - adapted from [38]

Further challenges presented by pressure tap measurements are the cost of manufacturing and accessibility. Although the first issue has seen a decrease thanks to the advent and proliferation of additive manufacturing techniques, it is still present in any test that requires drilling and piping. In terms of accessibility, this limits the type of bodies that can be fitted with pressure tapping systems due to a lack of thickness. This issue is especially pressing when one considers the widespread use of scaled-down models for wind-tunnel testing.

Alternatively, Pressure Sensitive Paint has been introduced as a method to acquire surface pressure information in experimental set-ups, overcoming some of the challenges of pressure tapping like intrusion and limited spatial resolution.

Pressure Sensitive Paint

Pressure Sensitive Paint or *PSP*, presents a reduced cost of manufacturing processes compared to pressure tapping methods. This technique makes use of oxygen's ability to quench to organic luminophores [49] to define pressure. This is because oxygen pressure directly affects the extent of quenching and thus the higher the partial pressure, the smaller the luminosity of the UV-excited dye. Accordingly, pressure can be inferred from luminosity using the Stern-Volmer equation 1.12 [49]. It is noted that the coefficients in this equation depend on temperature and can be defined in a calibration set-up.

$$\frac{I}{I_0} = A(T) + B(T)\left(\frac{P}{P_0}\right) + C(T)\left(\frac{P}{P_0}\right)^2 + \cdots \tag{1.12}$$

In addition, hints of the set-up required for PSP measurements are given in Figure 1.5. In this arrangement, it is required that the test object is fitted with a binder and sprayed with the light-reactive dye, which is illuminated during the test with UV light. Parallel to this, the acquisition equipment is formed by the camera and computer, where the information is processed to yield the light intensity maps.

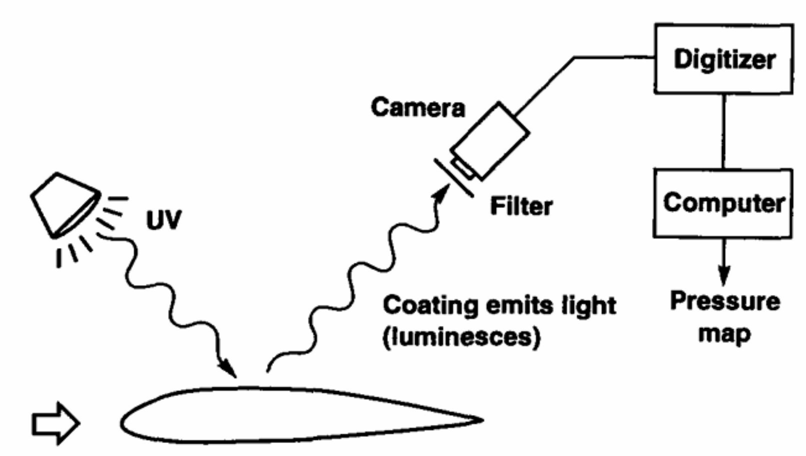


Figure 1.5: Pressure Sensitive Paint set-up and equipment - from [49]

Provided the above figure, it becomes obvious that, while this set-up addresses some of the key issues in pressure tapping, such as intrusion and spatial resolution, it presents a number of technical and application challenges. In particular, aeroelastic-induced deformations of the test body will result in variations not solely in the incident light intensity but also in the location of the test body with respect to the acquisition equipment. This is studied in [49], where the need to use reference points and algorithms to correct for these load-provoked modifications is highlighted.

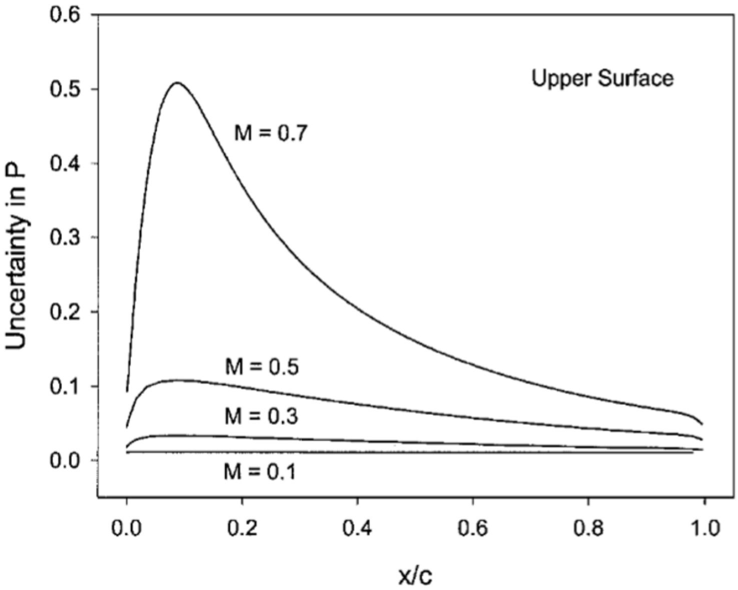


Figure 1.6: Pressure Sensitive Paint error relative to freestream pressure due to compressibility-related temperature [44]

Not only this, but also the speed envelope for applicability of the PSP technique could be rather limited, as is exposed in Figure 1.6, where the effect of the freestream Mach number on the uncertainty of the pressure distribution around the suction surface of a Joukowsky airfoil is presented. The rapid escalation of the uncertainty level with increasing M_∞ presents a bias around the suction peak and is stated to be related to thermal effects on *PSP*, with a peak of 50% uncertainty level around the suction peak at $M_\infty = 0.7$. Finally, some research [49] also challenges the idea that *PSP* is subject to some intrusion errors, as the *PSP* coating can cause dust particles entrained in the wind tunnel flow to adhere to the geometry under study, particularly affecting laminar-to-turbulent transition.

1.3. Pressure reconstruction from PIV - traditional methods

Overall, it has become apparent that conventional techniques to measure surface pressure experimentally present a number of limitations. Some of the most relevant ones are intrusion, lack of spatial resolution, or complexity of implementation.

As a result, more recent developments have focused on attempting to reconstruct pressure from alternative experimental techniques, such as Particle Image Velocimetry, which can provide time-averaged and/or time-resolved velocity fields.

1.3.1. Particle Image Velocimetry

Provided that this technique is utilized in the present work, its working principles, equipment, variables and output data are exposed here before introducing the conventional pressure reconstruction techniques from PIV data.

Firstly, a typical schematic of the set-up used in PIV experiments is displayed in Figure 1.7, where the wind tunnel test section is represented in light gray and the flow direction with blue arrows. It is noted that the example given corresponds to a set-up aimed at capturing two velocity components in two dimensions.

As observed, a pulsed laser device is fitted with a set of optical lenses to create the target laser sheet shape. In turn, this incident light illuminates the tracer particles of the flow, which need to have a certain set of characteristics to ensure they are representative of fluid motion [50]. Typical examples of tracer particles include fog, Helium-filled soap bubbles [63] and others, depending on the flow characteristics and light intensity requirements.

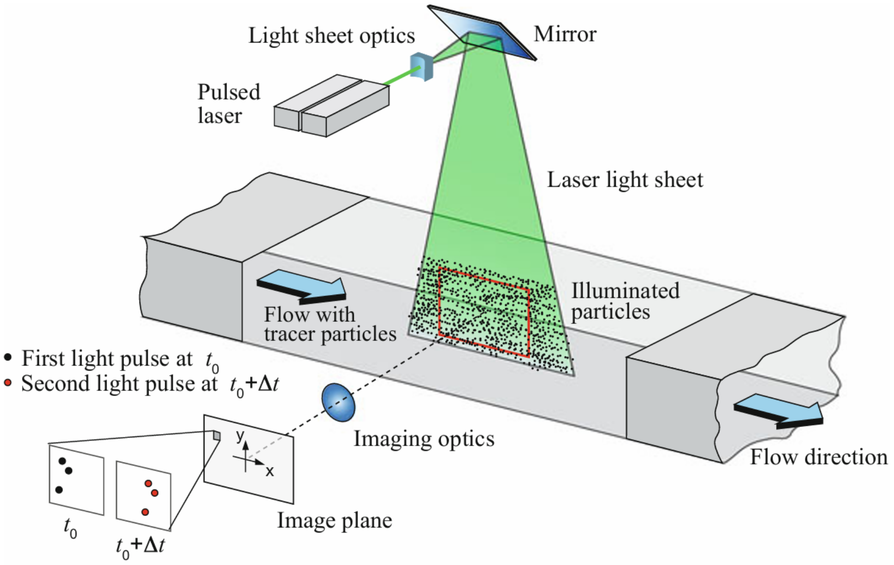


Figure 1.7: Example of set-up for two-dimensional, two-component PIV - from [57]

As the fluid flows around the test object, the laser firing is coupled with an image acquisition device that is placed normal to the illumination plane. The camera also features a set of lenses to achieve optimal imaging characteristics, controlling the amount of light captured and image particle size, among other parameters.

In order to measure flow displacement, an image pair is required, where two consecutive firings are performed in the order of a few microseconds [30]. This demands image acquisition technology that converts photons into electric charge in a low-latency framework, such as CCD and CMOS cameras [55]. This operation produces two images containing the light intensity at each pixel, which are processed to locate tracer particles, characterized by higher light intensity regions. To reduce uncertainty, these often benefit from batch-processing operations to remove background light and reflections [2].

The next step typically involves a correlation mechanism between the two images that form the pair in order to estimate particle displacement, for which spatial cross-correlation can be used, among other techniques [36].

In this algorithm, the first image in the pair is split using a windowing set of rules, including the window size and window displacement at each step. In addition, at each stage a correlation map is generated from the cross-correlation function (Equation 1.13). This can be seen as a similarity metric between the intensity map of the window in the first frame and the intensity map of a window of the same size shifted around the original position in the second frame of the pair.

$$\Phi\left(m,n\right) = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} I_1(i,j)\cdot I_2(i+m,j+n)}{\sqrt{\sum_{i=1}^{I}\sum_{j=1}^{J} I_1^2(i,j)\cdot\sum_{i=1}^{I}\sum_{j=1}^{J} I_2^2(i,j)}} \tag{1.13}$$

In the above equation, I_1 and I_2 denote the light intensity fields in the first and second frames. These are evaluated on the dummy indices $\{i, j\}$, which indicate the pixel location and range from $\{1, 1\}$ to the dimensions of the interrogation window, $I \times J$.

Finally, the output of the cross-correlation function is the pixel displacement yielding the highest correlation value for each window, Δ_{px}. In turn, this enables to compute the velocity vector in length units for that location through the magnification factor, M_f, and the time-step size between frames, Δt, as per Equation 1.14. It must be noted that, despite not being discussed here, there exist ways to achieve sub-pixel accuracy levels when estimating the particle displacement.

$$U = \frac{\Delta_{px}}{M_f \Delta t} \tag{1.14}$$

It is also noteworthy that the basic principles shown here to capture two velocity components across a planar domain can be extended to achieve more complex datasets. For example, a step in complexity adds a second camera and places both at a known angle, thus allowing one to determine the third velocity component for the particles in the laser sheet plane, which is termed stereoscopic PIV [57]. Finally, tomographic PIV uses laser volumes and at least four cameras to determine all three velocity components over the illuminated domain, or a portion of it, to then assemble the three-dimensional measurements [21].

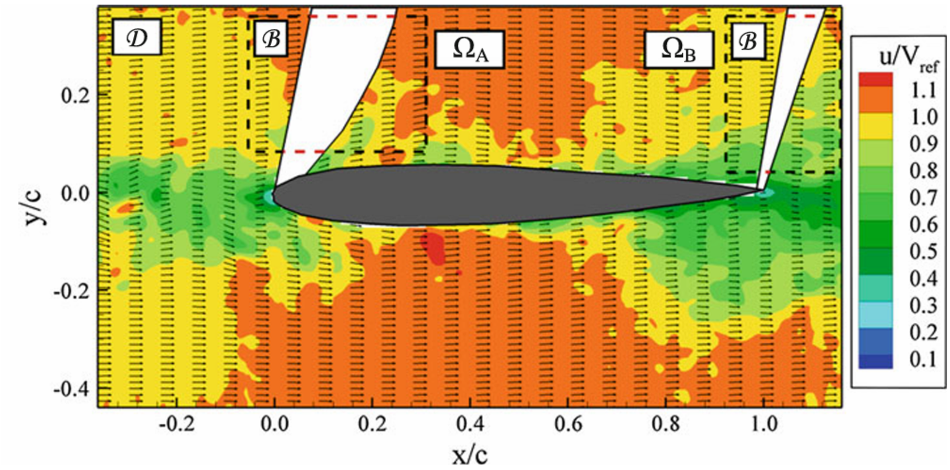


Figure 1.8: Normalized x-velocity of the flow around NACA 0012 via PIV - from [65].

An example of the data that can be obtained with PIV is shown in Figure 1.8, featuring the two-dimensional, normalized x-velocity profile around a NACA 0012 airfoil. This specific image has been selected due to the data gaps it contains around the leading and trailing edges, which appear due to the refraction of the laser-light rays across the curved plexiglass volume, showing one of the challenges found in PIV, as is the required direct view for the laser and imaging devices.

From the above explanation, it becomes clear that PIV presents solutions to several of the issues exposed for conventional experimental pressure measurement techniques. For example, it can provide instantaneous measurements of all velocity components simultaneously [78] and has the main advantage of not causing significant intrusion [2], compared to techniques that require the sensor to be immersed in the flow. Furthermore, its instantaneous nature and focus on tracer particles removes the calibration challenges in *PSP*.

1.3.2. Traditional Methods to infer pressure from PIV

In order to benefit from the advantages of PIV with respect to the other pressure measurement techniques, algorithms aimed at reconstructing pressure from the temporal [16] [45] [32] or time-averaged [54] velocity fields obtained with PIV have been developed. The techniques discussed here are conceptually similar because they are based on the principle of momentum conservation to obtain an expression for the pressure gradient.

Direct Integration of the Pressure Gradient

The first method exposed here yields an expression for the pressure gradient directly from the momentum conservation equation, either in its Eulerian (Equation 1.15) or Lagrangian (Equation 1.16) formulations, which correspond to a viewpoint that is stationary in space and moving with the flow, respectively.

$$\nabla p = -\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{u} - \nu \nabla^2 \mathbf{u} \right) = \mathbf{F} \tag{1.15}$$

$$\nabla p = -\rho \left(\frac{D \mathbf{u}}{D t} + - \nu \nabla^2 \mathbf{u} \right) = \mathbf{F} \tag{1.16}$$

In the above equations, the term on the left-hand side represents the pressure gradient, whereas on the term at the centre, the two formulations of the material acceleration component and the viscous term are shown. As noted in [54] or [45], while the contribution of the viscous term can be computed, it is generally a few orders of magnitude smaller than the other terms and can be neglected where the inviscid flow assumption holds.

In addition, it can be deduced that the right-hand side (*RHS*) represents a conservative vector field, F, which equals the gradient of a potential function, p, in this case. As a consequence of the gradient theorem [71], it can be stated that any line integral of the right-hand side from one point to another will equal the pressure difference between them.

Observing the above equations, one can see the term in the centre of Equation 1.15 can be evaluated directly using a finite-difference scheme in space and time with the velocity data obtained through PIV measurements. On the other hand, evaluating its Lagrangian counterpart requires that the fluid parcel trajectories be reconstructed. Nonetheless, this can be achieved directly from the velocity field and thus does not require particles to be tracked, as exposed in [46].

The last step consists of the integration of the pressure gradient term to yield the pressure field, which requires that a Dirichlet boundary condition is specified at least at one point. This is possible with limited uncertainty if selecting a reference point or region where viscous effects are limited, e.g. outside of shear layers, and applying the steady Bernoulli relation (Equation 1.3) or its unsteady counterpart (Equation 1.17).

$$p_\infty + \frac{1}{2}\rho U_\infty^2 = p_p + \frac{1}{2}\rho \left(\overline{\mathbf{u_p}} \cdot \overline{\mathbf{u_p}} + \overline{\mathbf{u_p}' \mathbf{u_p}'} \right) \tag{1.17}$$

It must be noted that a variety of methods have been proposed to carry out the integration step, such as space-marching integration [8] or the virtual-boundary omnidirectional integration algorithm presented in [46]. While the former is based on integration from neighboring points, the latter uses a more complex algorithm to produce integration paths between points placed on virtual boundaries, constructing the pressure on the internal nodes by averaging the pressure values yielded by all passing paths. Even if both have been used extensively, comparative studies such as [16] have proven the latter method is less sensitive to error propagation along the direction of integration.

Poisson Formulation

The alternative algorithm exposed here arises from the Poisson formulation of the momentum conservation equation, which derives from the application of the divergence operator to Equation 1.15:

$$\nabla \cdot \left\{ \nabla p = -\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{u} - \nu \nabla^2 \mathbf{u} \right) \right\}, \tag{1.18}$$

which, after applying the steady incompressibility relation, $\nabla \cdot \mathbf{u} = 0$ results in the Poisson equation for pressure:

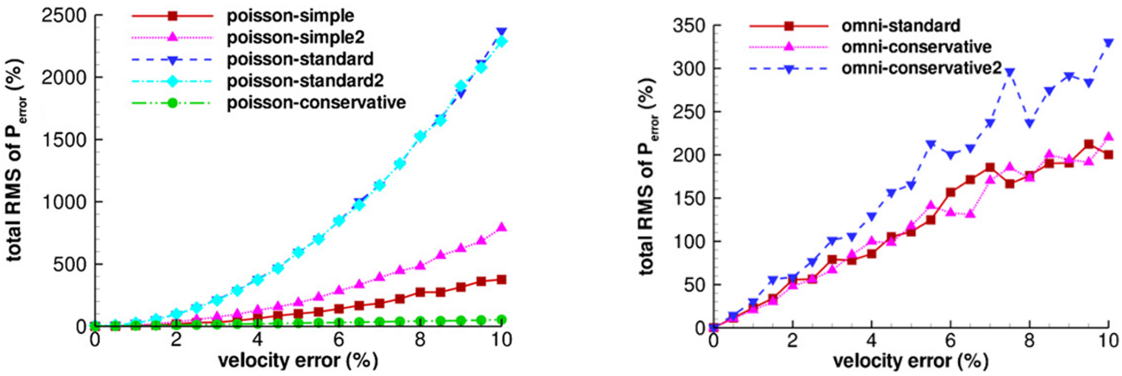
$$\nabla^2 p = -\rho \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}) = \nabla \cdot \mathbf{G} \tag{1.19}$$

where the vector field $\mathbf{G}$ represents the incompressible version of the conservative field $\mathbf{F}$. The second-order differential function represents a boundary value problem, which requires that a Dirichlet boundary condition is defined at least on one of the domain boundaries or internal region to provide a unique solution, while the rest of boundary conditions may be set to be of the Neumann type.

In this case, the resolution of the second-order differential equation given by the Laplacian of the pressure field and the forcing term may be approximated using a number of methods. For instance, via a second-order finite difference scheme, using local basis functions or through spectral methods [61].

Advantages and limitations

Despite the two methodologies being extensively used and compared back-to-back in a number of research articles, there is no data demonstrating that any of them presents a consistently better performance. For instance, comparison of the NS-based and the Poisson-based methods in [52] found that the latter is less sensitive to noise propagation from velocity to pressure for the flow around a Savonius rotor. In contrast, the results reported in [16] for a decaying Taylor vortex flow provide a different view, where most of the resolution methods used for both formulations display a significant error propagation from velocity measurements to pressure estimations.



(a) Poisson solver.

(b) Navier-Stokes based method.

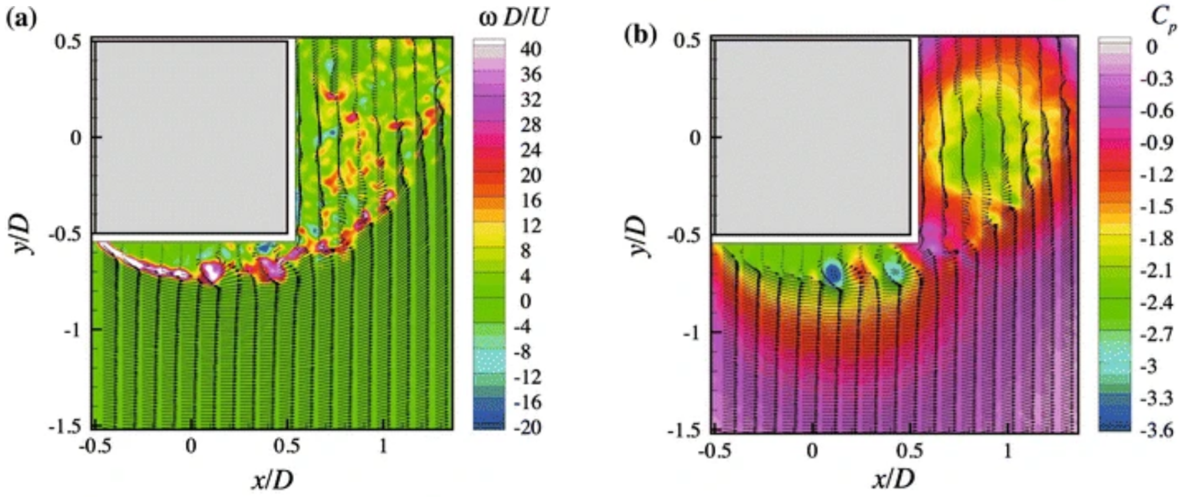
Figure 1.9: Relative reconstructed pressure RMS as a function of measured velocity error - from [16].

The results of the latter paper are included in Figure 1.9, where the total relative RMS of the reconstructed pressure is graphed as a function of the velocity measurement error. It is noted that the velocity error was introduced artificially by the authors following a Gaussian distribution with standard deviation equal to the error amplitude. As may be witnessed, different formulations of the Poisson boundary-value problem provide disparate responses to noise, with the conservative one delivering the most consistent results. On the other hand, the Navier-Stokes based results all give similar results, displaying a quasi-linear response between both error magnitudes.

Although some research argues that a standard Gaussian smoothing operation could suffice to address the issue of noise propagation, a number of solutions have been proposed, such as Proper Orthogonal Decomposition-based smoothing [10] or low-pass filtering.

A different challenge that derives from the formulation of both approaches is the divergence caused by outliers. As can be deduced, an outlier data point can result in a divergent gradient computation that propagates for path-integration-based algorithms. As a result, a careful data curation process is required for conventional pressure reconstruction, possibly followed by a reconstruction step, via interpolation or NS-based algorithms [65].

Finally, with typical engineering PIV set-ups, conventional pressure reconstruction algorithms make it difficult to reconstruct pressure up to the surface. This is because, in most engineering applications, the focus tends to be placed on capturing the most dominant integral scales, at the cost of spatial resolution. Additionally, near-the wall regions are often affected by light reflections, even if light-absorbing paints are used.



(a) Vorticity.

(b) Pressure coefficient.

Figure 1.10: Results from PIV around a square prism - from [32].

An example of this is visualized in Figures 1.10a and 1.10b, where the vorticity and reconstructed C_p fields around a square prism are plotted. As can be observed, while off-body structures around the wake and shear layer are properly captured, PIV data immediately around the solid boundary has been cropped, making surface pressure inference through traditional methods unfeasible unless a prior reconstruction step is performed. A more extreme case of this corresponds to the gappy-PIV in Figure 1.8, where missing data would make pressure reconstruction impractical.

From the above information, it can be concluded that, while traditional methods to infer pressure from PIV have the advantage of extensive research and already being implemented in commercial programs, they present a number of limitations. Some of the most relevant ones are noise and outlier sensitivity or the dependence to boundary conditions and choice of the integration paths.

1.4. Deep Learning in Fluid Mechanics

More recently, machine learning models - statistical tools optimized to learn patterns from large datasets [11], have been increasingly used in Fluid Mechanics applications for various reasons. Among them, increased data access, more sophisticated algorithms, advances in parallel computing or the rapid rise of AI-related research are highlighted in [12].

Within Machine Learning models, Deep Learning algorithms, which rely on neural network frameworks to learn the underlying patterns in training data, have proliferated due to the rich variety of model architectures and their efficiency in learning to map input to output vectors via (un)supervised training [26].

Although all neural networks (*NNs*) are based on similar principles, where neurons serve as nodes that are interconnected through linear operators of the form $output = weight \times input + bias$, it is precisely the layout of connections that provides the variability in network types.

While detailing the nature of each NN type is not the goal of this thesis, it is worth noting examples of their use cases in Fluid Mechanics for the reader's reference. For instance, the architecture of Long Short Term Memory (*LSTM*) networks has been found to be especially powerful in the reconstruction of temporal flow variations. An instance of this is reported in [20], where an LSTM architecture is used to yield the temporal POD coefficients from non-time-resolved PIV and temporal point velocity data, allowing to generate time-resolved PIV samples at a frequency of up to 2 kHz.

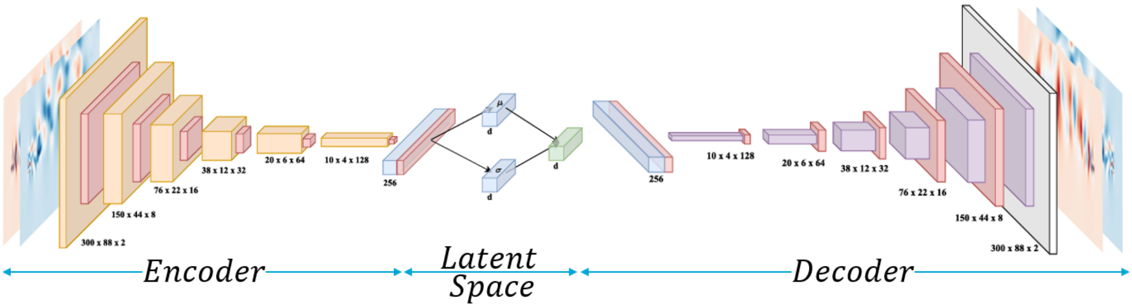


Figure 1.11: Transformer architecture - adapted from [70].

Further results on the use of NNs in modal-based flow decomposition are reported in [70], where the β-variational autoencoder architecture in Fig. 1.11 is used to find an efficient representation of the flow in a low-dimensional and near-orthogonal latent space. Comparison to conventional *POD* on a periodic and chaotic flow suggests that the proposed architecture might be superior to conventional techniques. This is because, when trained on the chaotic flow case, it is able to reconstruct an additional $+25\%$ energy content with the first 20 modes, making data representation more efficient.

In the above image, the block diagram of the convolutional network is displayed, including the encoder and decoder architectures, as well as the mean and standard deviation of the distributions that characterize the latent space of variational autoencoders.

Additionally, feedforward Deep NNs have also proven to be useful in a number of Fluid Mechanics applications. In [42], researchers were able to embed Galilean invariance in a DNN framework aimed at predicting Reynolds Stress anisotropy in *RANS* simulations.

A different application is reported in [40], where a DNN architecture is employed to provide the optimal wall blowing and suction levels to reduce skin-friction drag. The architecture, shown in Figure 1.12, is fed spanwise data of the shear-stresses and reportedly achieves drag savings of up to 20% in the simulation of a Turbulent Channel Flow case. As observed, the proposed architecture contains a single hidden layer to predict the optimal wall blowing ratio for a given shear stress distribution on the surface.

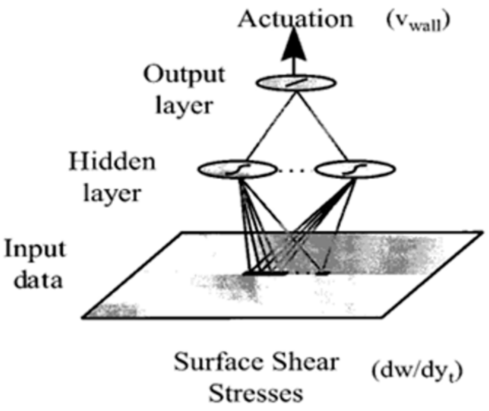


Figure 1.12: Feedforward network to control wall blowing ratio - from [40].

1.4.1. Physics-Informed Neural Networks

More recently, particular attention has been brought to the concept of Physics-Informed Neural Networks (*PINNs*), firstly introduced by Raissi et al. [58] in 2019. The core idea behind this family of Deep Learning models is to embed the physical laws that describe the problem by including a loss term based on the residuals of its governing equations. By doing so, the set of allowed solutions are ensured to have physical meaning, potentially adding a layer of regularization [51].

NN Architecture

In Figure 1.13, an example of *PINN* framework from [31] is shown, specifically for an unsteady, three-dimensional flow case. In this representation, the components of the NN architecture have been highlighted in green.

In the first place, the input layer takes the spatial and temporal coordinates of the domain. This information is propagated through the fully-connected, *hidden* layers, where information undergoes the aforementioned linear transformation, i.e. $output = weight \times input + bias$ from neuron to neuron. Here, the weights and biases represent the trainable parameters in the network. Finally, an output layer is present that contains the prediction for all flow variables of interest at each of the input coordinates.

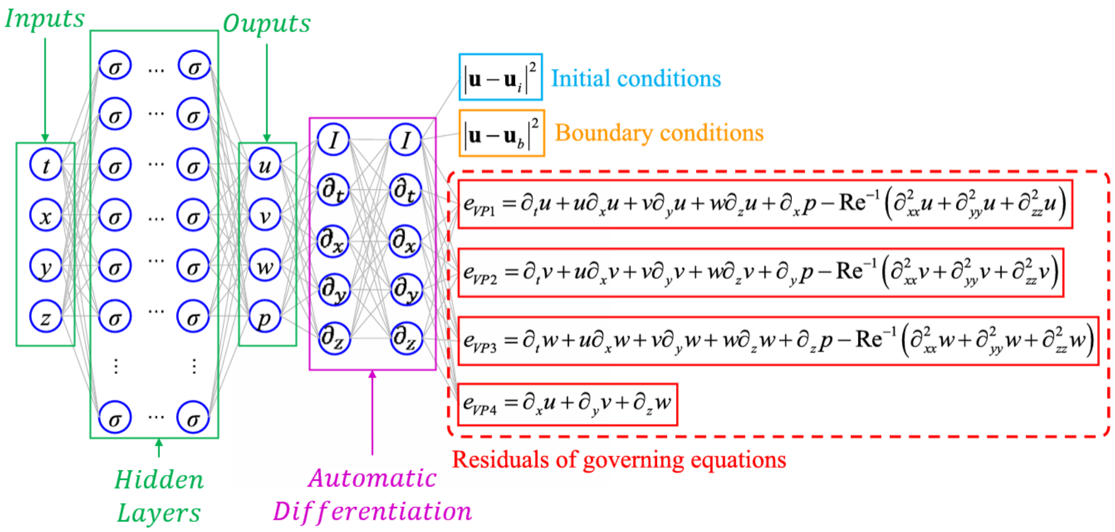

Inputs
Ouputs
t
x
y
z
σ
u
v
w
p
I
∂_t
∂_x
∂_y
∂_z
$|\mathbf{u}-\mathbf{u}_i|^2$ Initial conditions
$|\mathbf{u}-\mathbf{u}_b|^2$ Boundary conditions
$e_{VP1} = \partial_t u + u\partial_x u + v\partial_y u + w\partial_z u + \partial_x p - \mathrm{Re}^{-1}(\partial^2_{xx}u + \partial^2_{yy}u + \partial^2_{zz}u)$
$e_{VP2} = \partial_t v + u\partial_x v + v\partial_y v + w\partial_z v + \partial_y p - \mathrm{Re}^{-1}(\partial^2_{xx}v + \partial^2_{yy}v + \partial^2_{zz}v)$
$e_{VP3} = \partial_t w + u\partial_x w + v\partial_y w + w\partial_z w + \partial_z p - \mathrm{Re}^{-1}(\partial^2_{xx}w + \partial^2_{yy}w + \partial^2_{zz}w)$
$e_{VP4} = \partial_x u + \partial_y v + \partial_z w$
Residuals of governing equations
Hidden Layers
Automatic Differentiation

Figure 1.13: Example of PINN framework for unsteady 3D3C flow - adapted from [31].

As can be observed, the entire framework is more complex, since the loss function to tune the trainable parameters contains various components. In the above case, for an unsteady flow, initial conditions are enforced softly in blue. Additionally, all known data points in the space-time domain are learnt through the boundary condition loss term, in yellow. These two terms form the contribution of supervised learning to the total loss, where NN predictions are compared with labeled data.

However, the term that characterizes *PINNs* is the Partial Differential Equation loss, in red, which requires that the partial derivatives of the output variables are computed locally with respect to the input coordinates. In this framework, this is achieved in an efficient manner via Automatic Differentiation.

Automatic Differentiation

One of the key advantages *PINNs* offer over numerical-differentiation-based pressure reconstruction methods is their ability to compute partial derivatives locally and up to machine precision, without relying on finite-difference schemes. This is made possible thanks to automatic differentiation, a technique that evaluates the derivatives of composite functions efficiently using the chain rule [9], and is inherently supported by the deep learning framework of *PINNs* [58].

In the case of *PINNs*, where the derivatives of outputs with respect to inputs are required, automatic differentiation in reverse mode is used. To better understand this concept, a reduced version of a *PINN* system with two hidden layers for x-velocity is presented in Figure 1.14.

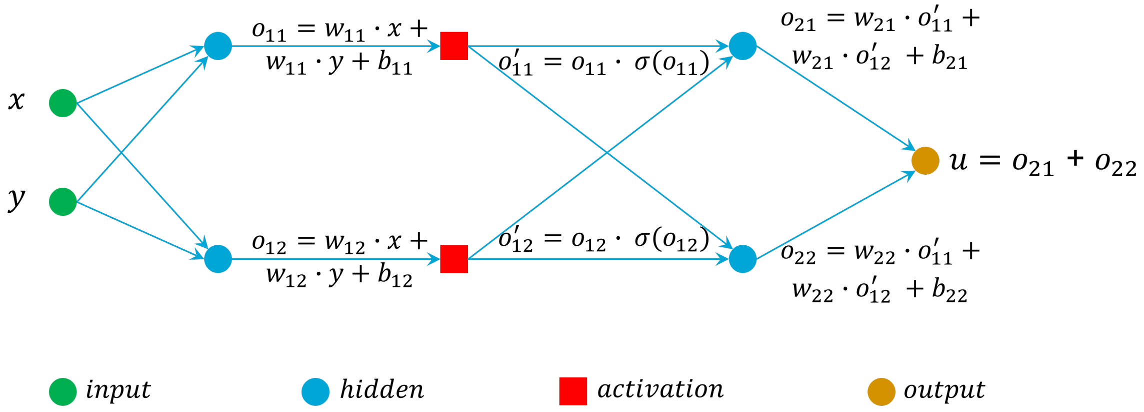

x
y
$o_{11} = w_{11} \cdot x + w_{11} \cdot y + b_{11}$
$o_{12} = w_{12} \cdot x + w_{12} \cdot y + b_{12}$
$o'_{11} = o_{11} \cdot \sigma(o_{11})$
$o'_{12} = o_{12} \cdot \sigma(o_{12})$
$o_{21} = w_{21} \cdot o'_{11} + w_{21} \cdot o'_{12} + b_{21}$
$o_{22} = w_{22} \cdot o'_{11} + w_{22} \cdot o'_{12} + b_{22}$
$u = o_{21} + o_{22}$
● input
● hidden
🟥 activation
● output

Figure 1.14: Forward pass operations.

In the above diagram, from left to right there are: the input layer, a hidden layer with two neurons, the activation layer, a second hidden layer with two neurons and finally the single-output layer. Furthermore, the operations that relate the output of each layer to its inputs are shown for each connection. This diagram, with its operations, reflects the so-called *forward pass*, where information is transferred from inputs to outputs.

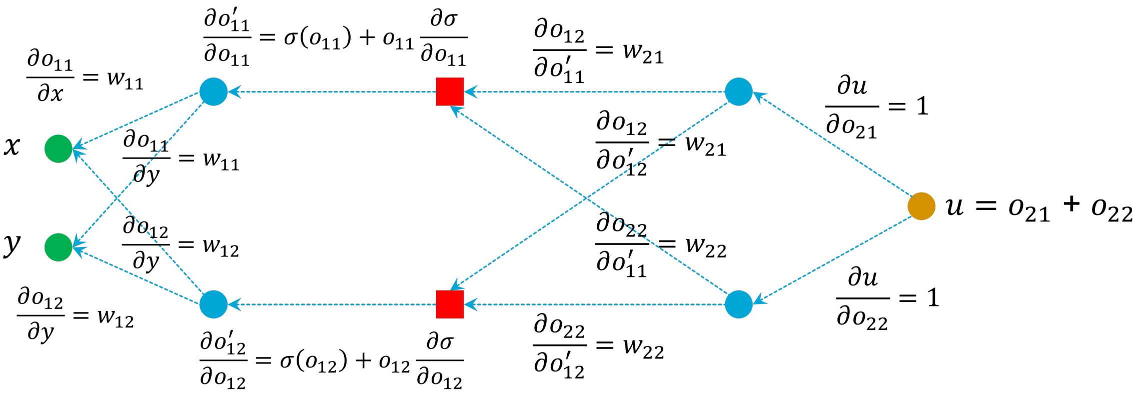

$\frac{\partial o_{11}}{\partial x} = w_{11}$
$\frac{\partial o'_{11}}{\partial o_{11}} = \sigma(o_{11}) + o_{11}\frac{\partial \sigma}{\partial o_{11}}$
$\frac{\partial o_{12}}{\partial o'_{11}} = w_{21}$
$\frac{\partial u}{\partial o_{21}} = 1$
x
$\frac{\partial o_{11}}{\partial y} = w_{11}$
$\frac{\partial o_{12}}{\partial o'_{12}} = w_{21}$
$u = o_{21} + o_{22}$
y
$\frac{\partial o_{12}}{\partial y} = w_{12}$
$\frac{\partial o_{22}}{\partial o'_{11}} = w_{22}$
$\frac{\partial u}{\partial o_{22}} = 1$
$\frac{\partial o_{12}}{\partial y} = w_{12}$
$\frac{\partial o'_{12}}{\partial o_{12}} = \sigma(o_{12}) + o_{12}\frac{\partial \sigma}{\partial o_{12}}$
$\frac{\partial o_{22}}{\partial o'_{12}} = w_{22}$

Figure 1.15: Derivatives of outputs with respect to inputs.

On the other hand, in Figure 1.15, the derivatives of the outputs with respect to the inputs of each node are shown for the same architecture. As may be observed, derivative expressions in neurons take very simple forms due to the linear transformation, whereas more involved expressions take place in the activation layer.

Thus, the expression to compute the derivative of the NN output with respect to one of the inputs, in this case x, would be as per Equation 1.20, where each term in the differentiation chain can be taken from the diagram above.

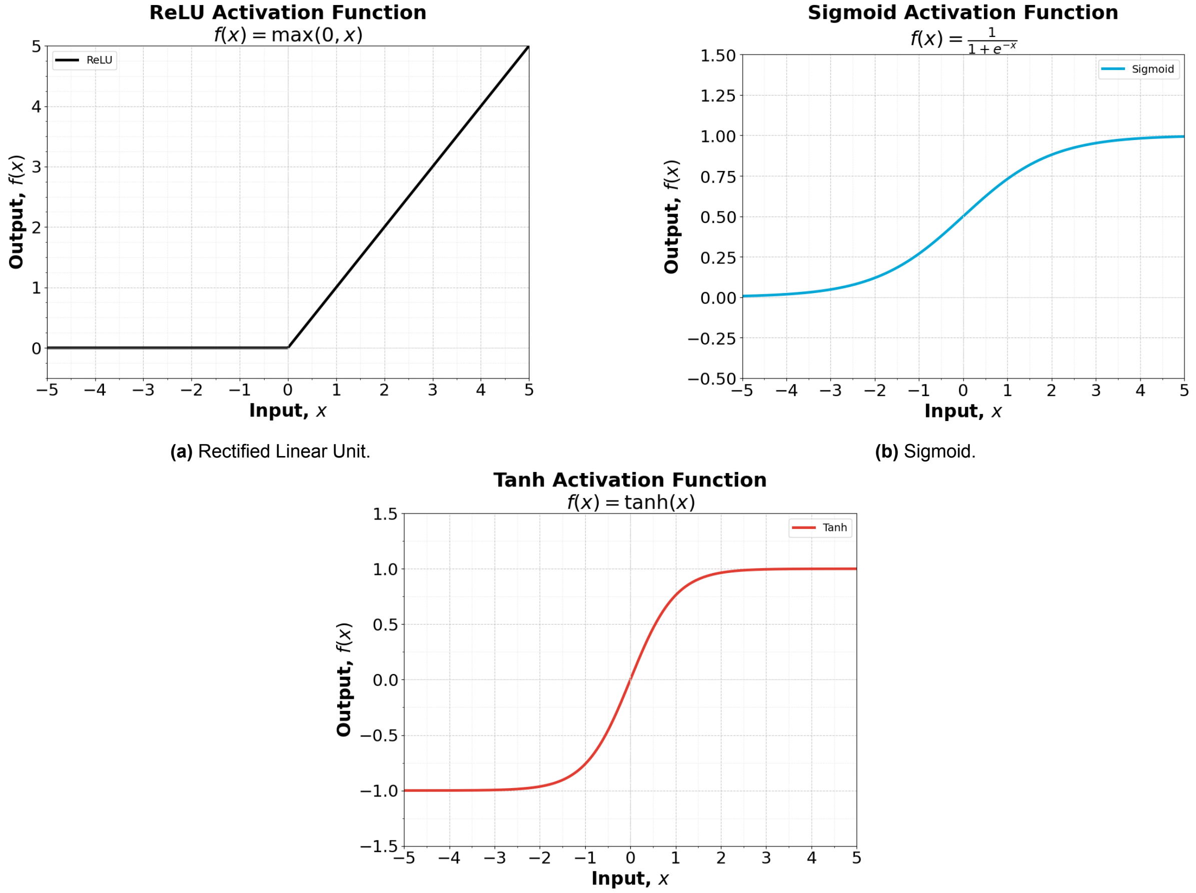
$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial o_{21}} \left(\frac{\partial o_{21}}{\partial o'_{11}} \frac{\partial o'_{11}}{\partial o_{11}} \frac{\partial o_{11}}{\partial x} + \frac{\partial o_{21}}{\partial o'_{12}} \frac{\partial o'_{12}}{\partial o_{12}} \frac{\partial o_{12}}{\partial x} \right) + \frac{\partial u}{\partial o_{22}} \left(\frac{\partial o_{22}}{\partial o'_{11}} \frac{\partial o'_{11}}{\partial o_{11}} \frac{\partial o_{11}}{\partial x} + \frac{\partial o_{22}}{\partial o'_{12}} \frac{\partial o'_{12}}{\partial o_{12}} \frac{\partial o_{12}}{\partial x} \right) \tag{1.20}$$

Even if the above expression can appear to be cumbersome, in practice this is implemented in a straightforward way using the *autograd* function in *PyTorch*, which creates a computational graph that allows one to efficiently compute the required first and second partial derivatives in tensor form. An example is given by the following command:

```
du_dx = torch.autograd.grad(u, x, torch.ones_like(x), create_graph=True)[0]
```

Activation functions

So far, the discussion on *PINNs* has considered internal NN operators as linear. However, focusing on turbulent-flow applications, where non-linearities are present, the so-called *activation* layers are required by the feedforward DNN to provide a sufficiently complex representation space that can contain the flow solution.



(a) Rectified Linear Unit.

(b) Sigmoid.

(c) Hyperbolic tangent.

Figure 1.16: Representations of common activation functions.

Activation layers include a non-linear input-to-output transformation of the form $output = f\left(input\right)$, where $f\left(\cdot\right)$ denotes a non-linear function. Although there are a number of possibilities for activation functions, such as *ReLU*, *sigmoid* or *tanh*, which are shown in Figures 1.16a, 1.16b, and 1.16c, respectively, not all activation functions are suitable for all problems.

After the above discussion on the use of automatic differentiation to compute local derivatives, it becomes apparent that the activation functions in *PINNs* require, at least, an order of differentiability equal to that of the largest partial derivative in the governing equations. In the specific case of turbulent flows, where the unsteady or Reynolds-Averaged Navier Stokes equations are used to compute the PDE loss term, the activation function has to be twice-differentiable, as required by the second-order derivative of the viscous term. This already discards activation functions such as the Rectified Linear Unit (*ReLU*) or its leaky version, which includes a non-zero slope for $\forall\, x < 0$.

Loss function

As exposed at the beginning of this section, the loss function is a conglomerate of various elements that depend on the specific problem. Nonetheless, these can generally be reduced to the two main loss contributions - the data and PDE losses.

- The *Data Loss* term is the error metric that represents the deviation in the reconstruction of all known variables in space-time. Accordingly, the process to compute it is rather straightforward, as it only requires a direct comparison between the labeled data and the prediction given by the *PINN*.

  For each $\{x,\, y,\, t\}$ set of inputs describing a point where labeled data is available, hereafter referenced as *measurement point*, the magnitude of the deviation between the *PINN* prediction and the ground truth is computed via the Squared Error across the literature, given by the expression in Eq. 1.21 for u.

  $$SE_u\left(x_i, y_i\right) \;=\; \left(\, u_{pred}\left(x_i, y_i\right) - u_{data}\left(x_i, y_i\right)\,\right)^2 \tag{1.21}$$

  Furthermore, each training iteration involves optimization based on a subsample of all available measurement points. The size of this subsample is defined as the *measurement batch size*, which is a random sampling of the entire pool of measurement points. Additionally, the number of iterations required to run through the entire set of measurement points conforms an epoch. Accordingly, in order to yield a single representative value over all points where data loss is computed, an unweighted average is performed, resulting in the Mean Squared Error. The expression of this metric for the x-velocity is shown in Eq. 1.22, where N_s reflects the measurement batch size.

  $$MSE_u = \frac{1}{N_s}\sum_{i=1}^{N_s} SE_u\;\left(x_i,\, y_i\right) \tag{1.22}$$

  Finally, the addition of the above expression for each of the labeled variables used for training, gives the total data loss. In Equation 1.23, this has been shown for the case discussed in Figure 1.13, where only the three velocity components are known.

  $$L_{data} = MSE_u + MSE_v + MSE_w \tag{1.23}$$

  Finally, it can be deduced that this format can accommodate a variety of loss types, including initial conditions, boundary conditions of the Dirichlet type, and others.

- Secondly, the Partial Differential Equation loss reflects the agreement of the flow solution given by the *PINN* to the governing equations, thus softly enforcing solutions with physical meaning.

In particular, the metrics used to enforce physics are the residuals of such governing equations. For instance, the example provided earlier in Fig. 1.13 uses the unsteady, incompressible Navier-Stokes equations to compute the PDE loss term. In accordance, the expressions for the momentum $(e_1 - e_3)$ and continuity (e_4) residuals take the forms in Equations 1.24 and 1.25 respectively.

$$e_i = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{1}{Re} \frac{\partial^2 u_i}{\partial x_j^2} \, ; \quad i = 1, 2, 3 \tag{1.24}$$

$$e_4 = \frac{\partial u_i}{\partial x_i} \, ; \quad i = 1, 2, 3 \tag{1.25}$$

The locations where residuals are computed are referred to as *collocation points*. Furthermore, given the flexibility offered by the Automatic Differentiation framework to compute derivative terms, the PDE loss can be evaluated at any point in space with its accuracy remaining unaffected by spatial resolution, since finite differencing is not required. Accordingly, a number of sampling strategies have been studied in literature to define their optimal distribution, such as residual-based adaptive refinement or *RAR* in [31].

This technique trains the *PINN* on an initial set of randomly placed collocation points, then evaluates the total loss at a much larger randomly sampled space and retains the locations where the PDE loss falls above a certain threshold to keep training. Nevertheless, while reporting error reductions of up to 50% for the Kovasznay flow, little attention is given to the collocation sample size sensitivity. This could mean that error reduction could be tied to the sample size rather than the sampling strategy, hence reflecting the challenge to efficiently define collocation points.

Similarly to the data loss, the Mean Squared Error is generally selected to compute a representative metric of the PDE loss across the batch of collocation points that is evaluated at each iteration. Consequently, the PDE loss takes the form of Equation 1.26. In the expression, the dummy index i reflects each collocation point, while j represents the number of equations that form the PDE loss.

$$L_{PDE} = \frac{1}{N_c} \sum_{i=1}^{N_c} \sum_{j=1}^{N_{eq}} e_j \left(x_i, \, y_i \right) \tag{1.26}$$

In general, the construction of any *PINN* framework requires at least these two loss terms, which form the most basic form of total loss. In the literature, a number of investigations have been conducted to determine whether the unequal weighting of both terms can lead to increased noise robustness or improved learning dynamics. A generic formulation of this is shown in Equation 1.27, where α and β represent the weights.

$$L_{tot} \, = \, \alpha \, L_{PDE} \, + \, \beta \, L_{data} \tag{1.27}$$

In this regard, static weighting of the N-S loss for the 2D decaying vortex problem with artificial Gaussian noise in [76] showed that biasing the loss towards the PDE term resulted in no additional regularization for noise levels up to 20%. On the other hand, in [31] the dynamic weighting of the data and PDE loss terms is proposed to balance their contribution during training. Results applied to the 2D Kovasznay flow using the velocity-vorticity and velocity-pressure formulations of the N-S equations, suggest that the dynamic formulation can bring a consistently balanced contribution of both loss terms during training, addressing the issue of problem-specific optimal weights.

Training Process

Once the various components of the *PINN* framework have been discussed, it is worth exposing how the trainable parameters are adjusted to provide an optimal flow solution. In order to achieve this, the loss function acts as a key performance indicator that must be optimized.

Before the optimization process is started, the NN architecture must be created and its parameters initialized, for which several methods exist such as He or Xavier [24]. After initialization and definition of the various NN hyperparameters, such as the learning rate or batch size, the iterative optimization process is conducted.

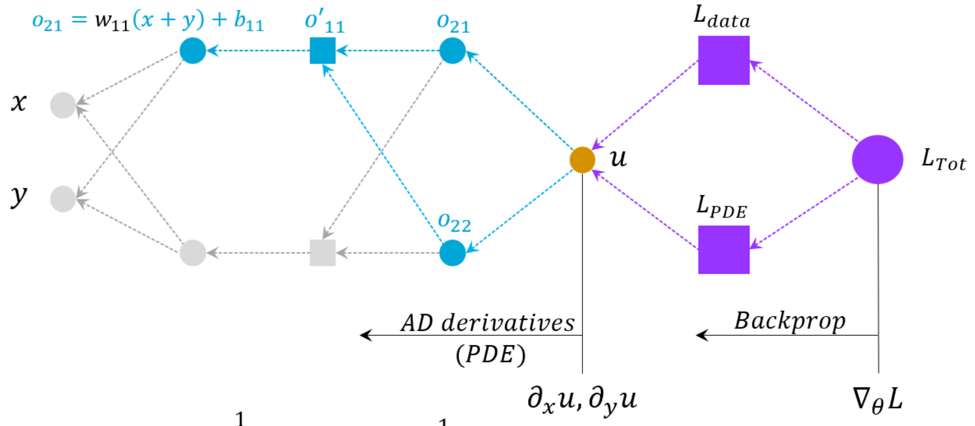
The first step in gradient-based methods is the evaluation of the *PINN* model in the batch of measurement and collocation points. This operation generates a set of output variables as a function of the inputs, allowing to compute an instance of the loss function. Since the weights and biases are the tunable parameters, these are to be adjusted such that the loss function is minimized.

For stochastic gradient-descent (SGD) this step, known as *backpropagation*, takes the form of Eq. 1.28, where θ represents the set of trainable parameters, n is the iteration step, η the learning rate and L is the loss function. Interpretation of this formula leads to the conclusion that the trainable parameters are modified in the direction specified by the negative gradient of the loss function with respect to the trainable parameters themselves, which should provide the direction of maximum loss descent. Furthermore, the magnitude of the update step is controlled by the learning rate, η.

$$\theta_{n+1} = \theta_n - \eta \cdot \nabla_\theta \, L\left(\theta; \mathbf{x}\right) \tag{1.28}$$

Nevertheless, given the complexity of the optimization space, various approaches exist to determine in which direction trainable parameters should be adjusted. These are known as optimizers, among which the most wide-spread are variants of SGD, such as momentum, RMS-prop or Adam [26]. The latter is the most widespread in most deep learning applications, as it uses exponentially-weighted moving averages to efficiently store first and second order statistical moments of gradients through iterations, which are used to set the optimization direction.

However, a crucial advantage of PINNs lies in the fact that the gradient computation required for back-propagation incurs only minimal additional computational effort. This is shown in Figure 1.17, which reflects back-propagation (magenta) and the computation of PDE derivatives (blue) in the simplified architecture used prior.



$$\frac{\partial L_{Tot}}{\partial w_{11}} = \left(\frac{\cancelto{1}{\partial L_{Tot}}}{\cancel{\partial L_{data}}}\frac{\partial L_{data}}{\partial u} + \frac{\cancelto{1}{\partial L_{Tot}}}{\cancel{\partial L_{PDE}}}\frac{\partial L_{PDE}}{\partial u}\right)\left(\frac{\partial u}{\partial o_{21}}\frac{\partial o_{21}}{\partial o'_{11}}\frac{\partial o'_{11}}{\partial w_{11}} + \frac{\partial u}{\partial o_{22}}\frac{\partial o_{22}}{\partial o'_{11}}\frac{\partial o'_{11}}{\partial w_{11}}\right)$$

Figure 1.17: Diagram depicting the reuse of PDE derivatives in backpropagation.

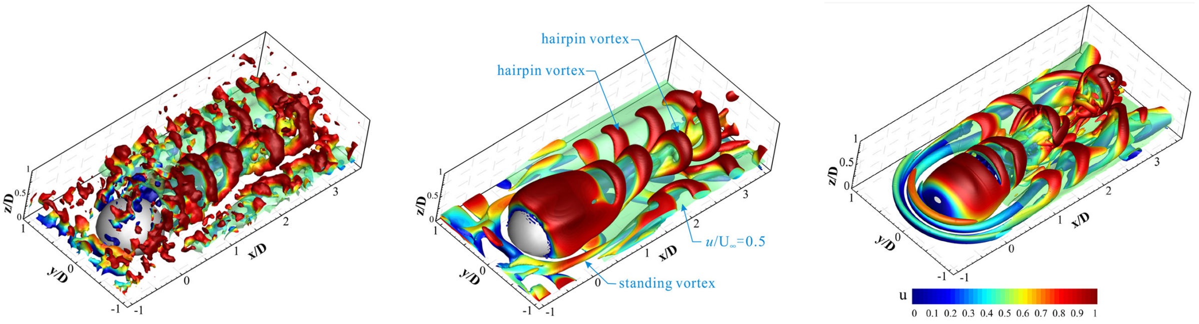
As is observed in the expression of the total loss gradient with respect to the trainable parameter w_{11}, the chain rule of differentiation allows to reuse a number of the derivatives obtained during the PDE loss computation, which is performed prior to backpropagation. In particular, all the terms in blue inside the expression are reused from the PDE loss calculation, making backpropagation, which requires higher computational cost than the forward pass [58], more efficient.

PINNs for Pressure Reconstruction

The above summary on how *PINNs* work already highlights some of its core advantages, such as the ability to compute derivatives without discretization error, the freedom to define collocation points or the flexibility it offers when defining the PDE loss. As a result, extensive use cases have been studied. These include forward and inverse problems, as well as unsteady [39] and time-averaged [27] problems for data assimilation [76] and direct simulation [31] of various engineering flows and even in medical use cases [59].

Focusing on the data assimilation branch and, in particular, on PIV applications, a number of use cases have been studied for *PINNs*. The researchers in [39] have reported their use to increase the temporal resolution of PIV. Their proposal is based on embedding an unsteady Navier-Stokes loss term that allows to reconstruct the flow field between CFD velocity snapshots at 15 Hz when used in conjunction with point pressure data. Additionally, comparison of the results with a high-frequency simulation of the flow around one, two and three cylinder arrangements resulted in relative errors as low as 4% in the most complex flow case, showcasing its ability to approximate unsteady flow dynamics.

Additionally, researchers in [76] have studied the performance of *PINNs* on real tomographic PIV data of the flow around a hemisphere. Their findings support the suitability of the framework to de-noise experimental data and to enhance spatial resolution when experimental measurements are rather sparse, of particular relevance in tomo-PIV.



(a) Tomo-PIV data.

(b) PINN output

(c) DNS simulation.

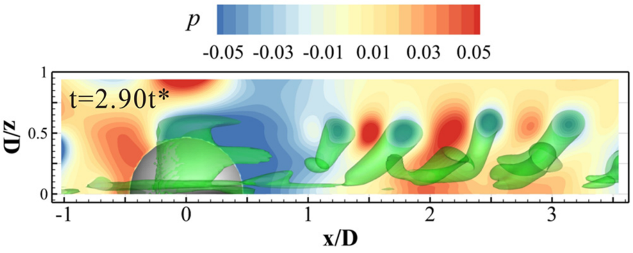
Figure 1.18: Isosurfaces of Q-criterion, $Q = 0.2$ colored with dimensionless velocity and of $u/U_\infty = 0.5$. [76]

An instance of this is represented in Figure 1.18, where Q-criterion isosurfaces with $Q = 0.2$ reflect the effect of the *PINN* algorithm on flow interpretability. This can be observed by direct comparison of Figures 1.18a and 1.18b, corresponding to the 3D PIV data before and after the *PINN* execution respectively. Visibly, the NN output data offers a clearer picture of the flow pattern, not only in the hairpin vorticity but also regarding the sheet vorticity along the shear layer. What's more, comparison of the structures with the high-fidelity simulation data in Figure 1.18c shows that certain structures such as the standing vortex can only be appreciated after running the DNN.

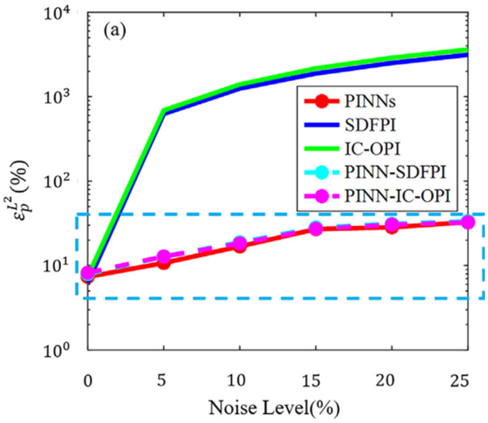
Even if the above study places no special emphasis or accuracy assessment on it, reference is made to the ability of the *PINN* framework to reconstruct the pressure field without additional data requirements. In particular, the pressure contours along the centerline of the wake shown in Figure 1.19a are included, of which the only comment is the correspondence between the low pressure regions at the core of the eddies.

This ability of *PINNs* to infer pressure derives from the fact that the PDE loss term used during the NN optimization includes the momentum conservation equations, of which the only unknown variable remain the spatial derivatives of the pressure field. Consequently, the pressure field can be reconstructed up to a constant as the spatial pressure gradient terms are adjusted to minimize the momentum residuals, while all the other terms including the velocity field are additionally constrained in a soft manner by the supervised learning component.

This idea was already presented in the initial paper by Raissi et al. [59] and has been studied extensively with artificial PIV data generated from CFD simulations [76] [31] [51] [14], highlighting the robustness of PINNs to artificial Gaussian noise and its performance when compared to more conventional pressure reconstruction methods. An instance of this is displayed in Figure 1.19b, where the pressure reconstruction error has been represented as a function of the artificial Gaussian noise level for 2D Taylor decaying vortices. The graph depicts a superior robustness of *PINNs* compared to conventional spectral (SDFPI) and direct-integration (IC-OPI) methods.



(a) Pressure field on the wake centerline of the flow around a hemisphere. [76]



(b) Noise propagation for 2D Taylor decaying vortices [22]

On the contrary, research on the ability of *PINNs* to reconstruct the pressure field from real experimental PIV data is scarce. However, certain examples exist which reflect the suitability of the method for data obtained experimentally. For example, the research group in [15], demonstrated that the unsteady pressure field along the lateral line of swimming fish can be reconstructed with superior accuracy than the *Queen 2.0* algorithm, introduced in [18], when compared to DNS simulation data. Similarly, results reported in [22] expose the performance of *PINNs* in the reconstruction of the pressure field from 2D-3C PIV measurements of a synthetic jet impinging on a wall. In this study, the assessment against conventional methods also reflects the reduced sensitivity of the proposed *PINN* architecture to measurement noise.

Overall, the research based on experimental data appears to be limited and has only been compared with high-fidelity simulation data or even just judged qualitatively, seldom focusing on surface pressure in particular. As a result, the need appears to directly compare *PINNs* with more established pressure measurement techniques such as pressure tapping and conventional PIV-based measurements.

1.5. Research proposal

1.5.1. Background synthesis

Synthesizing the above section, the relevance of surface pressure in fluid dynamic applications has been introduced first, highlighting its role as a force generation mechanism and its influence on the production and evolution of turbulence. Subsequently, the analytical approach, CFD simulations and experimental measurements have been introduced as the methods that allow to obtain surface and off-body pressure information. After placing emphasis on the advantages and limitations of all three techniques, the relevance of experimental measurements has become apparent due to the measurement of real flows and their flexibility when conducting parametric studies.

Keeping the interest on experimental simulations, the conventional surface pressure measurement techniques of pressure tapping and pressure-sensitive paint have been exposed. Even though these are used extensively, especially pressure tapping, they present a number of limitations. In the case of pressure taps, major drawbacks are intrusion and low spatial resolution, while the experimental set-up and variations in light intensity due to aero-elastic effects are particularly challenging in the case of pressure-sensitive paint.

As a consequence, alternative pressure reconstruction methods have emerged that work with Particle Image Velocimetry data, an experimental technique used to obtain instantaneous and simultaneous velocity fields, thus addressing some of the above issues. Such methods, which are based on the principle of momentum conservation, have been tested in extensive flow applications with success. However, they still present certain limitations, such as dependence on the integration path or sensitivity to experimental measurement noise.

Finally, the advent and proliferation of Machine Learning in Fluid Mechanics has been introduced, highlighting their versatility and the wide range of applications that have been found for the various types of architectures. A special emphasis has been placed on Physics-Informed Neural Networks, a specific group of Deep Neural Networks that blends supervised and unsupervised training, thanks to the addition of the residuals of governing equations as a loss term. Among the wide range of applications explored by researchers so far, the attention has been focused on *PINNs* as a novel technique to infer pressure from PIV data.

The *PINN* framework presents certain advantages with respect to conventional PIV-based pressure reconstruction methods, among which the avoidance of discretization schemes, the flexibility to define different PDE formulations or their robustness to noise stand out.

However, after revision of the literature on the use of *PINNs* for pressure reconstruction, a number of knowledge gaps have been observed by the author:

- The majority of papers focus on artificial PIV data [13] [31] [39] [51] [58] [59] (i.e. CFD data with added Gaussian noise), avoiding the challenges associated to experiments, such as reflections and correlated noise.
- Even if actual experimental data is used in a couple of papers, reference pressure is either not provided [22] [76] or obtained with high-fidelity simulations [15], thus obviating the layer of uncertainty introduced by the experimental setup.
- The challenges of reconstructing surface pressure from experiments, such as data gaps, are avoided across literature by placing the focus on off-body structures [76].

1.5.2. Research Questions

Consequently, the following primary and secondary research questions have been formulated in order to provide further insight on the applicability of *PINNs* as a novel technique for reconstructing pressure information from experimental PIV data.

Primary research question

- How does the performance of *PINNs* quantitatively and qualitatively compare to a conventional PIV-based pressure reconstruction technique like the Poisson Solver in the inference of surface pressure information when compared to pressure tap data?

Secondary research questions

- How do data gaps close to the solid boundary affect the accuracy of surface pressure reconstruction accuracy for *PINNs*?
- To what extent does the inclusion of physics-based boundary conditions on the surface (e.g. the no-slip condition) affect the accuracy of surface pressure reconstruction?
- How much does the provision of reference surface pressure data from pressure taps positively affect the pressure reconstruction accuracy of *PINNs*?
- To which extent is the performance of *PINNs* on CFD simulation data representative of its ability to reconstruct pressure from experimental data?

2 Problem definition

After the definition and justification of the research objectives of this thesis, the present chapter focuses on laying out the methodology and specific tests conducted to answer them. Furthermore, a detailed insight is given on the characteristics of the *PINN* framework used in this thesis due to its central role.

2.1. Tools and Methods

With the aim of answering the main and secondary research questions, the two-dimensional and time-averaged flow around a cylinder will be used as a test case. In addition, the project is divided into two major branches - while the first one is focused on the assessment of *PINNs* performance using data from CFD simulations, the second and most novel part uses experimental data to run the algorithm.

In order to help visualize the processes and tools used in the project, the flow charts in Figures 2.1 and 2.2 are referenced. Furthermore, these charts include the following color codes:

- gray boxes and black arrows for CFD data and processes.
- green boxes and arrows for experimental data and processes.
- orange boxes for *PINN* items with red arrows to indicate the flow of information during training.
- blue boxes to include the tools and software used to obtain or process the data.

2.1.1. Data acquisition and processing

Using the chart in Figure 2.1 as support and starting from the top-left, the simulation data of the unsteady flow around a cylinder is produced using the *STARCCM+* software package. Even though the simulation is three-dimensional, the output of interest are the unsteady flow variables on the symmetry plane, among which $u,\ v,\ p$ are extracted. Since the *PINN* framework used here is based on the flow averaged over time rather than on instantaneous snapshots, the time-average is computed using the same software via Equation 2.1, where N denotes the number of snapshots.

$$\overline{u} = \frac{1}{N}\sum_{i=1}^{N} u_i \tag{2.1}$$

The above yields the mean flow variables $\overline{u},\ \overline{v},\ \overline{p}$, which appear in the two-dimensional steady continuity (Eq. 2.2) and momentum conservation (Eq. 1.7) equations. Furthermore, it enables one to perform Reynolds decomposition and compute the time-averaged Reynolds Stress components, thus completing all the required flow variables.

$$\frac{\partial \overline{u}}{\partial x} + \frac{\partial \overline{v}}{\partial y} = 0 \tag{2.2}$$

As depicted in the diagram in Figure 2.1, the process to acquire experimental data of the flow around a cylinder is similar but contains some key differences. Firstly, from the top right, experimental tests are conducted in the M-tunnel of the Low-Speed Lab Facility in the TU Delft using a PIV acquisition system, as well as pressure taps across the top semi-circumference of the cylinder, which provide the time averaged pressure values at each angular station.

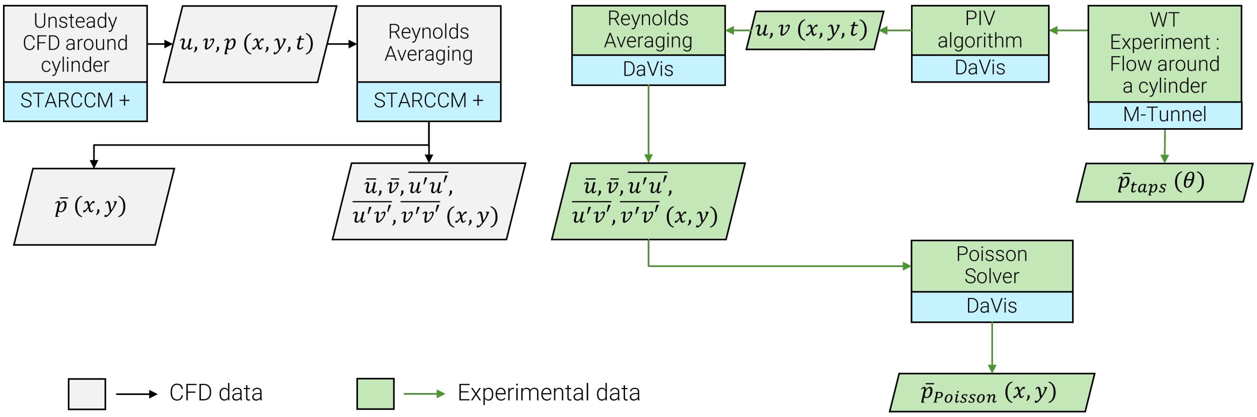

Unsteady CFD around cylinder
STARCCM +
$u, v, p\ (x, y, t)$
Reynolds Averaging
STARCCM +
$\bar{p}\ (x, y)$
$\bar{u}, \bar{v}, \overline{u'u'},$
$\overline{u'v'}, \overline{v'v'}\ (x, y)$
Reynolds Averaging
DaVis
$u, v\ (x, y, t)$
PIV algorithm
DaVis
WT Experiment : Flow around a cylinder
M-Tunnel
$\bar{u}, \bar{v}, \overline{u'u'},$
$\overline{u'v'}, \overline{v'v'}\ (x, y)$
$\bar{p}_{taps}\ (\theta)$
Poisson Solver
DaVis
$\bar{p}_{Poisson}\ (x, y)$
CFD data
Experimental data

Figure 2.1: Flowchart depicting the methods and tools used to obtain data.

Subsequently, the flow images are processed to obtain the two-dimensional velocity field after running a PIV algorithm in *DaVis*. Later, statistical analysis is performed on the instantaneous samples, analogously to CFD, to provide the mean velocities and Reynolds Stress components.

Another key difference from CFD simulations is the source of reference pressure values, as this is provided directly by the simulation in the case of CFD. On the contrary, two sources are used in the experimental branch. On the one hand, the aforementioned pressure tap data on the cylinder surface, and, on the other hand, the conventional pressure reconstruction technique from PIV, i.e. the Poisson solver. It is noted that the Poisson solver built in *DaVis* is used and run directly on the time-averaged data obtained from PIV.

2.1.2. PINN training and results assessment

Once the flow data is processed, the variable fields on the plane across the centerline of the cylinder are split into known variables, namely $\overline{u}$, $\overline{v}$, $\overline{u'u'}$, $\overline{u'v'}$, $\overline{v'v'}$, and the unknown variable, $\overline{p}$.

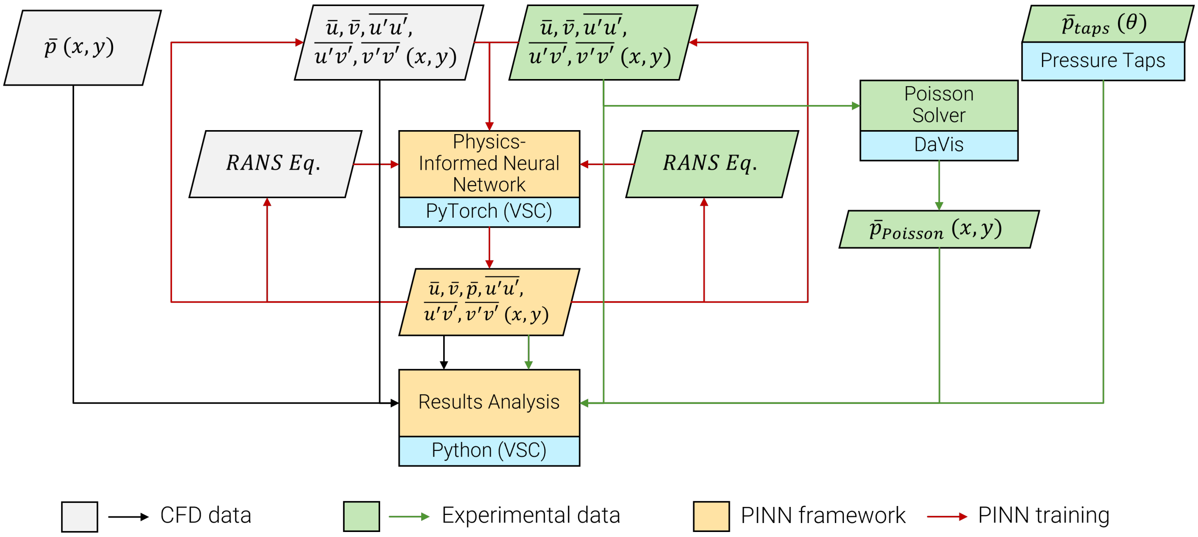

$\bar{p}\ (x, y)$
$\bar{u}, \bar{v}, \overline{u'u'},$
$\overline{u'v'}, \overline{v'v'}\ (x, y)$
$\bar{u}, \bar{v}, \overline{u'u'},$
$\overline{u'v'}, \overline{v'v'}\ (x, y)$
$\bar{p}_{taps}\ (\theta)$
Pressure Taps
Poisson Solver
DaVis
$RANS\ Eq.$
Physics-Informed Neural Network
PyTorch (VSC)
$RANS\ Eq.$
$\bar{p}_{Poisson}\ (x, y)$
$\bar{u}, \bar{v}, \bar{p}, \overline{u'u'},$
$\overline{u'v'}, \overline{v'v'}\ (x, y)$
Results Analysis
Python (VSC)
CFD data
Experimental data
PINN framework
PINN training

Figure 2.2: Flowchart depicting the methods and tools used to train and analyze the performance of PINNs.

As is reflected in the diagram in Figure 2.2, the known variables are used during the *PINN* training process, since comparing them to the *PINN*-predicted fields yields the supervised learning contribution of the loss. On the contrary, pressure data is only used to compute the validation loss. Additionally, all *PINN*-output variables are used to compute the residuals of the governing equations, in this case the time-averaged and two-dimensional continuity and momentum conservation equations.

It is noted that Python is used as the programming language to develop, train and analyze the results given by *PINNs*, making extensive use of the open-source PyTorch library. Besides, Visual Studio Code is selected as the programming environment.

Once an overview of the processes and tools used in this thesis has been given, a more detailed look is provided next on the characteristics of the *PINNs* used in the present thesis, with a prior exposition of the computational resources used to carry out the above activities.

2.1.3. Computational resources

With the exception of the PIV acquisition and algorithm, the rest of processes exposed above are carried out with an *HP-Omen 16* laptop, featuring the following specs:

- **CPU:** 12th Generation Intel(R) Core i7-12700H @2.30 GHz.
- **GPU:** NVIDIA GeForce RTX 3060.
- **RAM:** 16.0 GB dedicated memory.
- **OS:** Windows 11 Home.
- **Framework:** Python 3.9.19, PyTorch 2.4.1, CUDA 12.1.

2.2. Physics Informed Neural Network framework

The Physics-Informed Neural Network framework used in this thesis is focused on learning the values of fluid variables at each location of a two-dimensional plane describing the mean flow around a cylinder. To achieve this, the adjustable parameters in the network are optimized such that the loss function is minimized during the training phase.

In the following sections of the chapter, the *PINN* framework is exposed and justified in detail, covering its architecture, the different terms that form the loss function and the training parameters that have been selected.

2.2.1. NN architecture

Architecture

First of all, the neural network used is of the *deep* type, specifically a feed-forward deep neural network. In practice, this means it has an input layer, with N_{in} neurons, a number (L) of intermediate or *hidden* layers, with N_{hidden} neurons each, and an output layer, with N_{out} neurons.

Since all layers are fully-connected, this implies that the data passing through each node in a given layer, i, is propagated to all the nodes in the following layer. The result are $N_i \times N_{i+1}$ connections plus a bias term for each neuron in the second layer. With this in mind, one can compute the total number of trainable parameters by means of the expression in Eq. 2.3, where the above notation has been used.

$$n_{params} = N_{in} \times N_{hidden} + N_{hidden} + L \times (N_{hidden} \times N_{hidden} + N_{hidden}) + N_{out} \times N_{hidden} + N_{out} \quad (2.3)$$

In the present case study, where the focus is placed on the time-averaged and two-dimensional flow field around a cylinder, the number of neurons in the input layer is just two, $N_{in} = 2$, namely the x and y coordinates. On the other hand, since the flow variables of interest are the time averaged velocity components, pressure and all three Reynolds Stress components for two dimensions, the output layer has six output neurons, i.e. $N_{out} = 6$.

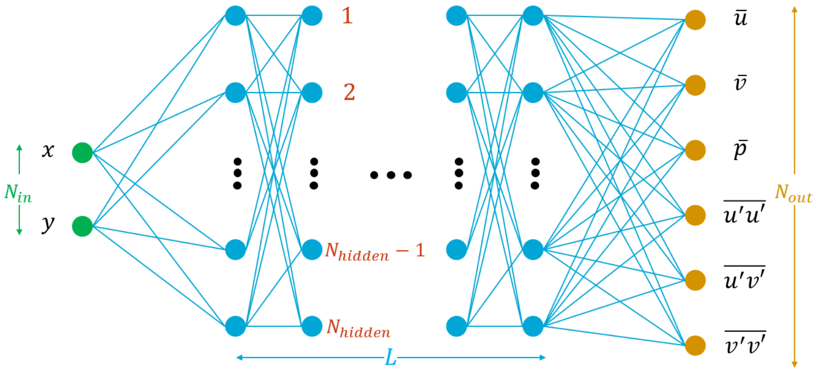


Figure 2.3: Architecture of the neural network used.

A visual description of the above is provided in Figure 2.3, where neurons are represented by circles and each connection by a line.

Variables and Coordinate System

It must be noted that, based on the observations in [69], all variables are made dimensionless, as this can foster a reduction in training time and prediction error. In order to achieve non-dimensionality, a number of approaches can be followed, such as normalization in the range $[-1, 1]$, standardization using μ and σ or any arbitrary definition. However, since the PDE loss includes all variables in the same physical dimension, an additional de-normalization step would be required prior to its computation.

If, instead, one makes use of characteristic magnitudes of the problem in the normalization process, the dimensionless version of the Navier-Stokes equations (Eq. 2.4) can be employed in the PDE loss term, thus saving a de-normalization step.

$$\overline{\tilde{u}_j}\frac{\partial \overline{\tilde{u}_i}}{\partial \tilde{x}_j} + \frac{\partial \widetilde{\overline{u'_i\, u'_j}}}{\partial \tilde{x}_j} = -\frac{\partial \overline{\tilde{p}}}{\partial \tilde{x}_i} + \frac{1}{Re}\left(\frac{\partial^2 \overline{\tilde{u}_i}}{\partial {\tilde{x}_i}^2}\right) \tag{2.4}$$

In the above equation, the tilde $(\tilde{\cdot})$ denotes the dimensionless version of each variable. For spatial coordinates, the cylinder diameter, D, is used as the reference length, namely:

$$\tilde{x}_i = \frac{x_i}{D} \tag{2.5}$$

whereas for each velocity and Reynolds Stress components, U_∞ serves as a reference magnitude:

$$\tilde{u}_i = \frac{u}{U_\infty}, \qquad \widetilde{u'_i\, u'_j} = \frac{u'_i u'_j}{U_\infty^2}. \tag{2.6}$$

Finally, static pressure has been made dimensionless with twice the dynamic reference pressure:

$$\tilde{p} = \frac{p}{\rho\, U_\infty^2} = \frac{p}{2\, q_\infty} \tag{2.7}$$

Additionally, along the thesis, both the Cartesian and Cylindrical reference frames are used, adjusting the choice to aid in the interpretation of results. As depicted in Figure 2.4, both coordinate systems have their origins at the cylinder center. On the one hand, the dimensionless $\{\, x,\, y \,\}$ coordinate axes take positive values downstream and above the cylinder, respectively. Besides, two variants of cylindrical coordinates $\{\, r,\, \theta \,\}$ are employed. Firstly, all calculations are conducted with the standard convention in black, where θ finds it origin to the right horizontal of the cylinder and increases counter-clockwise.

Moreover, in cyan, the plotting convention is reflected, where the origin of the angular coordinate coincides with the location of the stagnation point for the non-lifting cylinder. It is direct to see that the two variables are related through the expression in Equation 2.8.

$$\theta_{plot} = \pi - \theta \tag{2.8}$$

Note that the radial coordinate is made dimensionless with the cylinder diameter, as per Equation 2.9.

$$\tilde{r} = \frac{r}{D} \tag{2.9}$$

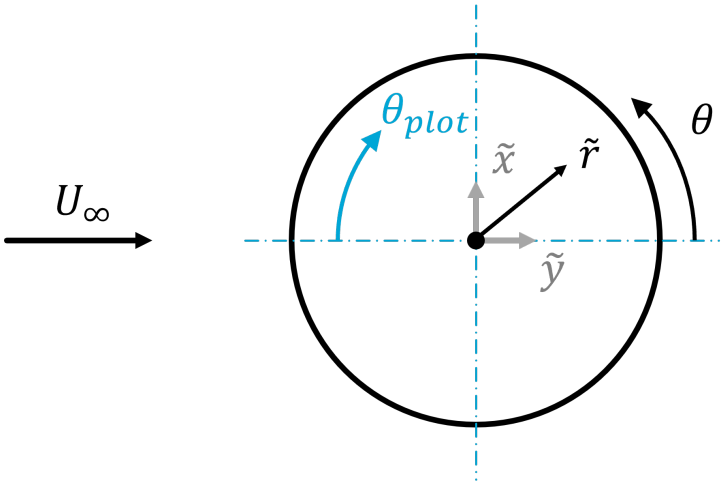


Figure 2.4: Cylindrical and Cartesian reference frame conventions used in the thesis.

Activation Function

After definition of the overall network architecture, an appropriate choice of non-linearity is required to affect the output at each neuron in the hidden layers. Following the observations in §1.4.1, the second-order differentiability requirement set by the viscous term in the momentum equations allows to disregard non-linearities with discontinuities, such as *ReLU* or *leaky ReLU*.

Additionally, research in [25] reports the difficulties the *tanh* and *sigmoid* activation functions present for deep architectures to converge to global optimal minima, as well as their poor learning dynamics during training due to the problem of vanishing gradients. This phenomenon appears specifically in deep neural networks or those with large recurrent dependencies and consists of the inability to back-propagate information during training due to the cumulative product of near-zero gradients, causing certain neurons to be effectively deactivated.

Although this phenomenon can be partially offset with proper weight initialization [25], alternative activation functions such as the Swish or Adaptive Swish functions, introduced in [60], allow to circumvent it. Additionally, results in [76] for *PINNs* on the two-dimensional Taylor decaying vortex problem show the superior performance of these two functions in training and validation loss, especially when compared to the other mentioned possibilities.

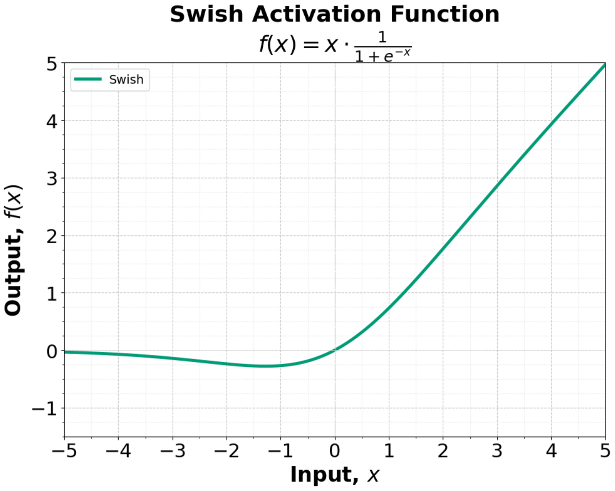


Figure 2.5: Swish activation function..

When comparing the Swish and Adaptive Swish functions, the latter demonstrated marginally better performance in the above study. However, it introduces an additional layer of complexity due to the extra trainable parameter β. Therefore, in this study, the Swish activation function — illustrated in Figure 2.5 — is chosen for its balance between performance and simplicity. Note that this non-linearity is applied after every neuron in all hidden layers, except for the output layer, where no activation functions are used to prevent imposing undesired boundedness on the predicted variables.

2.2.2. Loss function components

Once the input/output data and all the architecture components have been exposed, it is of interest to introduce the different loss terms that intervene in the NN training process. In this thesis, these are:

- data loss term, accounting for the flow variables with labeled data.
- PDE loss term, including the laws that describe the physical flow behavior.
- reference pressure loss term, to achieve the unique pressure field solution.
- Reynolds Stress component loss, to ensure physically sensible values for the components normal Reynolds Stresses.
- boundary condition loss term, to add known values (Dirichlet) or spatial derivatives (Neumann), at any point in space.

Next, the formulation of each loss term will be included along with a description and justification.

Data Loss

As exposed in §1.4.1, the *Data Loss* term is the error metric that represents the deviation in the reconstruction of all known variables in space. For this purpose, the Squared Error averaged over the batch of measurement points at each iteration is used as an indicator.

In the present thesis, where the known data are the time-averaged velocity and Reynolds Stress fields in two dimensions, the data loss takes the expression in Equation 2.10:

$$\mathcal{L}_{data} = MSE_{\overline{\tilde{u}}} + MSE_{\overline{\tilde{v}}} + MSE_{\overline{\widetilde{u'u'}}} + MSE_{\overline{\widetilde{u'v'}}} + MSE_{\overline{\widetilde{v'v'}}} \tag{2.10}$$

From the above, it is deduced that the loss at each measurement point and for every variable is given the same relevance in the present study. However, it becomes apparent that a different strategy could be implemented, such as weighting data points based on measurement uncertainty or biasing the loss function towards first-order statistical moments, i.e. $\overline{\tilde{u}}$, $\overline{\tilde{v}}$.

Partial Differential Equation Loss

Secondly, the Partial Differential Equation loss reflects the agreement of the flow solution given by the *PINN* to the governing equations, thus softly enforcing solutions with physical meaning.

In this case, for the time-averaged, two-dimensional and turbulent flow around a cylinder, the Reynolds-Averaged Navier Stokes equations are used. Consequently, the expressions for all three residuals, namely the conservations of mass, x and y-momentum, are given by Equations 2.11, 2.12 and 2.13 respectively:

$$e_1 = \frac{\partial \overline{\tilde{u}}}{\partial \tilde{x}} + \frac{\partial \overline{\tilde{v}}}{\partial \tilde{y}} \tag{2.11}$$

$$e_2 = \overline{\tilde{u}}\,\frac{\partial \overline{\tilde{u}}}{\partial \tilde{x}} + \overline{\tilde{v}}\,\frac{\partial \overline{\tilde{u}}}{\partial \tilde{y}} + \frac{\partial \overline{\widetilde{u'\,u'}}}{\partial \tilde{x}} + \frac{\partial \overline{\widetilde{u'\,v'}}}{\partial \tilde{y}} = -\frac{\partial \overline{\tilde{p}}}{\partial \tilde{x}} + \frac{1}{Re}\left(\frac{\partial^2 \overline{\tilde{u}}}{\partial \tilde{x}^2} + \frac{\partial^2 \overline{\tilde{u}}}{\partial \tilde{y}^2}\right) \tag{2.12}$$

$$e_3 = \overline{\tilde{u}}\,\frac{\partial \overline{\tilde{v}}}{\partial \tilde{x}} + \overline{\tilde{v}}\,\frac{\partial \overline{\tilde{v}}}{\partial \tilde{y}} + \frac{\partial \overline{\widetilde{u'\,v'}}}{\partial \tilde{x}} + \frac{\partial \overline{\widetilde{v'\,v'}}}{\partial \tilde{y}} = -\frac{\partial \overline{\tilde{p}}}{\partial \tilde{y}} + \frac{1}{Re}\left(\frac{\partial^2 \overline{\tilde{v}}}{\partial \tilde{x}^2} + \frac{\partial^2 \overline{\tilde{v}}}{\partial \tilde{y}^2}\right) \tag{2.13}$$

As elaborated in §1.4.1, the above expressions are evaluated at each iteration in the batch of collocation points thanks to the use of the *autograd* package for differentiation in *PyTorch*. Accordingly, the expression to compute the total PDE loss is as given in Eq. 2.14.

$$\mathcal{L}_{PDE} = \sum_{j=1}^{3} \mathcal{L}_{e_j} = \frac{1}{N_c}\left\{\sum_{i=1}^{N_c} e_1^2\left(\tilde{x}_i\,\tilde{y}_i\right) + e_2^2\left(\tilde{x}_i\,\tilde{y}_i\right) + e_3^2\left(\tilde{x}_i\,\tilde{y}_i\right)\right\} \tag{2.14}$$

In the equation, the dummy index j ranges through the different residuals, while i sweeps through collocation points. As may be witnessed, the expression represents the mean-squared residual over the collocation batch size, N_c, which is the subsample of the entire collocation point space that is optimized at each iteration. Notably, each residual is squared to ensure both positive and negative residual values equally contribute to the loss term.

Reference Pressure Loss

The reference pressure loss term represents the deviation in the *PINN* prediction with respect to a known or inferred pressure value, forcing the predicted pressure field to take a unique solution.

As has been made clear so far, the aim of the *PINN* framework in this thesis is to reconstruct pressure from a set of velocity measurements, and, accordingly, pressure can be taken to be a *hidden* state. In order to reconstruct it, the PDE loss term is used, which allows to find pressure fields that comply with the spatial pressure gradient constraints specified by the RANS equations. It is direct to see that this approach can only provide pressure field uniqueness up to a constant, hence an additional constraint is needed to find the right unique solution. This constraint is contained in the reference pressure loss.

While its expression takes the same form as the data loss term, the source for the reference pressure can take various forms. For example, in this thesis, two different approaches have been used:

- in Chapter 3, where the solution is obtained via CFD, the reference pressure is readily available.
- on the other hand, in Chapter 4, where no pressure data is known except for the pressure tapping output, reference pressure values are computed via the C_p relation to velocity magnitude (Eq. 2.15), for which a region away from the cylinder and outside of the wake is selected, where the inviscid flow assumption holds.

$$C_{p,ref} = 1 - \frac{U^2}{U_\infty^2} = 1 - \frac{\sqrt{u^2+v^2}^2}{U_\infty^2} = 1 - \tilde{u}^2 - \tilde{v}^2 \tag{2.15}$$

According to the above, the expression for the reference pressure loss term for an experimental dataset where pressure is unknown would be as per Eq. 2.16. As stated, the expression takes the form of the squared error with respect to half the reference pressure coefficient averaged over the selected number of reference points, $N_{p,ref}$. It is noted that the expression takes this form due to the definition of dimensionless pressure given in Eq. 2.7, which results in $\tilde{p} = C_p \, / \, 2$.

$$\mathcal{L}_{p,ref} = \frac{1}{N_{p,ref}} \sum_{i=1}^{N_{p,ref}} \left(\tilde{p}_{pred} \left(\tilde{x}_i \, \tilde{y}_i \right) - \frac{1}{2} \, C_{p,ref} \left(\tilde{x}_i \, \tilde{y}_i \right) \right)^2 \tag{2.16}$$

An example of the effect of including the reference pressure loss term is attached in Figure 2.6, where the dimensionless pressure distribution and its x-gradient along a horizontal line at $y/D = 0.75$ are shown for the flow around a cylinder at $Re = 250$, where the reference data is provided by CFD [73]. Note that the reference pressure gradient is obtained through a central differencing scheme, while automatic differentiation provides a noiseless pressure gradient in the case of *PINNs*.

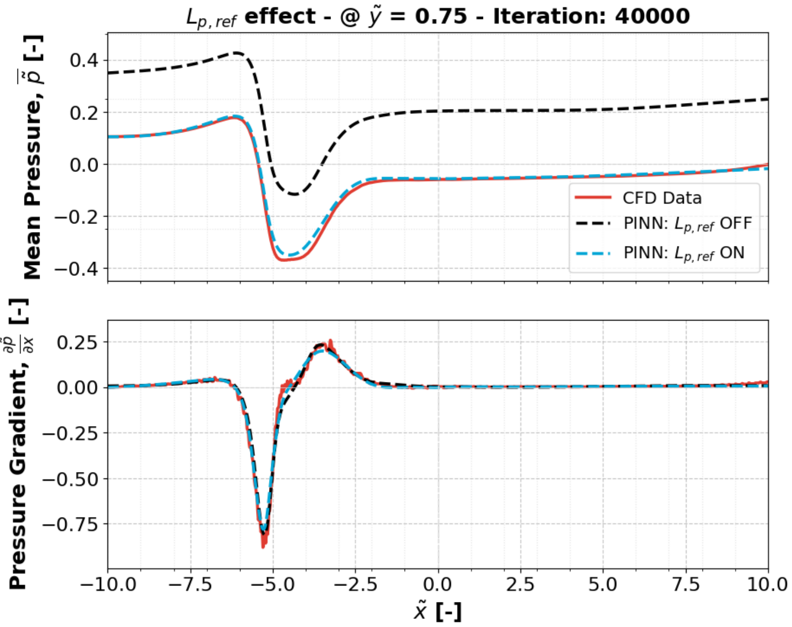


Figure 2.6: Effect of including $\mathcal{L}_{p,ref}$ - distributions of $\overline{\tilde{p}}$ and $\partial_{\tilde{x}}\overline{\tilde{p}}$ along a horizontal line at $y/D = 0.75$.

In this case, the solid red line depicts the CFD reference data, while the black and blue dashed lines, respectively, represent the cases before and after adding the $\mathcal{L}_{p,ref}$ term during optimization. As can be observed by looking at the pressure gradient plot, both *PINN* versions replicate the data pressure gradient with minimal deviation. However, a constant offset is seen in the pressure distribution when the reference pressure loss term is not included. This shows the need for this loss term to compute meaningful error metrics and sensible pressure values.

Reynolds Stress Loss

The Reynolds Stress loss, hereafter referred to as *prior loss*, aims to ensure that the main diagonal Reynolds Stress components predicted by the *PINN* are physically sensible by making sure they only take larger-than-zero values. The term *prior loss* is used because the information enforced is known before any results, thus prior information is embedded into the solution.

While it could be argued that this loss is not required because the PDE and data losses should already constrain the solution space to comply with physics, it is here argued that this is only partially true. On the one hand, data loss weighs all the squared error terms equally, which inherently biases loss optimization to locations and variables which contain the largest error magnitude.

To exemplify this, consider a point where $\overline{\tilde{u}}$ and $\overline{\widetilde{u'u'}}$ take the values of

$$\overline{\tilde{u}} = 0.5 \qquad \& \qquad \overline{\widetilde{u'u'}} = 0.05, \tag{2.17}$$

considering a 10% error on both variables would result in respective data loss contributions of

$$\mathcal{L}_{\overline{\tilde{u}}} = (0.45 - 0.5)^2 = 0.0025 \qquad \& \qquad \mathcal{L}_{\overline{\widetilde{u'u'}}} = (0.045 - 0.05)^2 = 0.000025 \tag{2.18}$$

Then, it can be deduced that, for unweighted data losses, the locations where Reynolds Stress components take values close to zero, the contribution to the total data loss term is reduced and thus small negative values could be allowed.

The second reason is that, because the x and y momentum equations in the PDE loss contain the pressure gradient terms, which are otherwise unconstrained, $\partial_x\overline{\tilde{p}}$ and $\partial_y\overline{\tilde{p}}$ can act as sinks for error, thus making it challenging for the NS loss to regularize the Reynolds Stress components. In contrast, this is not the case for $\overline{\tilde{u}}$ and $\overline{\tilde{v}}$, as they need to satisfy the mass conservation equation.

Since no reference is made to this type of loss in any of the literature reviewed, three different approaches were investigated to enforce positive diagonal Reynolds Stress values:

- on the one hand, a $ReLU$ activation function layer previous to the $\overline{\widetilde{u'u'}}$ and $\overline{\widetilde{v'v'}}$ output neurons is studied. The reason is that, given its definition equation, $ReLU(x) = max(0, x)$, all negative values are clipped to zero. However, provided that the initialization of the NN weights can take negative values, gradient information is prevented from flowing due to the null slope of $ReLU$ for negative values. The phenomenon is reflected in Figure 2.7a, where the evolution of all three Reynolds Stress component loss terms is shown as a function of the training epoch. Given the constant value of its loss term, it can be deduced that $\overline{\widetilde{u'u'}}$ is not optimized during training.
- the second proposal attempts to solve the above by using the $leaky$ version of $ReLU$, where a non-zero slope can be specified $\forall x < 0$, as given in Equation 2.19. While this addresses the stagnant variable during optimization, probably only a careful tuning of the slope, λ, can achieve optimal results in our primary intent. This is because a small slope of 0.01 causes negative values to vanish slowly, i.e. still be present at 40k iterations, while large slopes do not address the issue.

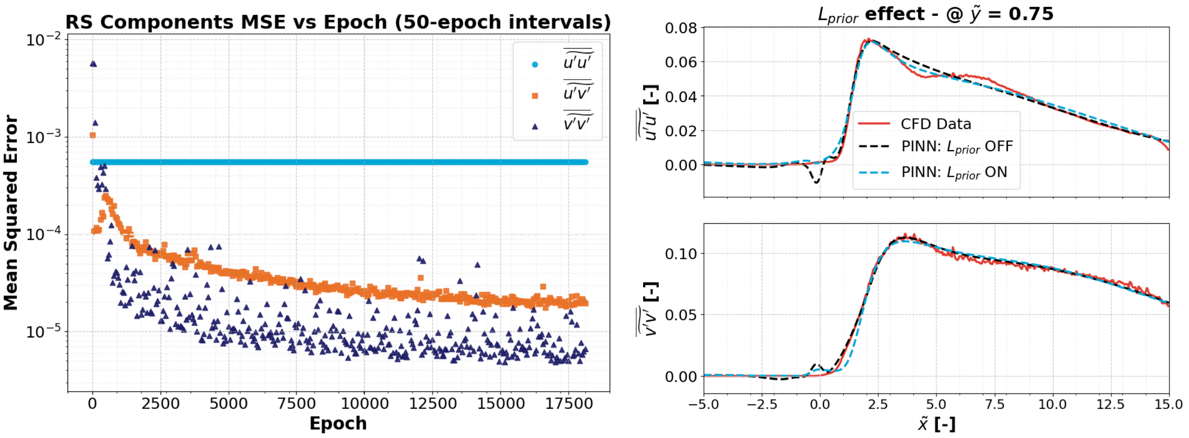
$$y = \begin{cases} \lambda \cdot x\,, & \text{if } x < 0 \\ x\,, & \text{if } x \geq 0 \end{cases} \tag{2.19}$$

- the third approach attempts to penalize negative normal Reynolds Stress components, $RS(i,i)$, directly. After testing a penalty term affecting only the largest negative value for each component at each iteration, a very slow convergence of the loss term was identified. To address this, a modification is made to include the mean of all negative values.

Accordingly, the loss term is defined as per Eq. 2.20, where N_{all} represents the batch size and $N_{p,i}$ is the number of points where the given $RS(i,i)$ component takes negative values.

$$\mathcal{L}_{prior,RS} = \frac{1}{N_{p,1}} \sum_{i=1}^{N_{all}} max\left(0, -\overline{\widetilde{u'u'}}(\tilde{x}_i, \tilde{y}_i)\right) + \frac{1}{N_{p,2}} \sum_{i=1}^{N_{all}} max\left(0, -\overline{\widetilde{v'v'}}(\tilde{x}_i, \tilde{y}_i)\right) \tag{2.20}$$

Below, Figure 2.7b shows the effect of adding the prior loss term on the predicted $\overline{\widetilde{u'u'}}$ and $\overline{\widetilde{v'v'}}$ components along a horizontal line at $y/D = 0.75$. In this case, the reference data corresponds to a CFD dataset of the flow around a cylinder at $Re = 1000$. On the one hand, it may be observed that both *PINN* models match the CFD prediction downstream of the cylinder origin, i.e. $\tilde{x} = 0$, with similar error levels. On the other hand, notable differences are found in the freestream, where the lack of fluctuations causes both variables to be null. In this region, it becomes evident that the penalization allows to avoid negative errors and with these, errors of up to ~ 0.01.



(a) MSE of Reynolds Stress Components - training with $ReLU$.

(b) Effect of prior loss on diagonal RS components.

While this is not of paramount relevance in terms of absolute values unless these components are of specific interest (which could presumably be fixed by weighting their data loss components), their gradients become non-zero, thus potentially adding error to the pressure gradient terms through the N-S loss.

Boundary Condition Loss

Finally, the boundary condition loss is conceived to enforce values or derivatives that must be fulfilled due to the nature of the problem or even physical laws. To exemplify, specifying velocity or pressure at the inlet would be a Dirichlet boundary condition bound to the given problem. On the other hand, enforcing the no-slip boundary condition on a solid wall would represent the imposition of a physical constraint.

In the present study, only boundary conditions bound to the cylinder surface are considered, since the focus is placed on surface pressure reconstruction. In particular, the following terms are implemented:

- no-slip condition, to ensure that viscous effects are fulfilled exactly at the cylinder surface, where PIV resolution of typical engineering flows can make it difficult to capture the boundary layer velocity profile.
- no through-flow, which prevents velocity predictions inward or outward of the cylinder surface from taking place.
- no fluctuations due to the presence of the wall, thus forcing Reynolds Stress components to take null values on the cylinder surface.

In order to evaluate these loss terms, a grid of points on the cylinder surface is generated and a forward pass, that is a *PINN* evaluation, is performed on them. Once the values of the output state variables are known, the various boundary condition loss terms can be calculated.

It can be deduced that, for the flow around a cylinder, the no-slip and no through-flow conditions require that the tangential and radial velocity components be computed at the cylinder radius. Bearing in mind that the Cartesian origin of coordinates is placed on the cylinder center, one can obtain them via the expressions in Eqs. 2.21 and 2.22 respectively, which consist of projections of the cartesian velocity components into the cylindrical system of coordinates.

$$\overline{\widetilde{u_\theta}}\left(\widetilde{x},\widetilde{y}\right) = -\overline{\widetilde{u}}\left(\widetilde{x},\widetilde{y}\right) \cdot sin\left(\theta\right) + \overline{\widetilde{v}}\left(\widetilde{x},\widetilde{y}\right) \cdot cos\left(\theta\right) \tag{2.21}$$

$$\overline{\widetilde{u_r}}\left(\widetilde{x},\widetilde{y}\right) = \overline{\widetilde{u}}\left(\widetilde{x},\widetilde{y}\right) \cdot cos\left(\theta\right) + \overline{\widetilde{v}}\left(\widetilde{x},\widetilde{y}\right) \cdot sin\left(\theta\right) \tag{2.22}$$

In addition, the relation of θ to the Cartesian coordinate system is given by Eq. 2.23.

$$\theta = arctan\left(\frac{\widetilde{y}}{\widetilde{x}}\right) \tag{2.23}$$

Finally, the no-slip and no-through flow boundary condition loss terms take the forms given in Equations 2.24 and 2.25 respectively. Observation of the expression leads to the identification of squared errors averaged over the boundary condition collocation points, N_{bc}, which are placed on the cylinder surface.

$$\mathcal{L}_{bc,no-slip} = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \overline{\widetilde{u_\theta}}^{2} \left(\widetilde{x}_i,\, \widetilde{y}_i\right) \tag{2.24}$$

$$\mathcal{L}_{bc,nothrough-flow} = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \overline{\widetilde{u_r}}^{2} \left(\widetilde{x}_i,\, \widetilde{y}_i\right) \tag{2.25}$$

On the other hand, the last term, which forces the Reynolds Stress components to take null values at the cylinder surface, takes the form shown in Equation 2.26, where the interpretation of the expression is analogous to the previous.

$$\mathcal{L}_{bc,no\,fluct} = \frac{1}{N_{bc}} \sum_{i=1}^{N_{bc}} \overline{\widetilde{u'u'}}^{2} \left(\widetilde{x}_i,\, \widetilde{y}_i\right) + \overline{\widetilde{u'v'}}^{2} \left(\widetilde{x}_i,\, \widetilde{y}_i\right) + \overline{\widetilde{v'v'}}^{2} \left(\widetilde{x}_i,\, \widetilde{y}_i\right) \tag{2.26}$$

Although these loss terms are not implemented initially, their effect has been studied separately and will be discussed in further detail in Chapters 3 and 4 due to their relevance to the research questions.

Total Loss

Having explained and justified the role of each loss term used in the present thesis, the expression for the key performance indicator governing the NN training is shown. As introduced in §1.4.1, numerous papers have exposed the possible benefits of including static or dynamic weights in different loss terms to improve learning dynamics and/or the global optimum. However, these have been disregarded at first in the present work and will be discussed in the following chapters given their problem-specific nature.

Consequently, the expression for the total loss takes the form of Equation 2.27, which is simply an unweighted sum of the terms covered above.

$$\mathcal{L}_{Tot} = \mathcal{L}_{data} + \mathcal{L}_{PDE} + \mathcal{L}_{Prior} + \mathcal{L}_{p,ref} \tag{2.27}$$

2.2.3. Training process

In this section, the specific tools and hyperparameters that have been selected to control the initialization and NN training process are exposed. Furthermore, every test shown has been performed on the CFD data of the flow around a non-lifting cylinder that will be exposed in Chapter 3.

NN construction and initialization

As a reminder from §1.4.1, once the loss function has been defined, the neural network architecture needs to be created and its learnable parameters initialized before the training process can be started.

On the one hand, the initial neural network architecture has been defined arbitrarily, as the effect of the number of hidden layers and neurons per hidden layer will be exposed in detail in the coming chapters. Hence, the initial neural network architecture features 8 hidden layers and 50 neurons per hidden layer, which amounts to a total of $N_\theta \approx 21000$ trainable parameters for the entire network.

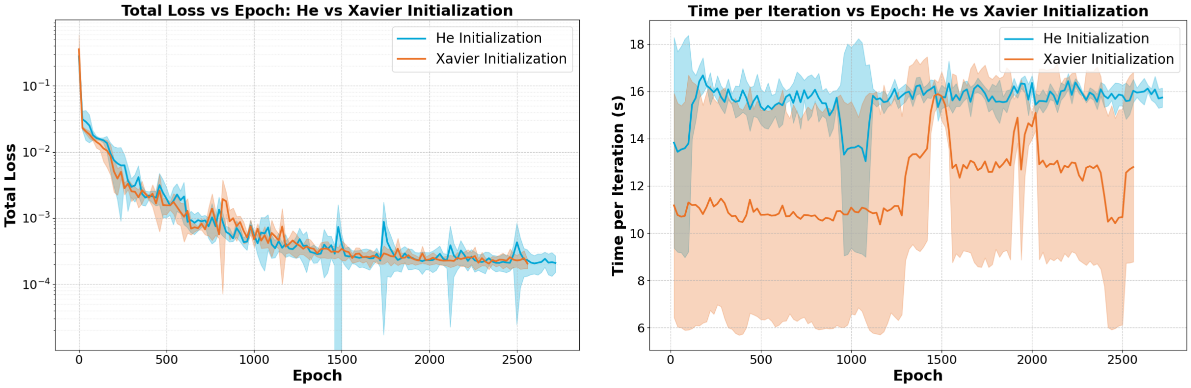
Furthermore, two initialization schemes for the weights and biases, namely Xavier and He [24] are compared to define the most suitable one given the NN architecture and loss function definition. Note that the two schemes are based on similar principles, as they are conceived to keep a similar variance to the activations across different layers in order to stabilize the flow of gradients along the network.

In order to achieve this, they modify the definition of the distribution from which they randomly sample the weights and biases. On the one hand, Xavier is inherently more suitable for symmetric activation functions such as *tanh* or *sigmoid*, as it scales the variance of the distribution with the number of input and output connections at each neuron. Conversely, He is inherently more appropriate for non-symmetric activation functions like *ReLU*, given that it merely uses the number of input connections at each neuron to define the variance of the sample distribution. The respective expressions are included in Equation 2.28, where the number of input and output connections are denoted by n_{in} and n_{out} respectively.

$$\sigma^2_{Xavier} = \frac{2}{n_{in} + n_{out}}; \qquad \sigma^2_{He} = \frac{2}{n_{in}} \tag{2.28}$$

With the aim to define which initialization scheme is more suitable to the problem of reconstructing surface pressure for the dimensionless, two-dimensional flow around a non-lifting cylinder, batches of 4 runs are conducted with each initializer, tracking the evolution of the various loss components during training.

In Figure 2.8a, the total loss function evolution during training has been represented for the He and Xavier initialization schemes. Note that, since four different runs were used, the mean is represented by the solid line, while the shaded bands reflect the standard deviation across the batch. Looking at the results displayed, it can be stated that both initialization schemes present very similar learning dynamics, with the Xavier scheme runs presenting reduced variance especially around the plateau region.



(a) Total loss evolution during training - He vs Xavier.

(b) Mean time per epoch - He vs Xavier.

Additionally, given the small variance observed between the two algorithms in the optimization of the loss function, the mean time per iteration for both schemes is shown in Figure 2.8b. Note that this plot is analogous in nature to that shown for the total loss, where the solid line reflects the mean across all four runs in the batch and the bands represent the standard deviation across the set. In this case, a major difference can be witnessed, since the Xavier scheme seems to be $\sim 25\%$ more efficient than its counterpart. Focusing on He, the different runs present a mean time per epoch of around $\sim 16\,s$ consistently and for the vast majority of the training process. On the other hand, results for Xavier offer a much more significant variance, with the upper band nearing the performance of He and the lower band touching mean execution times per epoch as low as $\sim 6\,s$, while the mean is kept at around $\sim 12\,s$.

From the above it is concluded that, despite not presenting a major advantage in terms of learning dynamics and noting that the mean time per iteration is not corrected for possible variations in computer workload, the Xavier scheme is selected given the potential time savings it could offer.

Optimizer and batch size

Regarding the control of the training process, one of the most relevant decisions is the use of the optimizer. As covered in §1.4.1, this determines the set of calculations that lead to the update of the NN trainable parameters at each iteration. In the present work, the Adam optimizer [26], which combines first and second-order statistical moments of the loss gradient with respect to the trainable parameters, has been selected.

The selection of this specific optimizer lies in its main advantages, which derive from its formulation and make the algorithm efficient and robust to noisy data altogether. These characteristics have led to its wide-spread use in similar use-cases across the literature [51] [76] [27] [31].

Furthermore, the Adam optimizer is often used in *PINNs* as a means to find an initial convergence of the solution towards the global minimum. In these occasions, it is often followed by additional iterations on the *Limited-memory Broyden–Fletcher–Goldfarb–Shannon* optimizer, more commonly referred to as L-BFGS. This optimizer, instead of focusing on the gradient, attempts to reconstruct the inverse Hessian to allow for more significant optimization steps. However, provided that it is a full-batch approach, which means it requires execution over the entire set of data points at each iteration, it typically only finds its application in simpler flows [76]. Accordingly, this second optimization step has been avoided in the present work, focusing on the mini-batch-based Adam optimizer.

Additionally, the batch size has been kept constant throughout the thesis to approximately $\sim 1/5$ of the total number of measurement points, unless otherwise specified. As a reminder of previous sections, this causes the number of iterations per epoch to be 5 on average.

Learning rate and scheduler

One of the most relevant hyperparameters that must be set is the learning rate, which controls the extent to which learnable parameters are updated at each training iteration. In order to comprehend the relevance of accurately tuning this parameter, one should attend to how gradient descent-based optimization methods operate.

Following the discussion in §1.4.1, one can understand its relevance by looking, for instance, at a simplified 2D optimization space with a global minimum and several local minima. In this scenario, too small a learning rate could cause the optimization process to be halted at a local minimum. The reason being that taking too small steps would prevent the optimization state from escaping the small basin. On the other hand, a learning rate that is too large could quickly converge to the main basin but be unable to take steps small enough to reach the global minimum point.

As can be deduced, an appropriate value for the learning rate, η, is problem-dependent, due to the influence of the various agents (the loss function, the neural network architecture, etc.) on the shape of the optimization space. Not only this, but its optimal value to minimize training time and final loss changes during training as it is influenced by the local optimization landscape. Consequently, a number of strategies exist when dealing with the learning rate, from fixed values, to schedulers that adjust its value according to the learning dynamics or even dynamically-adjusted learning rate schemes [26].

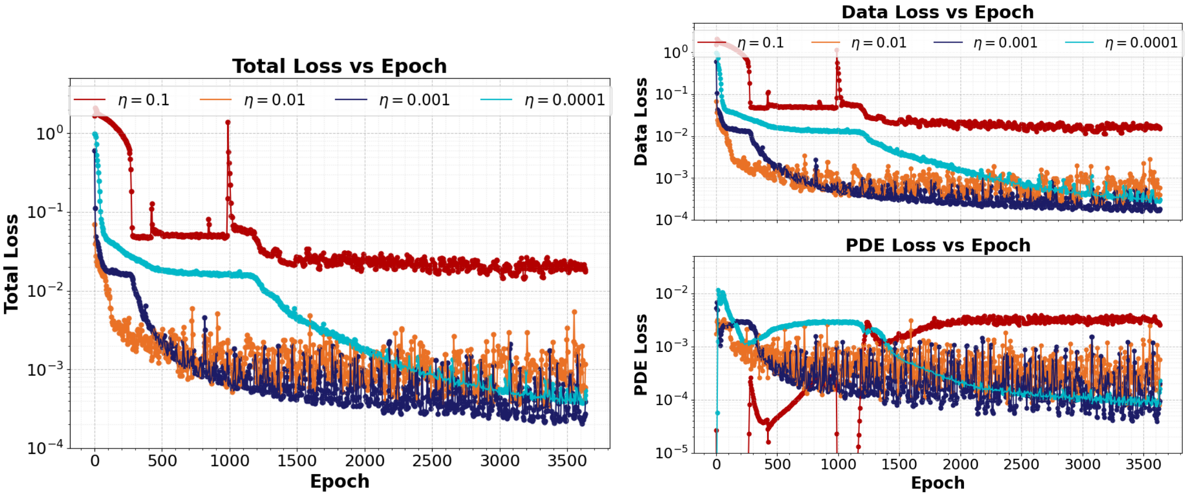
In the present thesis, the decision is made to use a scheduler that accommodates a dynamic learning rate, granting more granular control on the progression of values it takes during training. In particular, the *ReduceLROnPlateau* algorithm from the Pytorch library is used. This specific scheduler is based on the principle that an optimal progression of the learning rate is such that its value must decay gradually, taking steps when the loss is not reduced over a given set of epochs [72].

However, prior to the definition of the scheduler parameters, a test is run to get an estimate of the value that would be appropriate for the initial learning rate. The test consists of running 20000 training iterations with different values of the learning rate, which are kept constant throughout training. Additionally, this test helps form an understanding about how a too-small or too-large value could be identified during training, thus aiding in the subsequent selection of the scheduler parameters.

The results of the test are shown in Figure 2.9, where the various loss components have been represented during the training epoch for each of the selected η values, namely from $\eta = 0.1$ to $\eta = 0.0001$, with an order of magnitude variation.

Focusing on Figure 2.9a first, where the total loss optimization is reflected, various observations can be made:

- on the one hand, the largest learning rate, $\eta = 0.1$, is seemingly too large even at the beginning, as it takes $\sim 300\,it$ to converge into the main basin. In the following iterations, it appears that it is unable to optimize the learnable parameters any further, as the loss establishes at $\sim 2 \times 10^{-2}$.
- among the other runs, the second largest learning rate, $\eta = 0.01$, is the fastest to progress into the main basin and towards the global optimum.
- while the smaller learning rates eventually evolve towards the same global optimum, it appears that they take too small steps and therefore find an initial plateau at the main basin, where $\mathcal{L}_{tot} \sim 2 \times 10^{-2}$, before progressing into the optimal sink.
- in addition, the behavior of the various curves as the loss is reduced gives insight into how the optimal learning rate changes during training. For example, the $\eta = 0.01$ curve in orange has virtually no noise up to $epoch \sim 500$ with $\mathcal{L}_{tot} \sim 2 \times 10^{-3}$. Then, $\eta = 0.001$ starts to fluctuate significantly at $\mathcal{L}_{tot} \sim 5 \times 10^{-4}$, while the smallest η curve does not present noisy behavior in the range shown.



(a) Total loss vs epoch.

(b) Data (top) and PDE (bottom) losses vs epoch.

Figure 2.9: Effect of fixed learning rate value on loss components during training.

At the same time, the two graphs shown in Figure 2.9b provide further information on how the two most significant loss components are affected by the learning rate and behave during training:

- In particular, the PDE loss initially takes near-zero values for the highest and lowest learning rates, suggesting that the variables initially take near-zero values throughout the domain, which is to be expected given the initialization scheme for weights and biases.
- Consequently, the data loss initially appears to dominate the learning process, which can also be deduced from the relative magnitudes of both loss terms.
- Finally, it can be argued that the PDE loss potentially benefits more from the reduction in learning rate as the global optimum is approached. Proof of this is the wider noise amplitude in the $\eta = 0.01$ and $\eta = 0.001$ curves with respect to the same curves for the data loss. The consequence is that the PDE loss plateaus earlier than the data loss for both cases.

From the above observations, several of the parameters that govern the learning rate scheduler are defined. However, it is of interest to first introduce its formulation, so that the various parameters are understood by the reader.

The *ReduceLROnPlateau* algorithm starts with a fixed initial learning rate. Once an initial number of epochs have passed, the algorithm starts tracking the total loss at each epoch and, when this is not lowered over a number of epochs (defined by the *patience*), a step is taken to reduce the learning rate via Equation 2.29. As deduced from the expression, the reduction factor must be $f < 1$. After η is updated, a *cooling* number of epochs halts the scheduler to let the optimizer find an appropriate descent path. This process is repeated until the minimum specified learning rate, η_{min}, is attained, thus forming a staircase reduction trend.

$$\eta_{new} = f \cdot \eta_{old} \tag{2.29}$$

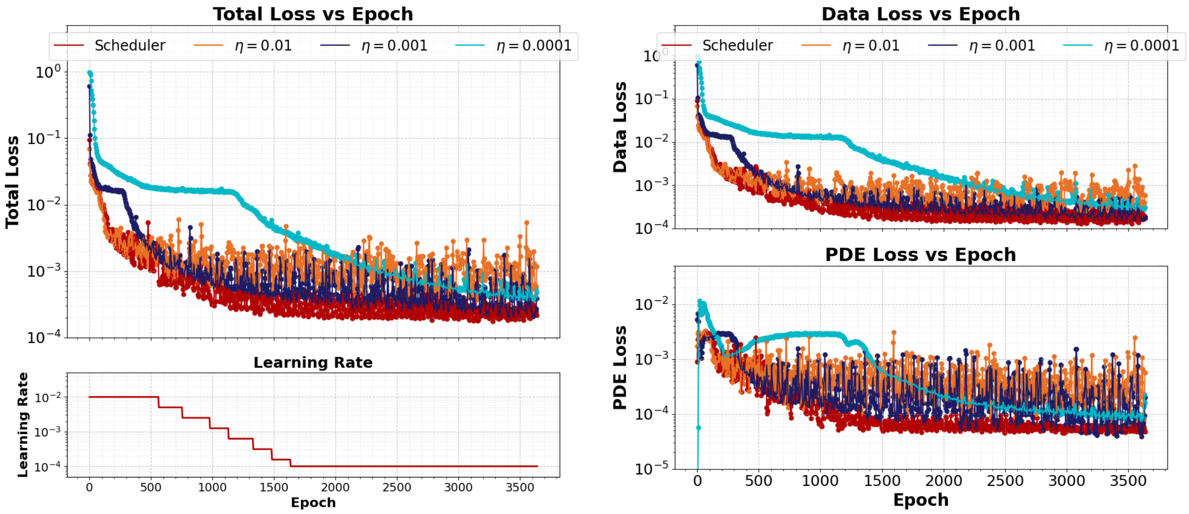
With this in mind, the parameters set from the results from the above study are:

- the initial learning rate is set to $\eta_0 = 0.01$.
- the number of initial iterations before the scheduler starts operating is set to 500.
- the minimum learning rate is set to $\eta_{min} = 1 \times 10^{-4}$.
- the reduction factor at each scheduler step is set to $f = 0.5$, given the large variation in response observed between the $\eta = 1 \times 10^{-3}$ and $\eta = 1 \times 10^{-4}$ curves.

Further study of how the scheduler works led to the introduction of a slight variation with respect to the original algorithm. By design, the scheduler stores the last loss term among all iterations in an epoch. If instead, the best total loss among all the iterations within an epoch is fed to the algorithm, robustness is gained to the specific batch of points used in the last iteration, making convergence smoother. Besides, an additional modification is made to restore the trainable parameters from the last best loss right after each learning rate update, avoiding an escape from the basin if the learning rate is too large.

With this approach, the *cooldown* parameter is fixed to 100 epochs, while an optimal patience for this problem is found at 50 epochs. This presents as a compromise between too frequent learning rate reductions, leading to stagnation at local minima, and too disperse updates of the learning rate, leading to slow convergence.

The resulting learning curves are represented in Figure 2.10 in conjunction with the $\eta = 1 \times 10^{-2}$, $\eta = 1 \times 10^{-3}$ and $\eta = 1 \times 10^{-4}$ curves.



(a) Total loss vs epoch.

(b) Data (top) and PDE (bottom) losses vs epoch.

Figure 2.10: Fixed vs scheduler-based dynamic learning rates effect on training.

On the one hand, Figure 2.10a to the left reflects not only the total loss but also the progressive reduction in the learning rate parameter of the scheduler run as the training takes place. About loss dynamics, it may be said that the scheduler run closely follows the $\eta = 0.01$ curve initially, which is sensible given that $\eta_0 = 0.01$. Slightly after epoch 500, when the initial holding period ends, the first learning rate reduction is triggered, which already positively impacts the total loss dynamics by reducing it further than any of the static η lines. Subsequently, steps are taken approximately every 110-150 epochs, which is just a few times the *patience* of 50 epochs. Finally, the minimum learning rate of $\eta_{min} = 1 \times 10^{-4}$ is reached at around 1600 epochs, after which the total loss is reduced only to a limited extent but in a smooth manner, signaling the plateau is reached.

Additionally, Figure 2.10b to the right provides further information about how the two loss components behave as the learning rate is progressively reduced. Firstly, it is worth noting the noticeable reduction in the noise amplitude when compared to all the fixed learning rate curves except for the $\eta = 1 \times 10^{-4}$ curve. Secondly, it can be appreciated how the PDE loss appears to plateau already on epoch 1500 at a final value of $\mathcal{L}_{PDE} \approx 5 \times 10^{-5}$, while the data loss component is still progressively reducing until $\mathcal{L}_{data} \approx 1.2 \times 10^{-4}$. Hypothetically, this could be due to the relative magnitude of both terms, which shall be discussed further in subsequent sections.

Overall, it can be argued that the NN version with scheduler presents a smoother yet faster convergence towards the respective loss term plateaus, achieving the intent of the algorithm. It is here noted that, in the subsequent sections, the training stopping criterion has been set to whichever is the minimum between the maximum number of iterations of $it_{max} = 20000$ or 500 epochs with no loss improvements. Furthermore, the final results of each version correspond to the set of trainable parameters yielding the lowest recorded loss before the convergence criterion is reached, and not necessarily the final set of parameters.

2.2.4. Framework overview

To aid the reader in visualizing the interconnectedness of the entire *PINN* framework, the block diagram in Figure 2.11 is referenced.

From the top-left, the representation includes the NN parameters initialization under the Xavier scheme. Additionally, the *PINN* takes in $\{\tilde{x}, \tilde{y}\}$ coordinate pairs and outputs all six time-averaged flow variables at the given spatial locations. Subsequently, the outputs are used to compute the various loss components - while the automatic differentiation (AD) block is required to yield the spatial derivatives for all three PDE loss equations, the rest of loss components do not require further steps other than the use of labeled data to yield the data loss.

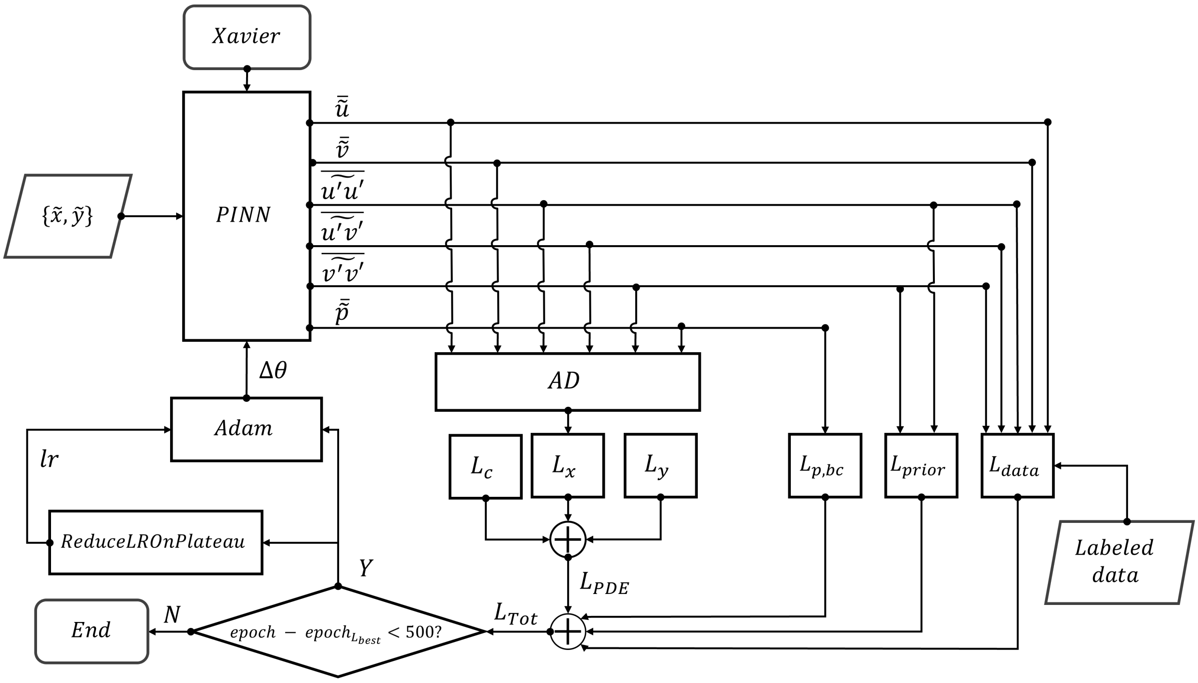


Figure 2.11: *PINN* framework used in the thesis.

Furthermore, the addition of all loss components yields the total loss, used to check the convergence criterion. That is, if less than 500 epochs have passed without any loss minimization, the total loss is fed to the *ReduceLROnPlateau* scheduler and the Adam optimizer, where the former is used to compute the appropriate learning rate from the learning dynamics. Finally, the optimizer provides the adequate adjustment of the weights and biases, $\Delta\theta$, and the network is updated. This process is repeated at a randomly sampled batch of measurement points at each iteration until the aforementioned convergence criterion or the maximum number of pre-specified iterations are satisfied.

3
PINNs performance on CFD simulations

In the first place, the performance of the *PINNs* framework discussed in Chapter 2 is analyzed on the CFD data of the two-dimensional flow around a non-lifting cylinder. This allows to assess the capability of the algorithm in the prediction of surface pressure from data without measurement noise or reflections, also benefiting from high spatial resolution close to the solid boundary of interest, as well as the ability of CFD simulations to provide data with no measurement error.

3.1. Dataset Generation

Initially, the dataset of the two-dimensional flow around a cylinder at $Re_D = 250$ from [73] was considered. However, execution of the *PINN* algorithm on the time-averaged flow solution led to the conclusion that the mesh size is too coarse to properly resolve the high spatial gradients around the shear layer and wake, causing too large a discrepancy in the data-calculated residuals.

Accordingly, the decision is made to generate a dataset, for which the *STARCCM+* software is used.

3.1.1. Pre-processing

Flow regime

Firstly, the various geometric and flow parameters are selected such that the turbulent flow regime is achieved, which is desirable as a realistic application for an engineering or research test-case.

Accordingly, the experimental data of the drag coefficient as a function of the Reynolds number for the flow around a cylinder shown in Figure 3.1 is analyzed. As may be witnessed, annotations on the characteristics that define the various flow regimes are included, as taken from [4].

From such annotations, a constraint for the flow regime can already be established, as distinct turbulent phenomena arise for $Re \geq 1000$. Above this Reynolds number value, while separation happens at the forward face of the cylinder, namely $\theta_{sep} \leq 90°$, the flow transitions into the turbulent regime at the shear layer, giving place to a fully-turbulent wake from the vortex-shedding. In turn, this translates into a virtually constant value for the drag coefficient, with $C_D \approx 1$.

In addition, it is of relevance to consider the vortex-shedding frequency, as this largely determines the simulation time-step size that is sufficiently small to properly capture flow dynamics. In this case, the Strouhal number is regarded, which, for periodic flows, defines the relevance of the flow oscillations produced by inertial forces relative to velocity changes induced by the convective acceleration of the fluid flow [33].

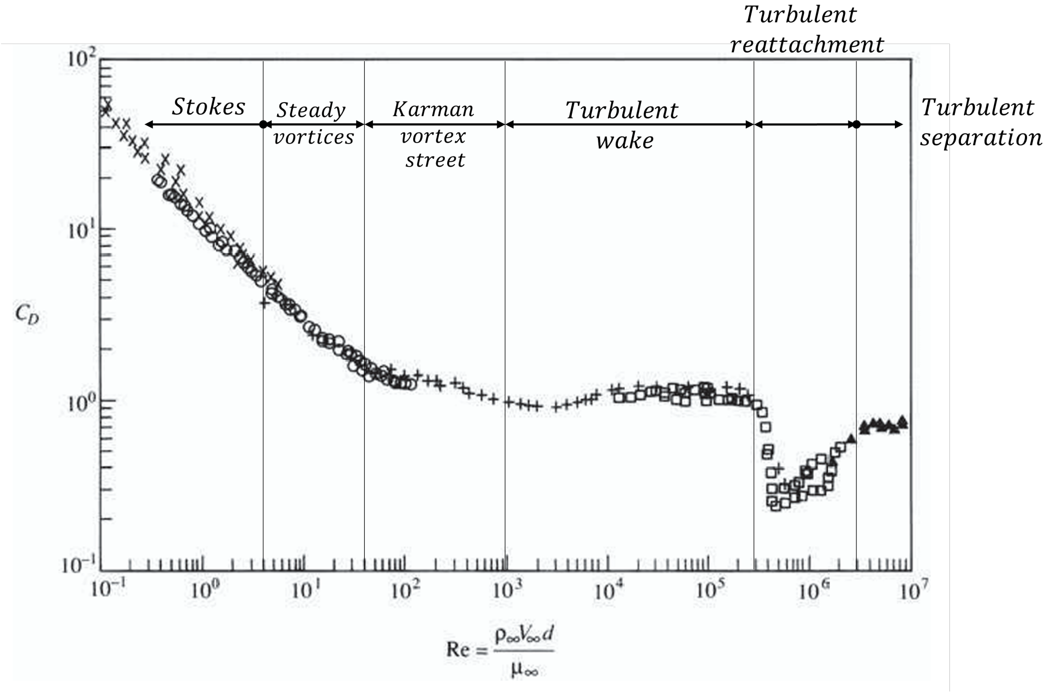


Figure 3.1: Drag coefficient vs Reynolds number for a smooth cylinder - adapted from [4].

Furthermore, the definition of the Strouhal number for the flow around a cylinder with diameter D and free-stream velocity U_∞ is given by Equation 3.1. In this expression, f represents the frequency at which eddies are shed off the cylinder surface.

$$St = f\,\frac{D}{U_\infty} \tag{3.1}$$

In Figure 3.2 next, experimental measurements around a smooth cylinder are used to show the dependency of the shedding phenomenon to the turbulent regime, via the Reynolds number. As observed, a virtually constant value of $St \approx 0.22$ is held in the aforementioned turbulent flow regime, where the turbulent transition happens at the wake after laminar separation.

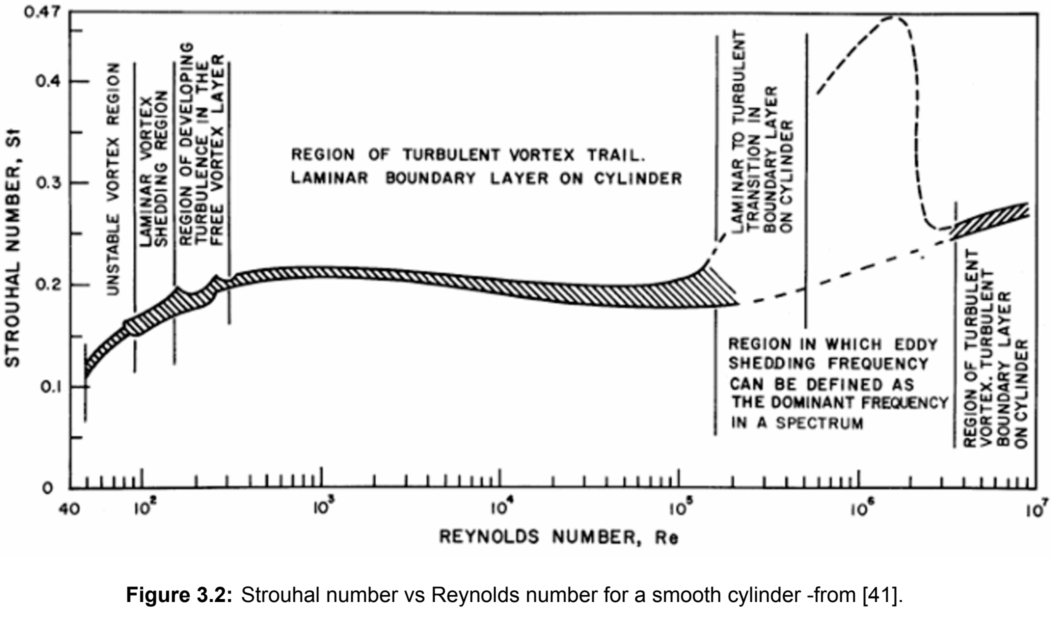


Figure 3.2: Strouhal number vs Reynolds number for a smooth cylinder -from [41].

Therefore, considering the above information leads to the selection of $Re = 1000$ as a desirable value. On the one hand, it provides a fully-turbulent wake to test the ability of *PINNs* to reconstruct pressure in the non-linearities of turbulent flow motion. At the same time, choosing the lowest possible Re number for this regime ensures that the shedding frequency (and hence the simulation time step) are kept at an acceptable value, due to their direct proportionality to the free-stream speed. This relation is included in Equation 3.2, which is a mere rearrangement of Equation 3.1 and has been particularised for $St \approx 0.22$.

$$f \,=\, St\,\frac{U_\infty}{D} \approx 0.22\,\frac{U_\infty}{D} \tag{3.2}$$

Finally, the following characteristics have been selected for the rest of variables:

$$D = 1\,m; \qquad U_\infty = 5\,\frac{m}{s}; \qquad \rho = 1\,\frac{kg}{m^3}; \qquad \mu = 5\times10^{-2}\,Pa\cdot s; \tag{3.3}$$

Note that D and ρ are selected such that the non-dimensionalization process is simplified. Furthermore, to ensure a sufficiently high free-stream speed can be selected, the fluid kinematic viscosity has been increased significantly with respect to standard values for air.

Computational domain and grid

After definition of the cylinder geometry, good practice standards are used to define a sufficiently large computational domain to prevent boundary conditions from affecting the flow field solution. Furthermore, since the focus is on the two-dimensional flow solution, the third dimension is kept to a minimum, thus saving computational effort without the need to force a purely two-dimensional solution.

With this information, the resulting computational domain is as shown in Figure 3.3. From left to right, the front, side, and perspective views of the computational domain are shown. As observed in the measurements, five diameters are left upstream and on each side of the cylinder, while fifteen characteristic lengths are left downstream to give sufficient space for the turbulent structures to fully develop. Additionally, the different types of boundary conditions used are shown, including a uniform velocity inlet, the pressure outlet and symmetry planes for the side and top/bottom faces.

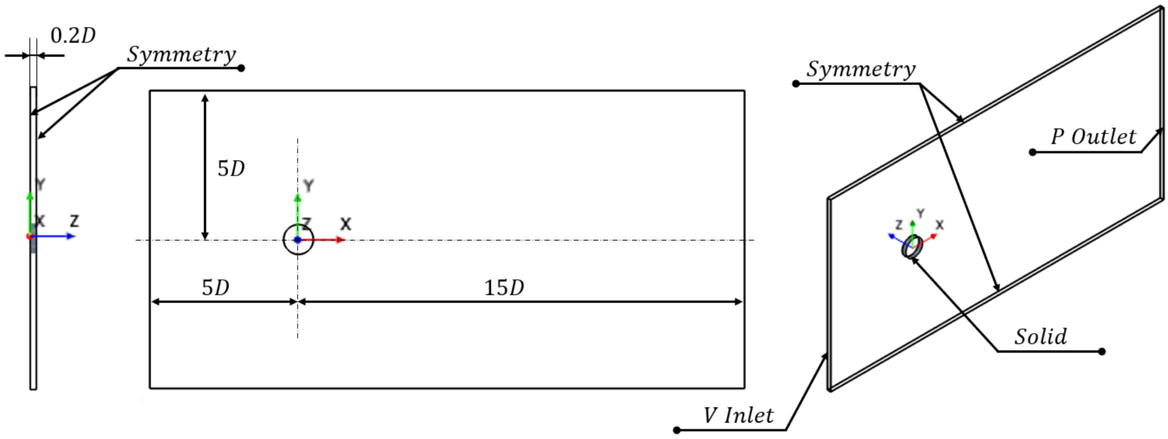


Figure 3.3: Front, side and trimetric views of the computational domain with boundary condition types.

In addition, the decision is made to resolve the boundary layer, avoiding the use of wall functions. In accordance, an inflation layer is generated to ensure that the wall-induced gradients are properly captured. Moreover, the experimentally-derived expression for the turbulent boundary layer thickness along a flat plate is used to get a proxy for the target inflation layer thickness. This is shown in Equation 3.4:

$$\delta_{99} = \frac{0.16 \cdot D}{Re^{1/7}} = \frac{0.16 \cdot 1\,m}{1000^{1/7}} = 0.060\,m \tag{3.4}$$

With the above in mind and targeting a wall y^+ below unity to ensure a proper resolution of the wall-normal gradients, the following settings are used in the inflation layer:

- total inflation thickness of $\delta = 0.06m$.
- a number of inflation cell layers equal to $N_{layers} = 10$.
- growth rate equal to $GR = 1.5$.

Furthermore, three additional refinement cubes are placed to ensure the gradients around the cylinder, shear layer, and wake are properly captured:

- high refinement block with target size of $0.045\,m$ around the cylinder, extending $1D$ upstream, $1.5D$ to the top and bottom and $2D$ downstream of the cylinder.
- medium refinement block with target size $0.060\,m$, placed right downstream of the previous block and having a size of $3D \times 3D$.
- low refinement block with target size $0.08\,m$, immediately downstream and extending until $12D$ from the cylinder center.
- rest of computational domain with target size of $0.6\,m$.

The above settings result in a polyhedral computational grid with the characteristics described in Table 3.1 below:

Table 3.1: Mesh Metrics

	Num. Cells	**Max. Skewness**	**Max. Wall y^+**
Mesh	105651	69°	0.1
Quality criterion	-	<89°	< 1.0

Finally, a close-up of the inflation layers and the computational mesh in the proximity of the cylinder is shown in Figure 3.4. In the image, the mesh outline is displayed on top of mean x-velocity contours, which allows to verify that the approximation of the boundary layer thickness is appropriate. This is because the inflation layers enclose the entire boundary layer before the flow separates from the surface.

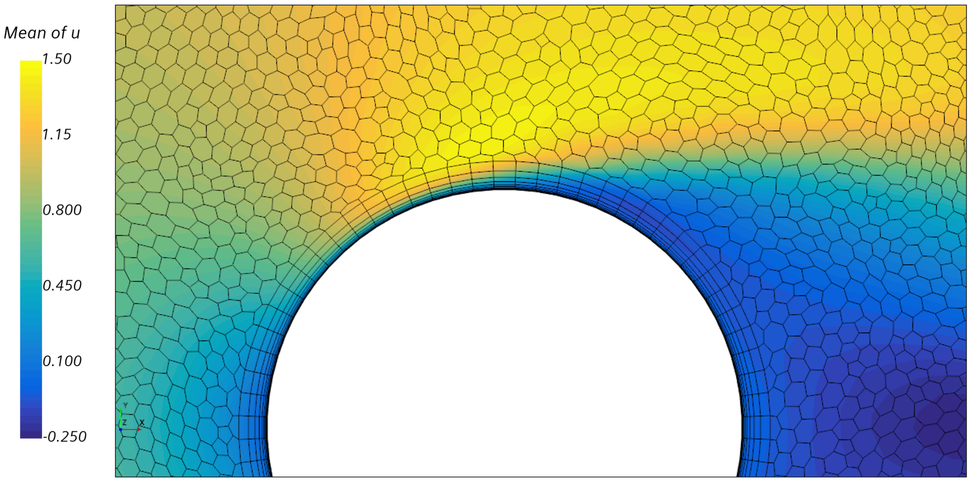


Figure 3.4: Mesh outline and mean x-velocity contours.

Turbulence modeling and numerical schemes

Once the computational mesh is generated, the various physical models that define the conservation equations to be solved are defined, as well as the appropriate numerical schemes that allow to discretize the solutions.

In the first place, the flow is assumed to be incompressible, provided that the expected maximum speed is well below the $M = 0.3$ threshold. More importantly, the Reynolds Stress turbulence treatment is selected, such that additional transport equations are added for all six components. The main reason behind this is the further ability of RSM to accommodate flow anisotropy, while avoiding the high computational cost of DES or LES, which is also acceptable provided that the intent is to obtain the mean flow variables. In particular, given the relevance of the near-the-wall region in the reconstruction of surface pressure, the Elliptic Blending variant proposed in [47] is used, which enhances transition from the outer region to the viscous sub-layer especially in flows with strong curvature.

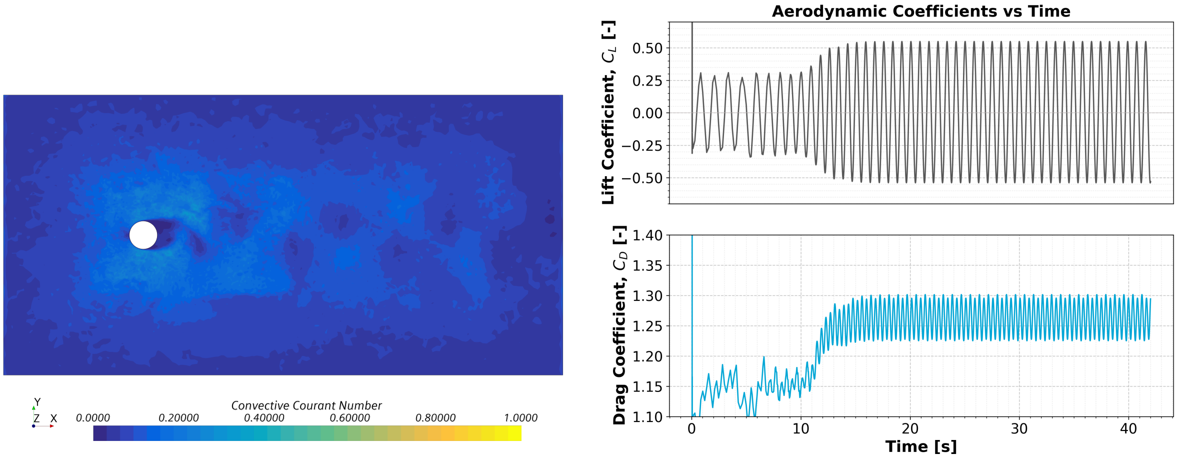
With the above, the number of transport equations amounts to 12:

1. mass conservation equation.
2. conservation of momentum in all three directions.
3. six, for all Reynolds Stress components.
4. one for the elliptic blending parameter α.
5. one for the turbulent dissipation rate.

Secondly, the implicit time-stepping approach is selected, not only because it can accommodate larger time steps thanks to its unconditional stability, but also due to its robustness to the complex stability of the RSM equations. In particular, a first-order time-stepping scheme is used with a manual reduction of the time-step size. In particular, since the flow is initialized from rest, a time step of $\Delta t_0 = 0.1\,s$ is kept initially to trigger the alternating vortex pattern. Subsequently, this time step is progressively reduced to $\Delta t_0 = 0.002\,s$ for the last ten cycles, which are used to get the time-averaged solution.

In Figure 3.5a, contours of the *Courant-Friedrichs-Lewy* criterion are shown along the $Z/D = 0.1$ plane. In order to understand its significance, its definition in Equation 3.5 is referenced, where the CFL number is computed at each cell, i. As seen, the criterion provides the ratio of the Euclidean distance traveled by the local flow at each time-step, $|U_i|\;\Delta t$, to the characteristic cell length, Δx_i.

$$CFL_i = |U_i|\,\frac{\Delta t}{\Delta x_i} \tag{3.5}$$



(a) Contours of CFL criterion - symmetry plane.

(b) Temporal evolution of lift (top) and drag (bot) coefficients.

Accordingly, an acceptable time-step is such that the flow does not travel more than a cell's distance over the course of an iteration, leading to $CFL \leq 1$. While this is not a hard stability constraint for implicit time stepping, it is checked as good practice for proper resolution of the unsteady phenomena.

Overall, for the plane shown, the maximum value of the criterion is $CFL \approx 0.3$, while for the entire domain is slightly above 1, with $CFL_{max} \approx 1.6$. In the contours, it is observed that the most critical location is the shear layer right after separation and into the early wake. In this area, even if the grid size is refined by the additional control volumes, the the turbulent eddies shed off the cylinder induce large local velocities.

On the other hand, the temporal evolutions of the lift and drag coefficients are plotted in Figure 3.5b. In this case, both lines depict a similar picture, where an initial transition process takes place as the oscillatory flow motion is settled. Specifically, at around $T = 20\,s$, both coefficients enter into a statistically steady, periodic, motion with amplitudes of $\Delta C_L \approx \pm 0.55$ around $C_{L,eq} = 0$ and $\Delta C_D \approx \pm 0.03$ around $C_{D,eq} = 1.26$ of which the last 10 flow cycles are used to perform the time-averaging process.

As additional information about the level of convergence achieved in the simulation, Tables 3.2 and 3.3 next display the residual for each conservation equation, averaged over the last 1000 iterations. It is here noted that the method to compute residuals in *STARCCM+* is such that the value at each iteration is normalized with the maximum of the set, hence forcing residuals to be comprised in $0 \leq R \leq 1$.

Furthermore, assessment of the values for each equation leads to the conclusion that all residuals show satisfactory levels of convergence except for the cross Reynolds Stress components, especially $u'v'$, which is kept in mind going forward in the present Chapter.

Table 3.2: Residual averages - last 1000 iterations.

Mass	**X-mom**	**Y-mom**	**Z-mom**	α	ε
2.2×10^{-4}	1.4×10^{-2}	6.1×10^{-2}	1.7×10^{-2}	5.8×10^{-5}	9.3×10^{-3}

Table 3.3: Reynolds Stress components residual averages - last 1000 iterations.

$u'u'$	$u'v'$	$u'w'$	$v'v'$	$v'w'$	$w'w'$
2.9×10^{-3}	0.2	5.2×10^{-2}	3.4×10^{-3}	5.5×10^{-2}	2.2×10^{-3}

3.1.2. Flow solution

Once the geometry, computational grid, simulation parameters and convergence level of the simulation have been exposed, it is meaningful to display some of the mean flow results, as these are referenced extensively along the following sections of the present Chapter.

As it has been hinted in the previous section, provided that the present thesis focuses on two-dimensional applications of the *PINN* framework, the region of interest corresponds to the flow imprint on the symmetry plane, namely $Z/D = 0.1$. The resulting dataset, thus, is formed by 38600 data points, corresponding to all the cells on that particular plane.

In the first place, contours of mean dimensionless velocity magnitude are represented in Figure 3.6, limiting the range to the cylinder proximity. Furthermore, in Figure 3.7 the pressure coefficient distribution on the cylinder surface is included, where $\theta = 0°$ corresponds to the left-most point on the cylinder.

Observation of the contours allows to identify a symmetric flow pattern around the $Y/D = 0$ line. Besides, the stagnation point is highlighted by the red arrow at $\{x, y\} = \{-r, 0\}$, characterized by a null local speed. However, it is acknowledged that the stagnation pressure coefficient appears to be larger than unity, signaling numerical error, lack of convergence, a by-product of the RANS-based approach, or a product of viscous effects, as noted in [7], which can likely be discarded at the given Reynolds number.

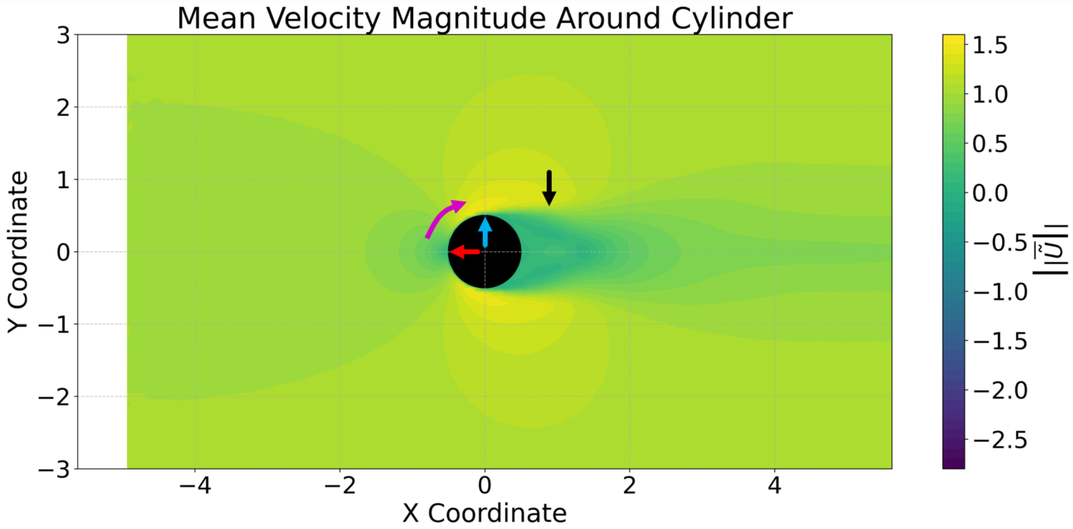


Figure 3.6: Contours of time-averaged velocity magnitude, $Z/D = 0.1$.

Subsequently, the flow acceleration region is reflected by the magenta arrow, induced by the cylinder curvature and translating into the pressure reduction that is shown in the C_p plot. This flow acceleration takes place up to $\theta \approx 75°$ and has associated a certain boundary layer build-up, which can be observed in more detail in Figure 3.4. Furthermore, right after the suction peak, with a value of $C_{p,min} \approx -1.5$ the flow is still attached, seemingly separating at around $\theta = 90°$. This is highlighted by the blue arrow, and can be identified thanks to the sudden change in the slope of the pressure coefficient slope.

Finally, at the back of the cylinder a recirculation region can be found, where the pressure coefficient presents a virtually constant value of $C_p \approx -1$. As pointed out by the black arrow in the above contours, the separation produces a wake downstream of the cylinder surface, creating a shear layer with the free-stream.

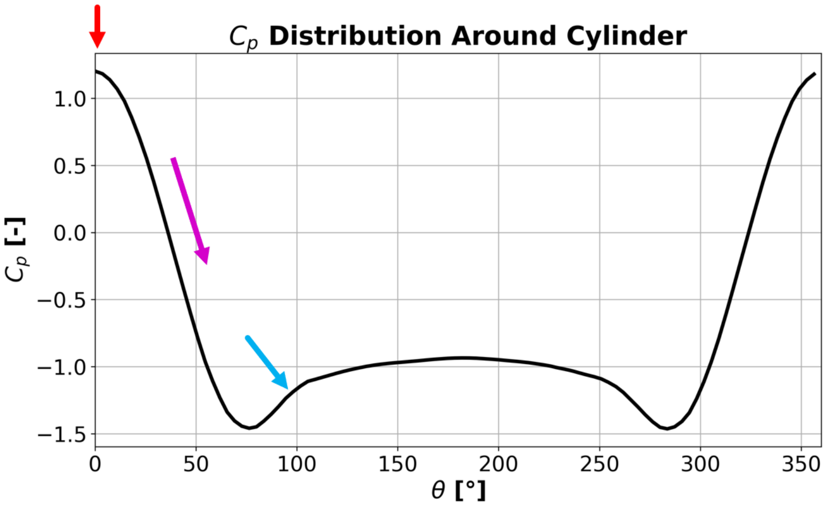


Figure 3.7: C_p distribution on the cylinder surface.

In order to provide further insights, the mean flow streamlines around the cylinder are shown in Figure 3.8, taken at the same plane as the previously discussed contours. In this representation, two coherent, counter-rotating vortex structures can be clearly identified in the wake region behind the cylinder. Additionally, based on the periodic nature of the lift and drag coefficients, it can be inferred that these eddies are generated in an alternating fashion. Accordingly, it may be said that the flow motion involves the roll-up of shear layers from either the top or bottom of the cylinder into coherent vortical structures.

In turn, each vortex is shed downstream, allowing the formation of the next eddy on the opposite side.

Besides, the streamline pattern provides further support to the aforementioned location of flow separation, given that the top and bottom separation streamlines emanate from the $\theta \approx \{90^\circ, 270^\circ\}$ points.

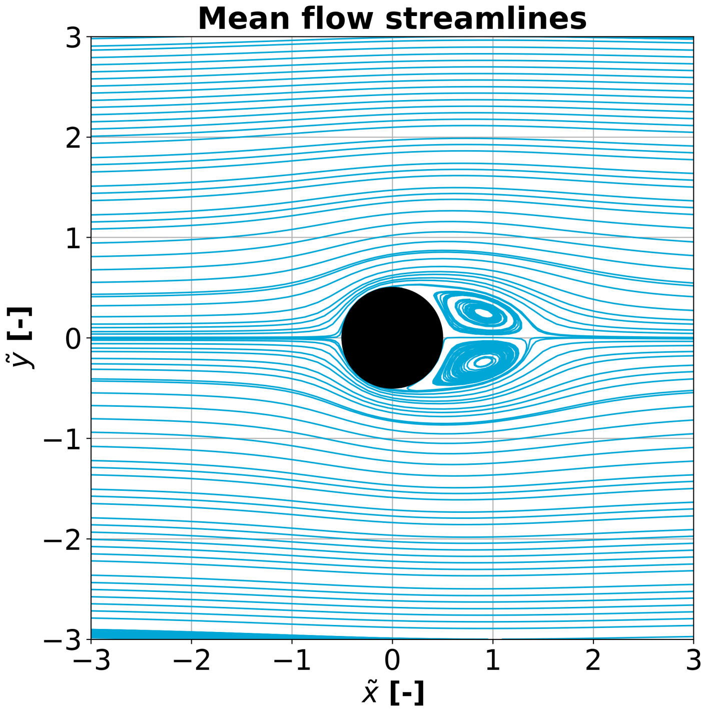


Figure 3.8: Mean flow streamlines on $Z/D = 0.1$.

3.2. Results from PINN baseline

After providing a broad overview of the flow that characterizes the selected regime, the performance of the *PINN* baseline framework that has been covered in Chapter 2 is exposed.

3.2.1. Numerical results

In the first place, it is of interest to analyze the various loss and error metrics of the *PINN* after convergence. To this end, the algorithm is evaluated at all measurement locations, hence providing more representative values than would be expected from its analysis over the training batch size.

On the one hand, Table 3.4 includes the total loss and the values for each of the contributing terms. From the results, it can be said that the data loss component dominates the learning process, followed by the PDE loss, which is about a third of the former. On the contrary, the prior loss and pressure boundary conditions are matched much more closely, meaning that they are barely contribute to the update of network parameters during the final iterations.

Table 3.4: Loss components for *PINN* baseline.

$\mathcal{L}_{\text{Total}}$	$\mathcal{L}_{\text{Data}}$	$\mathcal{L}_{\text{PDE}}$	$\mathcal{L}_{\text{Prior}}$	$\mathcal{L}_{\text{p, BC}}$
1.18×10^{-4}	8.98×10^{-5}	2.79×10^{-5}	4.90×10^{-7}	1.25×10^{-8}

Looking further into the sub-components of each loss term, Table 3.5 features the Mean Squared Error metric for each variable across all measurement points. Paying attention first to all variables except for pressure, as this does not directly contribute to the data loss, it may be said that both velocity components represent the largest contribution and are reconstructed with similar accuracy. On the other hand, all three Reynolds Stress components offer a similar precision with respect to the reference data.

Besides, in line with expectations, the pressure mean squared error for the hidden variable, namely pressure, is the largest but still in the same order of magnitude as the rest of loss contributions. This already reflects the power of the PDE loss, which can enforce precisions comparable to those achieved by a supervised loss term.

Table 3.5: Mean squared error per predicted variable.

$\bar{\tilde{u}}$	$\bar{\tilde{v}}$	$\overline{\widetilde{u'u'}}$	$\overline{\widetilde{u'v'}}$	$\overline{\widetilde{v'v'}}$	$\bar{\tilde{p}}$
3.30×10^{-5}	2.70×10^{-5}	1.02×10^{-5}	1.17×10^{-5}	8.00×10^{-6}	7.00×10^{-5}

Finally, all three residuals appear to display similar contributions to the Partial Differential Equation loss term, with a somewhat larger x-momentum value. While it is not especially remarkable, it is interesting to note that having a 'sink' term such as the x and y pressure gradients on both equations appears to have no major effect on the minimization of the x and y momentum loss terms with respect to the the continuity residual. This is likely due to the fact that the pressure terms undergo a constrained optimization, since the rest of terms in both equations are constrained by the data loss component.

Table 3.6: PDE residuals.

$\mathcal{R}_{\text{continuity}}$	$\mathcal{R}_{\text{x-momentum}}$	$\mathcal{R}_{\text{y-momentum}}$
9.24×10^{-6}	8.57×10^{-5}	1.01×10^{-5}

In order to provide a more comprehensive and visual representation of the above information for the reader, Figure 3.9 next reflects the proportion of each loss component with respect to the total loss, namely the full pie circumference.

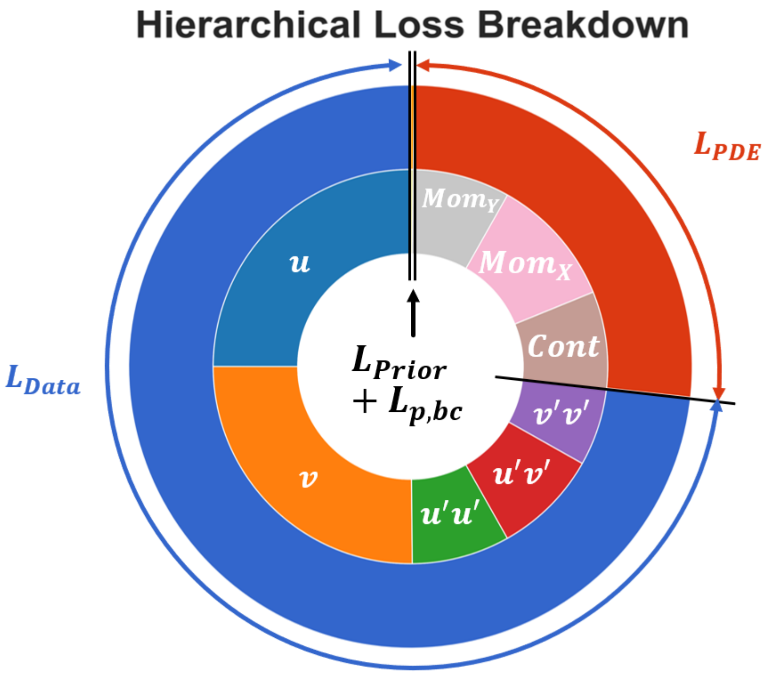


Figure 3.9: Pie chart of loss contributions.

3.2.2. Spatial distribution of errors

Once an idea of how accurate the *PINN* reconstruction is in overall terms, as well as how the error is distributed across the various variables, it is of interest to further investigate how such errors are distributed in space, paying special attention to the cylinder proximity to better understand surface pressure reconstruction accuracy.

Firstly, Figure 3.10 includes contours of the reference (left), *PINN*-predicted (right) and relative delta in dimensionless x-velocity, limited to the cylinder proximity.

Even though the comparison between the reference and predicted fields does not reflect major noticeable differences, the relative delta plot provides some more insight into which areas show the largest deviations. Note that the delta plot is calculated such that the magnitude shown is the difference between the reference and the prediction as a percentage of the free-stream magnitude. The expressions of the error metric for each variable are included in Equation 3.6. Rather than reflecting the Squared Error, which is used during training, this metric is selected to aid in the interpretation of *PINN* performance.

$$\varepsilon_{\overline{\tilde{u_i}}} = \frac{\overline{\tilde{u_i}}_{PINN} - \overline{\tilde{u_i}}_{Ref}}{U_\infty} \times 100; \qquad \varepsilon_{\overline{\tilde{p}}} = \frac{\overline{\tilde{p}}_{PINN} - \overline{\tilde{p}}_{Ref}}{\rho_\infty U_\infty^2} \times 100; \qquad \varepsilon_{\overline{\widetilde{u_i' u_j'}}} = \frac{\overline{\widetilde{u_i' u_j'}}_{PINN} - \overline{\widetilde{u_i' u_j'}}_{Ref}}{U_\infty^2} \times 100; \tag{3.6}$$

Focusing on the delta, it is possible to see that the freestream is largely error-free, with non-zero deviations being limited to the cylinder proximity and wake areas. In particular, it appears that the largest error accumulation happens around the recirculation region and the edge of the wake, which is to be expected given the higher concentration of gradients. The deviations present maximums of up to $\varepsilon = \pm 10\%$, but mostly limited to $\varepsilon \leq \pm 5\%$. Furthermore, it may be stated that the error pattern looks to be approximately symmetrical around the $y/D = 0$ line, which could signal a physical explanation for the deviation rather than a by-product of the stochasticity in the *PINN* training process. Focusing on the delta around the wake edge, the alternating positive and negative pattern signals a smoother wake edge, resulting in a $\sim 0.3D$ wider wake predicted by the *PINN*.

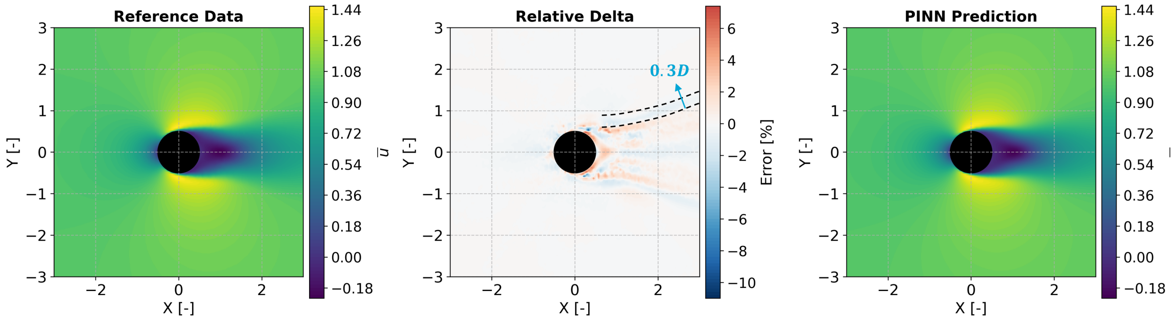


Figure 3.10: Contours of dimensionless x-velocity - results from *PINN* baseline.

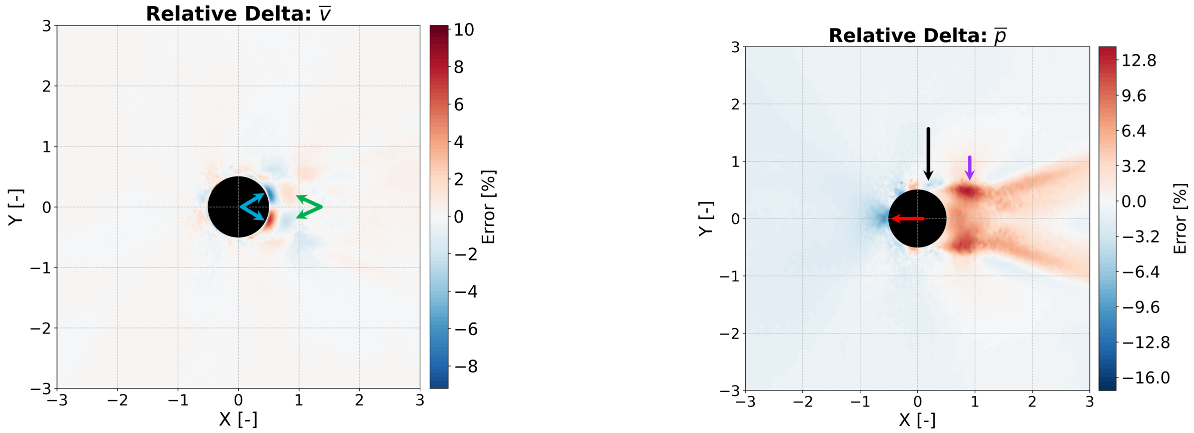
Before delving into further exploration of the error source, it is worth representing the delta magnitudes for the rest of the variables to form a more complete judgment on the overall performance of the baseline algorithm. Accordingly, Figure 3.11 includes the relative delta contours for the dimensionless y-velocity (Fig. 3.11a) and pressure (Figure 3.11b).

Placing the spotlight on the former, the recirculation region contains the principal deviation with respect to the reference data, with maximum errors of $\varepsilon \approx \pm 10\%$. As pointed out by the blue and green arrows and attending to the direction of rotation of the main vorticity in Figure 3.8, the observed difference in vertical velocity may be interpreted as a reduction of vortex-induced velocity and thus a possible loss of primary vortex strength in the *PINN* version.

Further support to this hypothesis is given by the results depicted in the pressure delta plot, where the purple arrow signals a larger local static pressure around the region where the cores of the primary eddies are located, explaining their reduced strength. Additionally, the higher wake pressurization matches the observations made earlier on the wake width, provided that this promotes the exchange of momentum with the mean flow.

Additional deviations in the pressure pattern are identified by the black and red arrows. With respect to the first one, the pointed region of additional suction can indicate a latter separation in the *PINN* solution, which can aid in the explanation of the above differences. Finally, the red arrow indicates a substantial deviation in the stagnation region, where the largest absolute error with respect to the reference pressure is located.

Overall, pressure errors are contained in $\varepsilon \in [-15, 12]$ %, which is a comparable level of reconstruction accuracy as that of the variables where labeled data exists, putting forward the advantage of the PDE loss.

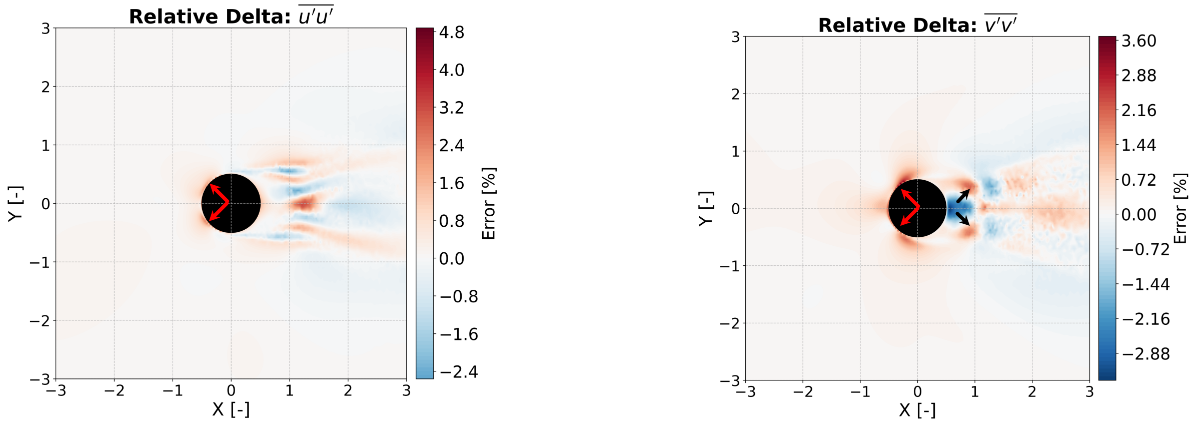


(a) Mean dimensionless vertical velocity - $\overline{v}$.

(b) Mean dimensionless pressure - $\overline{p}$.

Figure 3.11: Contours of relative delta: *PINN* - Reference.

Proceeding with the analysis, Figure 3.12 includes the relative delta contours for both normal Reynolds Stress components. In this case, reconstruction errors take values contained within $\varepsilon \in [-5, +5]\,\%$ of U_∞^2. In this case, pronounced deviations are observed close to the cylinder surface around the start of the shear layer, with over-predictions from the *PINN* solution on both cases. Since these two components form the mean turbulent kinetic energy, this can be interpreted as a $\sim 4-5\%$ increase in early turbulence production, which could explain the minimal delay in separation.



(a) $\overline{\widetilde{u'u'}}$.

(b) $\overline{\widetilde{v'v'}}$.

Figure 3.12: Contours of absolute delta - normal Reynolds Stress Components.

Finally, the Reynolds Shear stress reference data, *PINN* prediction and the relative difference between the two is attached in Figure 3.13. In this case, maximum errors are bounded to $|\varepsilon| \leq 3\%$ of U_∞^2, displaying a symmetric pattern around $y/D = 0$. The most remarkable variation, as pointed out by the black arrows, lies on the shear layer that forms the wake edge. Particularly, the *PINN* solution predicts a stronger exchange of momentum between the streamwise and transversal velocity fluctuations, thus enhancing turbulent mixing and potentially affecting the width of the wake downstream, as commented above.

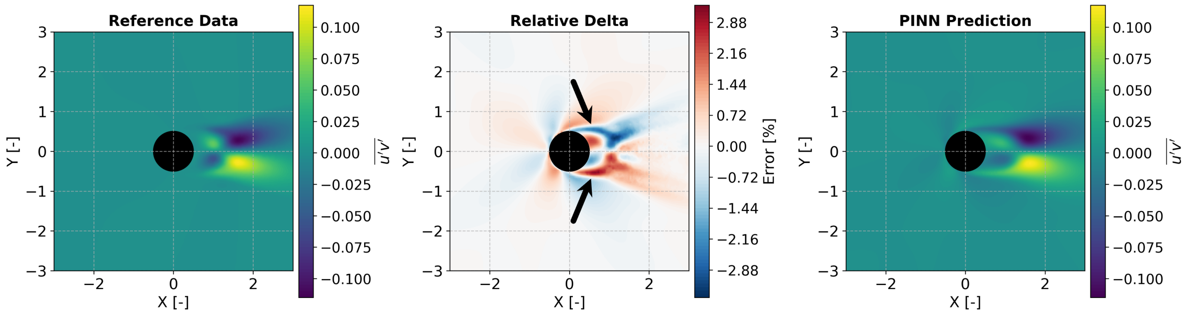


Figure 3.13: Contours of dimensionless turbulent shear stress - results from *PINN* baseline.

Finally, the reference and *PINN*-predicted pressure coefficient distributions on the cylinder surface are represented in Figure 3.14, as well as the difference between the two. Firstly, it can be argued that the overall reconstruction of the pressure is physically sensible. Furthermore, looking at the differences with respect to the reference data, the same observations which were hinted at the pressure contours are materialized here. From larger to smaller deviations, the stagnation pressure coefficient value predicted by the *PINN* at stagnation, namely $\theta = 0°$, is of $C_p = 1$, which is more in line with expectations than the reference data. Besides, smaller deviations of $\Delta C_p \leq 0.1$ are found right before and at the suction peak, with the *PINN* resulting in an overall smoother C_p distribution.

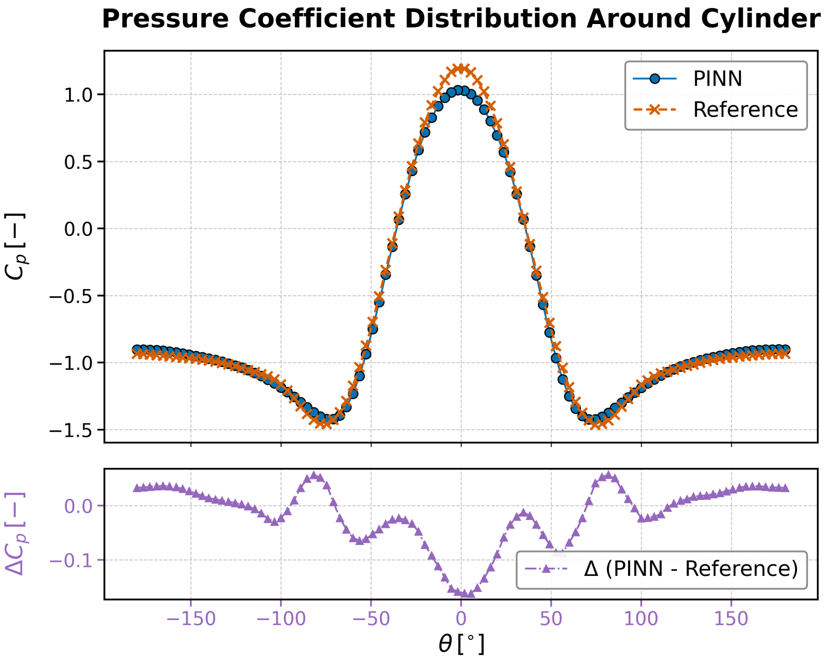


Figure 3.14: C_p distribution on the cylinder surface - *PINN* vs Reference data.

3.2.3. Investigation on the sources of reconstruction error

In order to better understand whether the observed differences have a physical foundation or rather are a lack of reconstruction ability of the *PINN*, a method is here developed to further investigate on the source of the deviations above exposed.

The proposed method consists of training an identical NN architecture solely with the data loss term, disregarding the PDE and other loss components. In this scenario, since pressure is learned directly from the data, the full state is required in advance and thus would be impractical in a PIV set-up. The main purpose of this method is to benefit from the ability of the NN framework to provide derivative calculations locally and independently of the discretization scheme, thus allowing to compute the corresponding residuals for all three conservation equations in an accurate manner.

It must not be neglected, though, that this method provides only an approximation of the reference data, since the NN still acts as a general function approximator for all flow variables. However, this is equally desirable, as it allows to gauge the extent to which the NN architecture and selection of training parameters allow to reconstruct the flow solution. In fact, such results are included in Table 3.7, where the *PINN* baseline results are included for reference.

Table 3.7: Mean squared error per predicted variable - $\mathcal{L}_{data}$ only vs *PINN*.

	$\bar{\tilde{\boldsymbol{u}}}$	$\bar{\tilde{\boldsymbol{v}}}$	$\overline{\widetilde{\boldsymbol{u'u'}}}$	$\overline{\widetilde{\boldsymbol{u'v'}}}$	$\overline{\widetilde{\boldsymbol{v'v'}}}$	$\bar{\tilde{\boldsymbol{p}}}$
PINN BSL	3.30×10^{-5}	2.70×10^{-5}	1.02×10^{-5}	1.17×10^{-5}	8.00×10^{-6}	7.00×10^{-5}
$\mathcal{L}_{\mathbf{data}}$ $Only$	9.69×10^{-6}	3.57×10^{-6}	1.41×10^{-6}	6.59×10^{-7}	1.48×10^{-6}	2.94×10^{-6}

By comparison of the above numbers, it is deduced that the solution space allowed by the NN framework makes it possible to increase the reconstruction accuracy beyond the baseline results. This is concluded since the mean squared error for most states is reduced by approximately an order of magnitude, signaling margin for further accuracy.

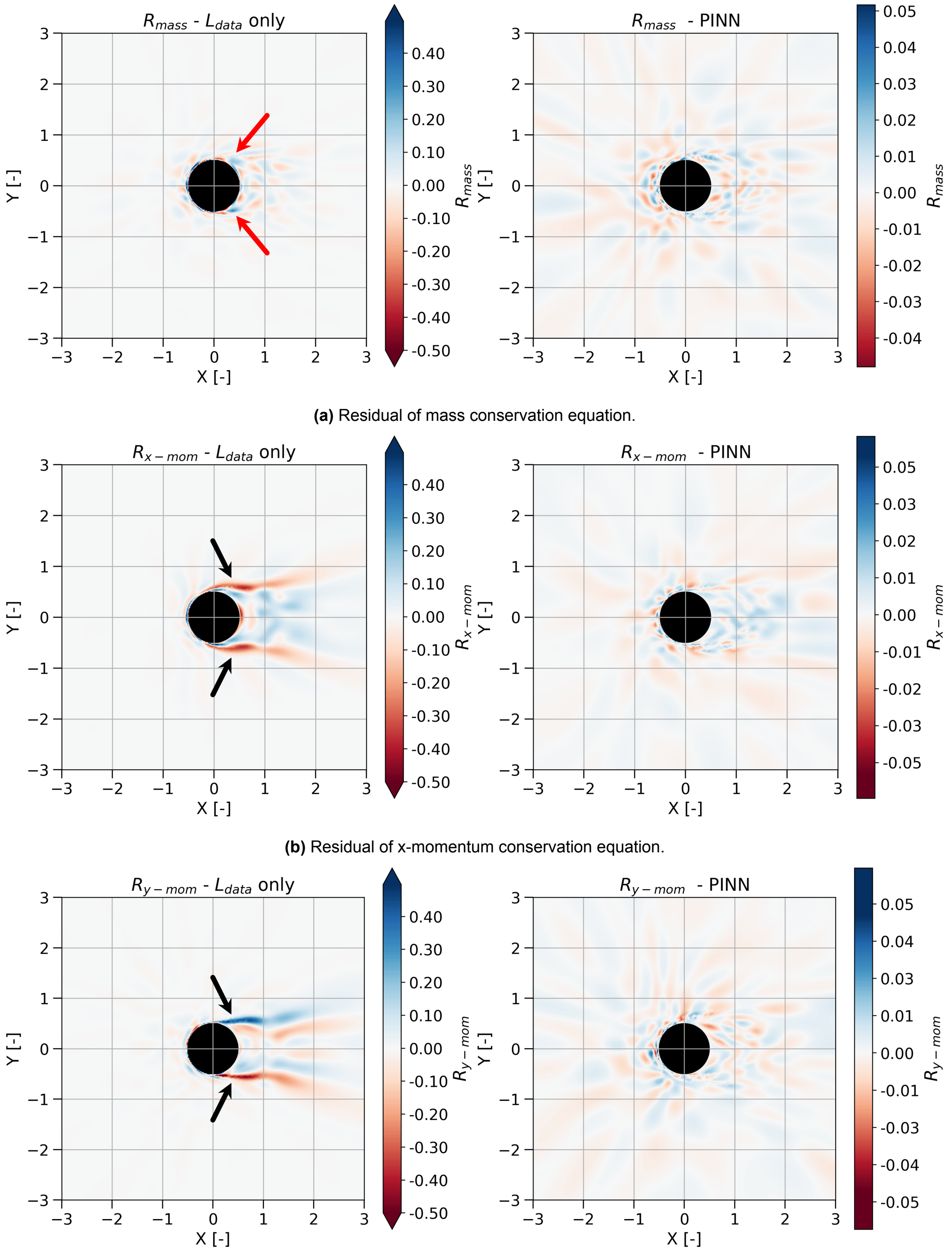
Furthermore, since the rest of loss contributions other than the PDE, namely $\mathcal{L}_{prior}$ and $\mathcal{L}_{p,bc}$ are indeed fulfilled by the reference data, it is concluded that the root cause for the observed differences is the PDE loss. With the aim to investigate further into this, contours of all three residuals are represented for both the $\mathcal{L}_{data}$ only (left) and the *PINN* baseline (right) in Figure 3.15. Before elaborating on the results, it is here noted that the color scale required by the $\mathcal{L}_{data}$ only version is ten times that required by the *PINN* baseline, already exposing the lack of converged residuals for the reference data.

Firstly, regarding the residual for mass conservation, the most relevant features in Figure 3.15a appear right after separation, where, by virtue of the onset of turbulence, three-dimensional motion appears to cause a positive residual in the reference data, as pointed out by the red arrows. On the contrary, a seemingly random pattern dominates the entire *PINN*-solution domain. Additionally, hints of a positive R_{mass} are observed around the stagnation region, but limited to the cylinder proximity.

On the contrary, a very different picture is laid out in the x and y momentum conservation residuals, attached in Figures 3.15b and 3.15c respectively. In both cases, the model that has been trained on data alone displays residual values of up to $R \approx 0.8$ around the edge of the wake, with especial emphasis on the edge of the shear layer. On the contrary, the *PINN* baseline results again reflect a chaotic distribution of residuals of at least an order of magnitude below.

It is noteworthy that the wake edge region, which is highlighted by black arrows, has been the center of discussion in the previous subsection as the main source of deviations between the prediction and labeled data. Accordingly, it can be firmly stated that these findings support the hypothesis that the differences with respect to the reference data are explained by an enhanced fulfillment of the governing equations by the *PINN*, rather than a limited reconstruction ability of the framework.

Finally, it is also noticed that the stagnation region contains a non-zero x-momentum residual close to the solid boundary, which is the main source of observed pressure reconstruction error in the C_p surface plot of Figure 3.14, further strengthening the above idea.



(a) Residual of mass conservation equation.

(b) Residual of x-momentum conservation equation.

(c) Residual of y-momentum conservation equation.

Figure 3.15: Contours of residuals: *PINN* vs $\mathcal{L}_{data}$ only.

Overall, it is concluded that the exposed *PINN* framework, when applied to the mean velocity data of the flow around a cylinder obtained via CFD, is able not only to reconstruct pressure with a similar accuracy to the fields with labeled data but it also proposes solutions constrained by the conservation equations, potentially improving the physical validity of the solution. Furthermore, these qualities are not solely limited to off-body regions, but also apply to the ability to reconstruct surface pressure.

3.3. Sensitivity studies

After an initial exposition on the ability of *PINNs* to reconstruct off-body and surface pressure, it is of interest to explore how the various parameters that characterize the framework affect the solution.

Accordingly, some of the parameters like the neural network shape or the distribution and number of collocation points are here studied not only to gauge their effect on the absolute accuracy of the *PINN*, but also to comprehend how these may be tuned to achieve similar results in a more efficient manner.

However, prior to exposing the results, it is worth showing the expected variation that a single *PINN* framework with no modifications can provide in terms of the most relevant loss terms. This can be visualized in the scatter plot of Figure 3.16, where the mean and standard deviation of each loss component are shown for five different runs with the baseline *PINN* architecture and training parameters.

From the plot, it is deduced that the PDE loss generally takes lower values than the data counterpart, with deviations of $\sigma_{PDE} = 3.4 \times 10^{-5}$ around a mean of $\mu_{PDE} = 6.8 \times 10^{-5}$. In the case of the data loss, the fluctuations are of $\sigma_{data} = 5.9 \times 10^{-5}$ around a mean of $\mu_{data} = 1.6 \times 10^{-4}$, hence making it more stable in relative terms. Besides, the prior loss deviation lies within $\sigma_{prior} < 1 \times 10^{-6}$, barely contributing to the overall variance. Finally, the addition of these deviations add up to $\sigma_{tot} = 8.5 \times 10^{-5}$ for the total loss.

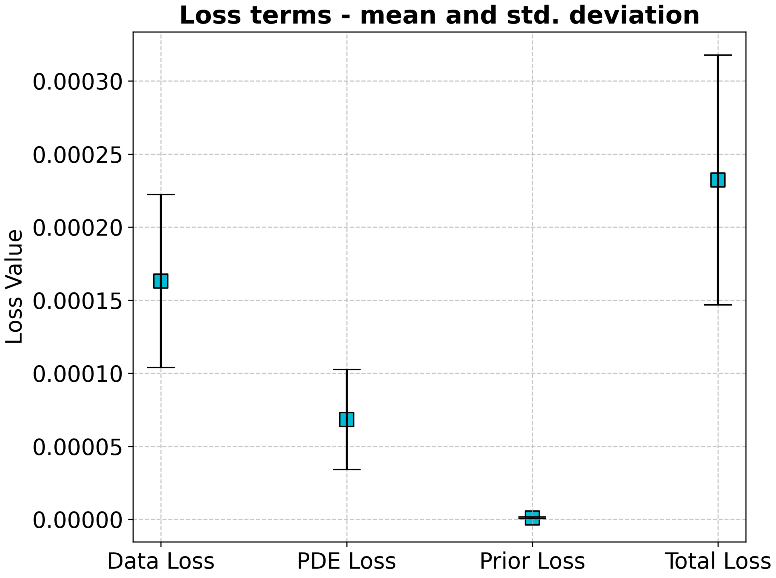


Figure 3.16: Mean and variance for each loss term - data from five baseline *PINN* runs.

With this variance in mind, which is mostly due to the randomness in the weight initialization scheme and stochasticity of the training process, the results presented in the forthcoming sections become more interpretable, as it allows to discern significant trends over variations that are expected due to the randomness inherent to the algorithm.

3.3.1. NN shape

Introduction

As a first investigation, the effect of the shape of the neural network is studied. To do so, various architectures of different widths, or number of neurons per layer, and depths, or number of layers, are analyzed with respect to the key performance indicators, such as the various loss components.

It is noted that, by modifying these variables, the total number of learnable parameters in the network is altered. In turn, this changes the complexity of the set of allowed solutions, as well as that of the optimization space. Accordingly, the intent is not only to understand how the complexity of the allowed solution space changes with size, but also to provide guidance on minimum requirements for both dimensions. Presumably, these findings will likely only be applicable to the present problem or, at best, to similar ones. The reason behind is that the complexity of the solution, while somewhat vague, includes factors as the number of variables, spatial gradients, data sample size and ultimately depends on how efficiently the NN stores information.

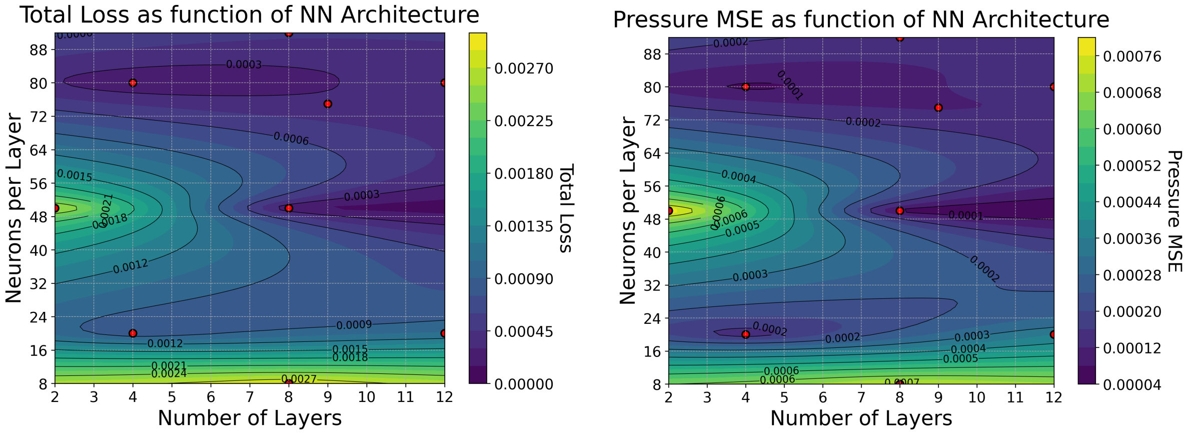
In order to conduct this study, a one-factor-at-a-time (OFAT) approach has been used to design a sensitivity study that allows to properly differentiate the effects of both variables on the results and performance metrics. To do so, four levels have been determined for each parameter based on Central Composite Design for $k = 2$ factors, resulting in the test matrix in Table 3.8. As reported, the number of layers, L, changes from an extremely low value of 2, up to a model that is 12 layers deep. On the other hand, the number of neurons per layer, N, sweeps from 8 until 92, covering a wide range of model widths. Finally, the last column features the total number of trainable parameters, which are obtained via Equation 2.3, taking into account that the number of input and output neurons are kept constant at $N_{in} = 2$ and $N_{out} = 6$.

Table 3.8: NN Shape/Size study.

Test #	L		N		N_{params}
	Level	**Value**	**Level**	**Value**	
1	-1	4	-1	20	1866
2	-1	4	1	80	26 646
3	1	12	-1	20	5226
4	1	12	1	80	78 486
5	0	8	0	50	20 856
7	-1.4	2	0	50	5556
8	0	8	1.4	92	69 282
9	0	8	-1.4	8	654

Results

In order to analyze the performance of the various model architectures, the contours in Figure 3.17 are firstly referred to. To the left, the total loss variation with the number of layers and neurons is shown. In red, scattered locations reflect the tested architectures and the interpolation and extrapolation of information is carried out via linear radial basis functions. To the right, an analogous interpretation can be made, but the variable represented by the contour colors is the mean-squared error of pressure, the main key performance indicator in the present thesis.



(a) Total loss.

(b) PDE loss.

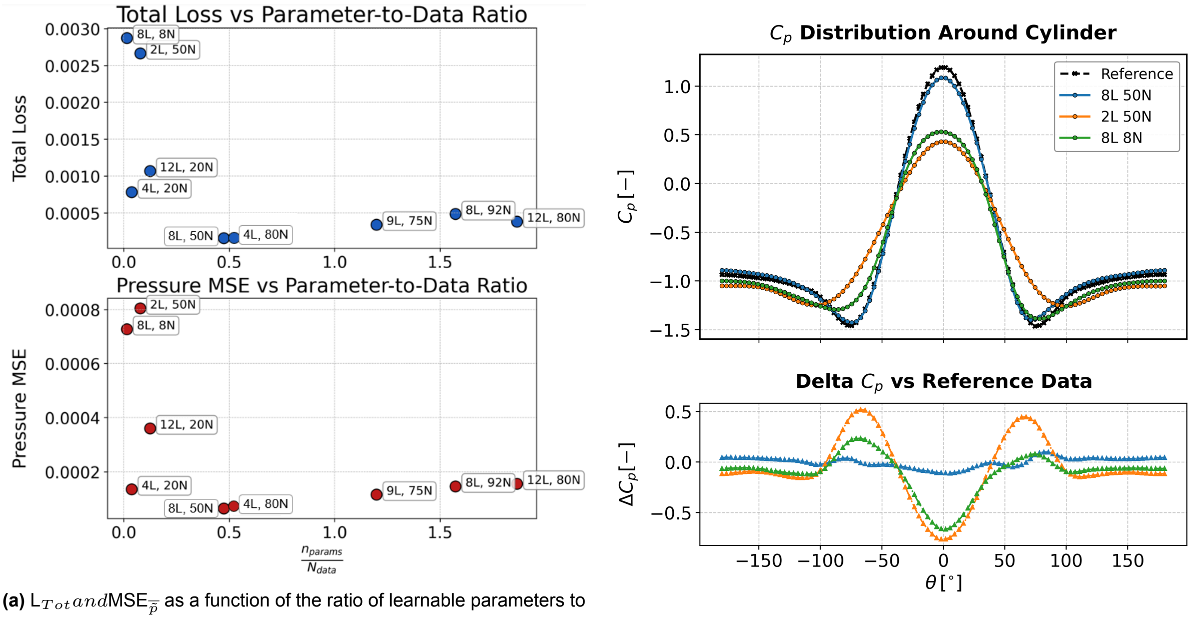
Figure 3.17: Contours of different KPIs as a function of the number of layers (L) and neurons (N).

Delving into the test results, observation of both Figures leads to the conclusion that these present very similar overall trends, thus reflecting the validity of the total loss optimization as a means to reconstruct the unknown pressure field. From the contours, it may be acknowledged that the low extremes for either the amount of neurons or of layers lead to the least accurate models, with MSE_p values as high as 8.0×10^{-4} and 7.3×10^{-4} for the shallowest and narrowest architectures respectively. Interestingly, displacement from such low dimensions seems to provide a sizable step in performance, achieving an $MSE_p = 1.3 \times 10^{-4}$ even for a reduced size model as the $4L, 20N$.

Overall, as the model size is increased it appears that the reconstruction error decreases. However, drawing the attention to the bigger picture shows a slight trend of worse accuracy for very deep or wide models. This can be drawn from comparison of the MSE_p values for the $L = 8, N = 50$ model with the $L = 8, N = 92$ architecture or contrasting the results for the $N = 20, L = 4$ NN with those from the $N = 20, L = 12$.

Provided these observations, it becomes apparent that the trend is not necessarily as straightforward as a direct proportionality between model size and reconstruction error. Accordingly, further insight is provided into the underlying trend via the graph in Figure 3.18a. Specifically, the two subplots show the evolution of the two same key performance indicators, namely $\mathcal{L}_{Tot}$ and MSE_p, as a function of the ratio of trainable parameters to the amount of data points used during training. Starting from the left, where the simpler models are represented, it appears that the absolute neuron and layer sizes gain relevance to the detriment of the ratio N_{params}/N_{data}. This is because the two extreme models produce more than twice the pressure reconstruction error than the models with a similar number of parameters but a more balanced share between layers and neurons.

Furthermore, as the number of trainable parameters grows to one half of the data points, the accuracy of the NN models seems to find a consistent optimum, as deduced from the two similar results provided by the $L = 8, N = 50$ and the $L = 4, N = 80$ architectures. As the number of tunable parameters is increased further still, the optimization appears to struggle to converge to the same optimum, potentially depicting an overly complex optimization landscape.



(a) $\mathrm{L}_{Tot and}\mathrm{MSE}_{\overline{p}}$ as a function of the ratio of learnable parameters to data points.

(b) Pressure coefficient distribution.

Figure 3.18: Effect of NN shape on pressure reconstruction error.

Finally, an illustration of the modes of error for the simplest models is given in the C_p distribution plot of Figure 3.18b. Review of both the absolute and delta plots offers an interesting insight into the preferential modes of failure when the model complexity is not sufficient. In particular, regions that concentrate the most significant spatial gradients, such as the stagnation point and the two suction peaks, seem to accumulate most of the error, contrary to a virtually constant error distribution in the case of the $L = 8, N = 50$ model.

While this study does not allow to define an universal law of appropriate NN sizes for every *PINN* application, key findings are made based on this specific problem:

- for equal amounts of trainable parameters, NN shapes with extremely low dimensions, e.g. $L = 2$ or $N = 8$, seem to incur the largest errors.
- the ratio of trainable parameters to the amount of data points seems to be a more representative metric than the NN shape to gauge the effect of the NN size.
- model complexity does not inherently aid reconstruction accuracy.
- regions of high gradients appear to be more susceptible to error from overly simplistic architectures.

With this in mind, certain guidelines can be provided on a general basis:

- start with a model architecture that aims for about one half of the total number of data points.
- ensure that model sizes lay above a certain threshold, avoiding too shallow ($L \geq 4$) or thin ($N \geq 20$) NNs.

3.3.2. Data loss weighting

Once the effect of the neural network geometry on the solution has been exposed, it is of interest to study how the definition of the key performance indicator used during training, that is the total loss, affects the reconstruction error for each variable and, ultimately, how it affects pressure reconstruction at the cylinder surface.

Introduction

To achieve this, the data loss term is weighted as per Equation 3.7, which modifies the relative contributions of $\mathcal{L}_{data}$ and $\mathcal{L}_{PDE}$ to the total loss during training. According to the definition of the weighting factor given, it is deduced that models trained with $\alpha \in [0, 1[$ will be more inclined to enforcing the Navier-Stokes loss, whereas those using $\alpha > 1$, will favor the learning of the labeled data variables.

$$\mathcal{L}_{Total} = \alpha \cdot \mathcal{L}_{data} + \mathcal{L}_{PDE} + \mathcal{L}_{Prior} + \mathcal{L}_{p,bc} \tag{3.7}$$

In particular, the values for alpha that are included in the present study are:

$$\alpha = \{0.01, \quad 0.1, \quad 1, \quad 10, \quad 100, \quad 1000\} \tag{3.8}$$

In addition to the variation in the bias of the NN to enforce the labeled data or the PDE equations in the final solution, it is expected that the training dynamics are affected too. The reason behind this is that the total loss is inherently dependent on the weighting factor and hence the value of α directly impacts the size of the gradients of the total loss with respect to the training parameters, which are used during the back-propagation step. To aid with understanding this phenomenon, a very simplified example is here included.

Consider a single-input-single-output neural network with one hidden layer and one neuron. Using this architecture with a mean-squared error loss can be considered as performing a linear regression on the set of input points such that the MSE is minimized.

Now, taking two different weighting factors, $\alpha_1 = 0.01$ and $\alpha_2 = 100$, one can compute the corresponding loss values for a prediction of $y_{pred} = \{1.2, 6.0, 10.5\}$ where the ground truth is $y_{ref} = \{0.5, 7.1, 12.1\}$ as per Equations 3.9 and 3.10, respectively.

$$L_1 = \alpha_1 \sum_{i=1}^{N} \left(y_{i,pred} - y_{i,ref}\right)^2 = 0.01 \left\{(1.2 - 0.5)^2 + (6.0 - 7.1)^2 + (10.5 - 12.1)^2\right\} = 0.0426 \tag{3.9}$$

$$L_2 = \alpha_2 \sum_{i=1}^{N} (y_{i,pred} - y_{i,ref})^2 = 0.01 \left\{ (1.2 - 0.5)^2 + (6.0 - 7.1)^2 + (10.5 - 12.1)^2 \right\} = 426 \tag{3.10}$$

As observed, while the real deviation is the exact same, the two loss terms differ by four orders of magnitude. Accordingly, computing $\nabla_\theta L$ for both versions will yield disparate results, which could add or subtract relevance to the problems of vanishing and exploding gradients for deep networks.

Additionally, the decision is made to consider the dynamic weighting scheme proposed in [31] as well. As exposed in §1.4.1, this method has the main reported advantage of dynamically adjusting the weights of each loss term to allow for an evenly distributed contribution during training. In particular, the formulation here implemented is as per Equation 3.11, where the PDE loss is scaled with the dynamic weight, β_{dyn}

$$\mathcal{L}_{Total} = \mathcal{L}_{data} + \beta_{dyn} \cdot \mathcal{L}_{PDE} + \mathcal{L}_{Prior} + \mathcal{L}_{p,bc} \tag{3.11}$$

Additionally, β_{dyn} takes the definition given in Equation 3.12, where the coefficient at each iteration is a blend between the weight at the previous iteration, β_{dyn}^{n-1} and the estimate for the current iteration, $\hat{\beta}_{dyn}^{n}$. Such a combination, controlled by the blending factor, λ, ensures that the weighting coefficient evolves smoothly, which is especially relevant given the stochasticity inherent to the mini-batch approach of the Adam optimizer. In the present case, λ is fixed to 0.1, following the recommendations given in the presentation paper.

Additionally, the estimated dynamic weight for the current iteration, $\hat{\beta}_{dyn}^{n}$, is presented as the quotient of the norms for the gradients of both loss terms with respect to the NN parameters. Intuitively, this ratio ensures that the contributions of both terms are balanced during the back-propagation step.

$$\beta_{dyn}^{n} = (1 - \lambda) \cdot \beta_{dyn}^{n-1} + \lambda \cdot \hat{\beta}_{dyn}^{n}; \qquad \hat{\beta}_{dyn}^{n} = \frac{\overline{|\nabla_\theta \mathcal{L}_{PDE}|}}{|\nabla_\theta \mathcal{L}_{data}|} \tag{3.12}$$

Finally, it is noted that an initial buffer of 500 iterations or ~ 120 epochs is given with no dynamic adjusting of the weights, keeping $\beta_{dyn} = 1$, as it would happen in an unweighted training process.

Results

Prior to exposing and commenting on the results, it is noted that the performance metrics shown are evaluated on the entire set of available measurement points, rather than just taking the value at the last training batch. With this in mind, the effect of the data loss weighting factor on the various loss terms are included in Figure 3.19.

First of all, it is direct to see that the PDE loss, in solid red, follows a quasi-linear positive relation to α, with lower values aiding to minimize the N-S residuals. In a similar but opposite way, the data loss, in blue, follows an inverse relation to the weighting coefficient with a saturation above $\alpha = 100$. Interestingly, this saturation value of $\mathcal{L}_{data}$ is only twice as large as that of the NN model when trained only on the data loss of all known states (namely, all but pressure). This is insightful, as it means that it likely represents a learning limit for the given combination of architecture, loss definition and optimizer. It can also be said that the behavior witnessed for both the data and PDE losses is expected given the definition of the loss weighting parameter.

Accordingly, more insightful are the total loss results, depicted by the black and light green lines. Respectively, these reflect the weighted total loss, as defined in Equation 3.7, and the unweighted counterpart, which would be a realistic metric for how well the NN optimizes the baseline problem. On the one hand, the weighted loss line finds a very similar trend to the PDE loss with respect to α, as it is mostly dominated by the weighted component of the data loss. However, it is possible to see that the two lines are not perfectly parallel, with the two being the closest when $\alpha = \{1, 10\}$. Intuitively, this coincides with the minimum absolute unweighted total loss, which happens at $\alpha = 1$. Additionally, given the difference in the slope of total loss with α to either side of the optimum, it can be stated that the sensitivity is larger for $\alpha < 1$ than for $\alpha > 1$.

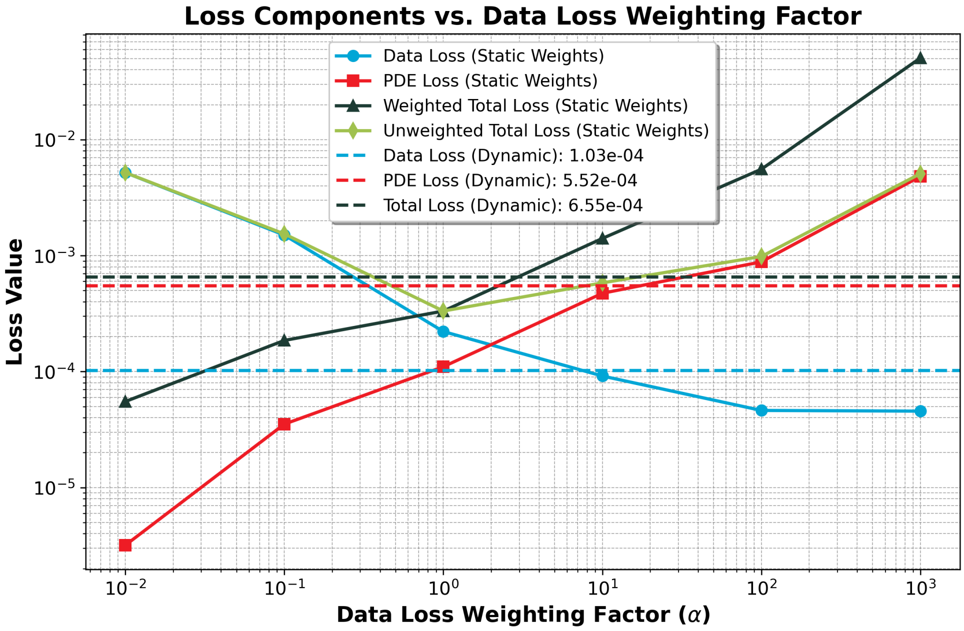


Figure 3.19: Dependence of loss components on data loss weighting coefficient.

In order to better understand how the data loss weights affect the region of focus in the present thesis, the pressure coefficient distribution on the cylinder surface is shown in Figure 3.20 for all weights except for $\alpha = 10^3$. Note that the reference data has been represented too with the dash and cross line. Additionally, the difference of each version with respect to the CFD output has been included in the second and third subplots.

First and foremost, the two versions with $\alpha < 1$ stand out due to the large deviation with respect to the ground truth data. Visibly, it appears that the lack of agreement with the reference dataset due to the small weighting only achieves a smoothed out version of the correct C_p distribution. Specifically, the version with the smallest weighting factor, $\alpha = 0.01$ in blue, differs from the reference data in values of up to $\Delta C_p \approx 1.3$ at stagnation and $\Delta C_p \approx 1$ in the suction peak. Increasing α by an order of magnitude offers a step improvement in accuracy, as reflected by the orange line, still containing errors of up to $\Delta C_p = 0.7$ at stagnation.

On the other hand, the version with $\alpha = 100$ in red shows a virtually perfect match to the reference data, with maximum ΔC_p values of 0.09. When compared to the baseline case, namely $\alpha = 1$, it appears that the biggest difference lies in the stagnation region, where the higher weight brings the $\alpha = 100$ solution closer to the reference data. This is indeed interesting, as pressure is obviously not being fed to either version. However, it appears that a larger data weight, which, as has been exposed, reduces the extent to which the N-S equations are enforced, brings the solution closer to the CFD data. This is in line with the observations made in §3.2.3, where it was hypothesized that the $C_p = 1$ at stagnation predicted by the *PINN* was due to the PDE regularization.

Having exposed the observations on the static loss weighting factors, it is of interest to dive into the performance of the dynamic weighting scheme. Firstly, the loss terms corresponding to this version are included in Figure 3.19 as dashed horizontal lines. Analyzing these results, it appears that the model performs overall very similarly to the $\alpha = 10$ model, with final values of $\mathcal{L}_{PDE} = 5.5 \times 10^{-4}$ and $\mathcal{L}_{data} = 1.0 \times 10^{-4}$. Interestingly, if one looks at the progression of β_{dyn} shown in Figure 3.21b, it is possible to deduce that there is an overall equivalence in weighting the data and the PDE loss terms. This is because the dynamic model, which is weighting the PDE loss by around $\beta = 1/10$, shows very similar performance metrics to the $\alpha = 10$ model. Besides, in the same Figure 3.21b, it is possible to verify that the dynamic weight definition equation is indeed representative of the inverse ratio of loss terms, making it a suitable balancing term.

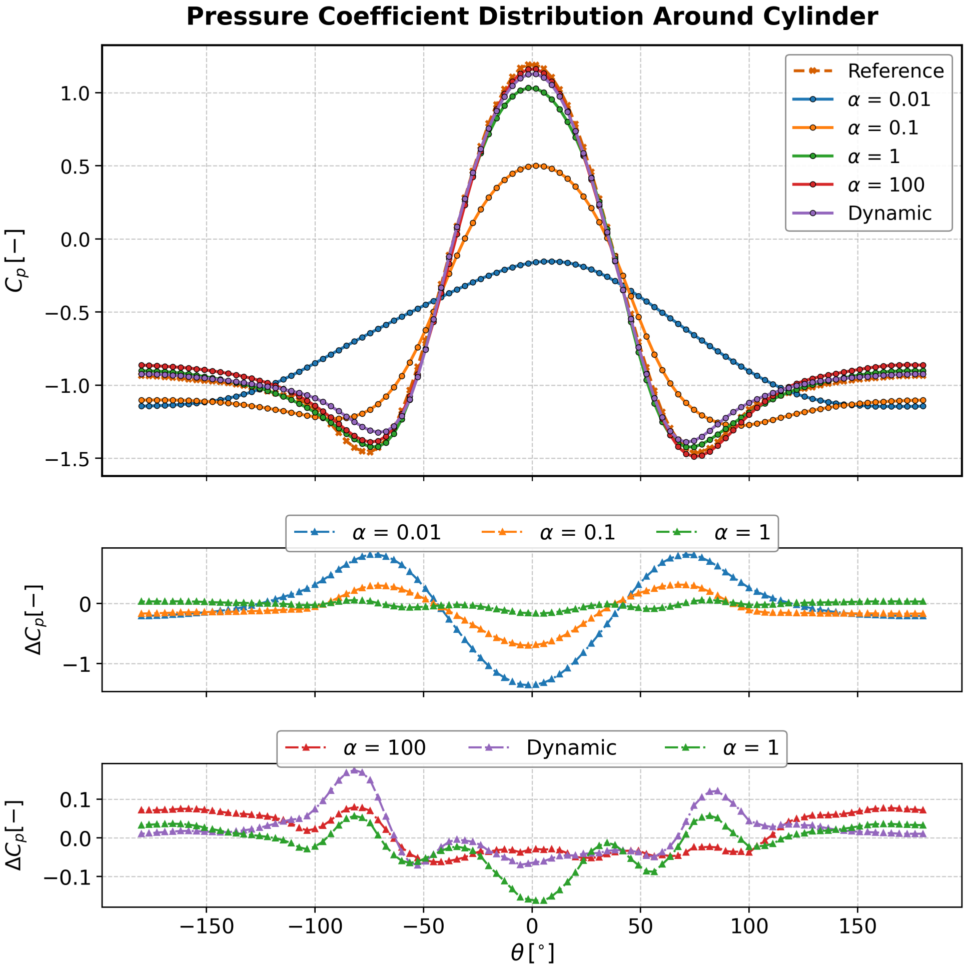
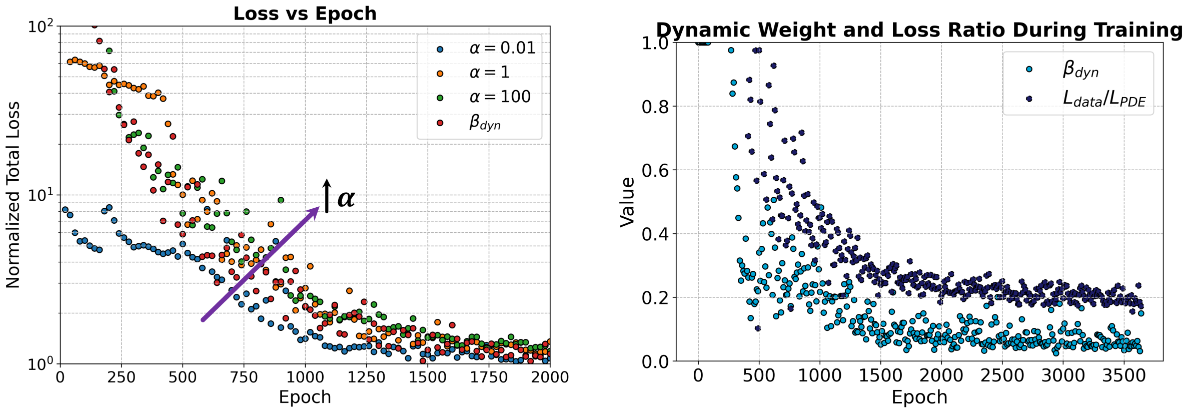


Figure 3.20: C_p distribution and comparison with reference data for various data weighting approaches.

Finally, as it was previously hinted, there is a subtle but consistent effect of the weighting factor on training dynamics. This is reflected in Figure 3.21a, where the total loss for each run has been normalized with the mean of their final value to provide a representative comparison of the learning dynamics. As highlighted by the purple arrow, there is a slight dependence to the loss weight in the training progression, such that lower values of α show a smoother yet quicker convergence to their final, asymptotic value. The interpretation, which can also be supported by the magnitude of the oscillations between epochs, is that the loss weight initially acts as an up or down-scaling factor for the weight updates, just as a larger or smaller learning rate would. However, as the training process advances and the asymptotic region of the learning curve is reached with a small learning rate, the effect of the loss weight is limited to the proportion of loss metrics shown above.

Overall, it can be concluded that the loss weighting factor for this problem has its optimum at the baseline value of $\alpha = 1$, where the solution appears to find an appropriate balance between enforcing a physically sensible solution and learning the labeled data. Along these lines, it has been proven that a too small value of α makes the NN unable to converge to the right pressure fields, showing the relevance of the reference data. Similarly, too large a weighting factor appears to subtract excessive relevance from the PDE loss, causing certain areas such as the stagnation region to converge to solutions that provide higher local residuals. Despite this, it becomes obvious that these guidelines are limited to the present problem, given that the data normalization selection and nature of the problem can affect the magnitude of each loss component.



(a) Dynamic weight and ratio of data and PDE loss during training.

(b) Dynamic weight and ratio of data and PDE loss during training.

Figure 3.21: Training dynamics for various data weighting schemes.

3.3.3. Data and collocation points

Introduction

Another parameter that influences the training process is the number and distribution of locations where the loss is computed during the optimization. As previously exposed, these are commonly referred to in literature as *measurement* points, where $\mathcal{L}_{data}$ is evaluated, and *collocation* points, where the Navier-Stokes equations are enforced.

So far in the present thesis, these two have been specified to be equal to all locations where labeled data exists, as this ensures that representative reconstructed pressure values are obtained at all points where reference pressure data is available.

Nonetheless, several questions of interest arise regarding these points, their proportion and distribution, as it becomes obvious that, the smaller the total amount of points, the more efficient the training will be, provided that all other parameters remain unchanged. Accordingly, some of the questions that are targeted here are:

1. Should labeled data be available with a smaller level of resolution, would it be possible to achieve similar reconstruction errors for all states at all initial locations via the PDE loss?
2. Can the reconstruction of pressure be positively affected by the inclusion of additional collocation points?
3. To which extent are the key performance metrics of each variable affected by the increase or reduction in each type of points?

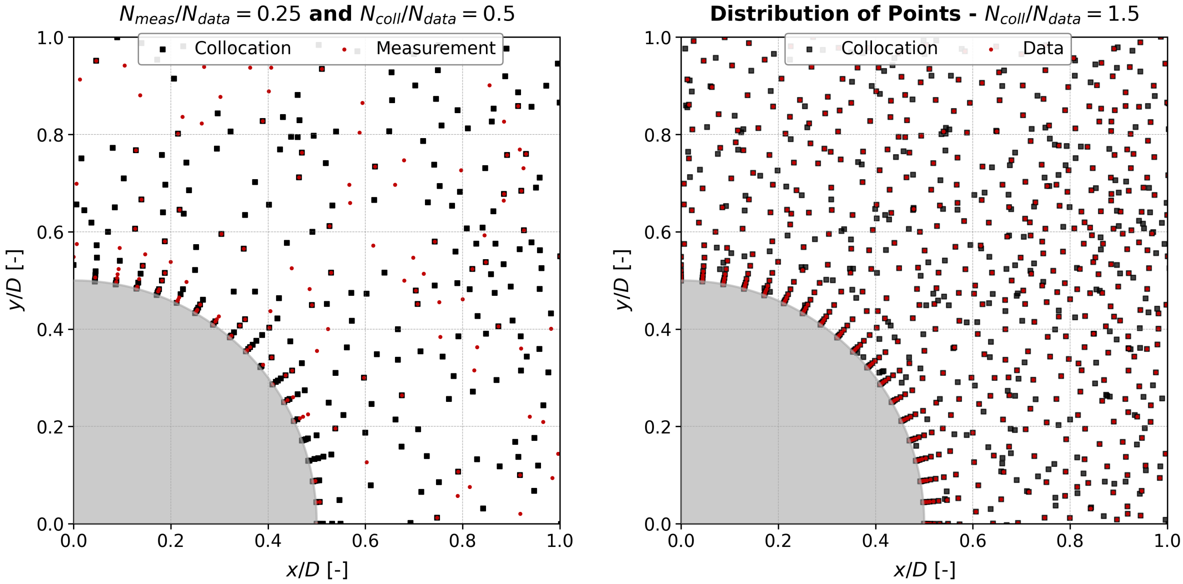
In order to address these questions, various tests have been conducted by training the *PINN* baseline model with different amounts of collocation and measurement points. Similarly to previous tests, the approach to define the test points has been a mixture of D-optimal and One-Factor-At-a-Time, that allows to properly discern the separate effects of both factors while covering the domain of interest with a reduced number of tests. In particular, the test matrix is as per shown in Table 3.9, where N_{meas} are the number of measurement points, N_{coll} are the number of collocation points and their amounts are defined as ratios to the total number of available data points from the CFD simulation, N_{data}.

In the design of the experiment, it is taken into consideration that the ground truth is only available at the CFD data point locations. Accordingly, the number of measurement points can only take values $N_{meas}/N_{data} \in [0, 1]$. When this ratio is below unity, the measurement points are randomly sampled from the entire pool of data points. On the other hand, the ratio of collocation points can theoretically take values of $N_{coll}/N_{data} \in [0, +\infty[$. Whenever the ratio is smaller than one, the same approach is followed, i.e. the entire batch of data points is randomly sampled. However, when this ratio becomes larger than unity, the collocations points are randomly sampled from a uniform distribution in the range of the domain dimensions.

Table 3.9: Number of measurement and collocation points study.

Test #	N_{meas}/N_{data}		N_{coll}/N_{data}	
	Level	**Value**	**Level**	**Value**
1	-1	0.25	-1	0.5
2	-1	0.25	1	1.5
3	1	0.8	-1	0.5
4	1	0.8	1	1.5
5	0	0.55	0	1.0
6	1.4	1.0	0	1.0
7	-1.4	0.1	0	1.0
8	0	0.55	1.4	1.7
9	0	0.55	-1.4	0.3

Given this approach, it can be deduced that the only tests where there will be a full overlap between measurement and collocation points are those where the ratio of collocation points is $N_{coll}/N_{data} \geq 1$. Otherwise, the random distribution can result in overlapping and non-overlapping regions. This random sampling approach is followed to provide an unbiased view on the extent of data resolution needed and on guidelines to define the number of collocation points in relative terms.



(a) $N_{meas}/N_{data} = 0.25$ and $N_{coll}/N_{data} = 0.5$. **(b)** $N_{coll}/N_{data} = 1.5$.

Figure 3.22: Data, measurement and collocation points on cylinder proximity for different point ratios.

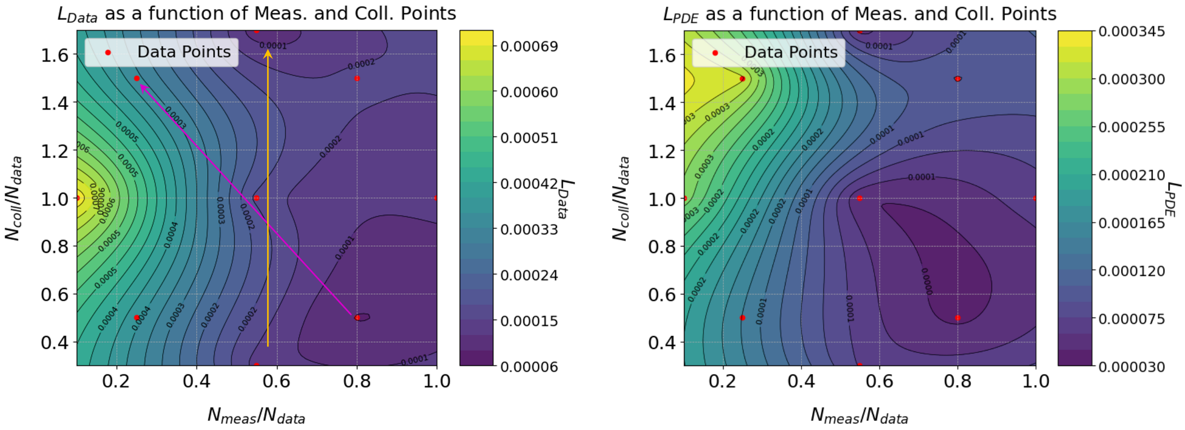
In order to provide a visual reference of the above description, Figure 3.22 is attached. On the one hand, to the left, Figure 3.22a includes the measurement and collocation points, in red and black respectively, for $Test\ \#1$, namely $N_{meas}/N_{data} = 0.25$ and $N_{coll}/N_{data} = 0.5$. As can be recognized, when both ratios are below unity, there is no need for them to overlap, since both distributions are randomly sampled from the total batch of data points. Moreover, in Figure 3.22b on the right, it is possible to see that, when the ratio of collocation points is larger than one, all data points are sampled and the remaining collocation locations are randomly placed across the domain.

Finally, it is noted that the mini-batch size, which is the amount of locations that are considered for the evaluation of the loss during training, is kept at 1/4 of the number of measurement points. This ensures that the batch size effect is left out of the equation, causing each epoch to be formed by four iterations on randomly sampled mini-batches.

Results

As a first look into the results, contours of the two main loss contribution terms are exposed in Figure 3.23, where the test points are reflected by red dots and the contours are produced with a second order inter and extrapolation method around the points.

The focus is first placed on Figure 3.23a, which depicts the dependency of the data loss to both parameters. Note that the data loss metric, which does not include the reconstruction error of pressure, is computed at all CFD data locations, independently of the number of points used during training. From the shape of the contours, a major trend can be identified which results in an inverse relation of the data loss term with the number of measurement points used during training. This can be inferred by looking at the evolution of the data loss along any horizontal line, with a consistent picture for any number of collocation points. For example, moving from the $\{1,1\}$ test to the $\{0.1,1\}$ result, an increase in the data loss from the $\mathcal{L}_{data} = 1 \times 10^{-4}$ level to $\mathcal{L}_{data} = 7 \times 10^{-4}$ is observed. Indeed, this trend is expected, given that feeding a bigger portion of the entire data batch allows to minimize the data loss further at all data locations.



(a) Data loss.

(b) PDE loss.

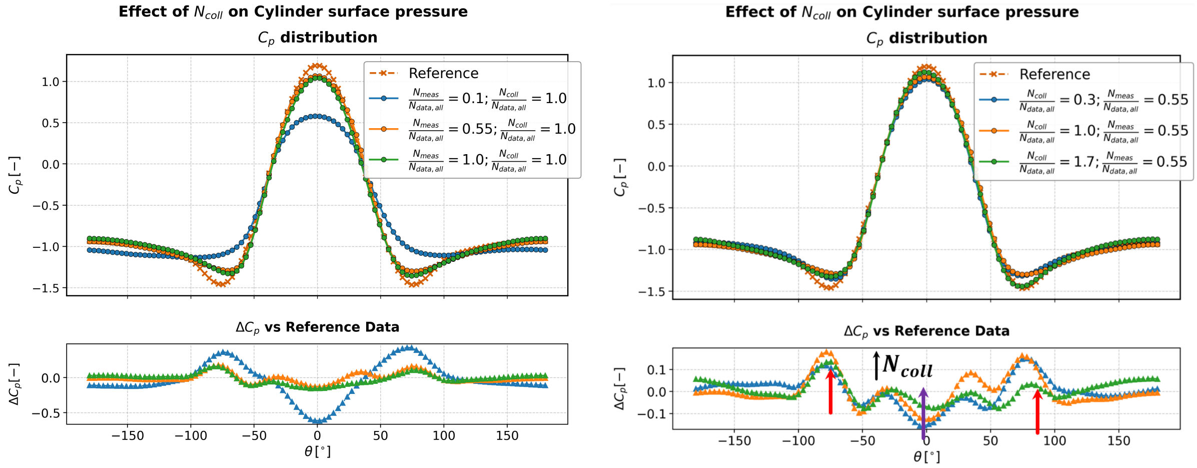
Figure 3.23: Contours of different loss terms as a function of the number of collocation and measurement points.

On the other hand, and to provide an answer to the first of the questions provided above, moving along a diagonal line such as the pink arrow allows one to realize that, should the number of data points diminish, the effect of the missing data would not be completely offset by additional collocation points. Not only this, but also displacement along a vertical line in the plot, such as the yellow line, depicts the limited extent to which collocation locations aid in the reconstruction of known states. This is because moving from $\{0.55, 3\}$ to $\{0.55, 1.7\}$ barely reduces the data loss from $\mathcal{L}_{data} = 1.3 \times 10^{-4}$ to $\mathcal{L}_{data} = 1.0 \times 10^{-4}$.

Secondly, the contours in Figure 3.23b reflect the extent to which the residuals of the N-S are minimized for each combination of measurement and collocation points. Interestingly, the major trend with respect to the relevance of measurement points is still present, as displacement to the left side of the domain along any horizontal line in the plot results in an increase in $\mathcal{L}_{PDE}$. Besides, it is also remarkable that modifications in the number of collocation points for a certain amount of measurement locations appears to yield no discernible advantage in terms of PDE loss minimization.

In order to better comprehend how these trends affect the main target of the thesis, the C_p distribution on the cylinder, Figure 3.24 is attached, showing the effect of the number of measurement points (Figure 3.24a) and number of collocation locations (Figure 3.24b) on the cylinder surface C_p distribution.

Focusing on the plot to the left first, it is clear to see that the version using only measurements at 10% of the total number of data locations incurs significant errors. These peak at $\Delta C_p \approx 0.6$ at stagnation and $\Delta C_p \approx 0.5$ at both suction peaks when compared to the reference distribution. However, only minor differences appear between the versions using 55% and 100% of the data points during training. This behavior differs significantly from the picture the two main loss terms offered in Figure 3.23, likely due to the large density of points placed in close proximity to the cylinder, which form the inflation layer mesh. This causes the 'random' point selection to be biased towards regions of high data point density in an inherent manner. Consequently, the distribution of the rate of change in reconstruction error is unequal, it being less significant for regions with fewer data points per unit surface.



(a) Effect of N_{meas}.

(b) Effect of N_{coll}.

Figure 3.24: Effect of measurement and collocation points on cylinder surface pressure.

Regarding the analogous plot in Figure 3.24b for the sensitivity to the number of collocation points, the extent of variability is significantly reduced. Even if a small inverse relation between the reconstruction error and N_{coll} is present at stagnation, reflected by the magenta arrow, its extent is very limited, namely $\Delta C_p \leq 0.1$. In fact, so small is the dimension of that trend, that it falls within variability of the accuracy of the $N_{coll}/N_{data} = 1.7$ model between the top and bottom suction peaks, as signaled by the red arrows.

Overall, this allows to partially answer question # 2 above, as it seems that, beyond a threshold, further collocation points have little influence on C_p accuracy. Again, this matter is **only partially** addressed because the increased point density in regions adjacent to the cylinder surface clouds the results.

Following with the analysis to provide an answer to question # 3 above, it is possible to provide a quantitative estimate for how the reconstruction error of each variable is affected by the number of data and collocation points. This information is contained in the bar plot of Figure 3.25, where the mean derivative of the mean squared error for each variable with respect to the number of collocation and measurement points are included.

Prior to discussing the results, it is of interest to explain how the following graph is obtained. Firstly, the mean squared error of each variable has been computed for each of the trained models in Table 3.9. Furthermore, to compute the derivatives, the 'One-Factor-at-a-Time' nature of the design of experiments has been used, thus calculating the derivative with respect to the number of measurement points at each level of collocation points and vice-versa. Finally, all derivatives for each variable are averaged out, and the standard variation computed, both shown as bars and wicks in the next Figure.

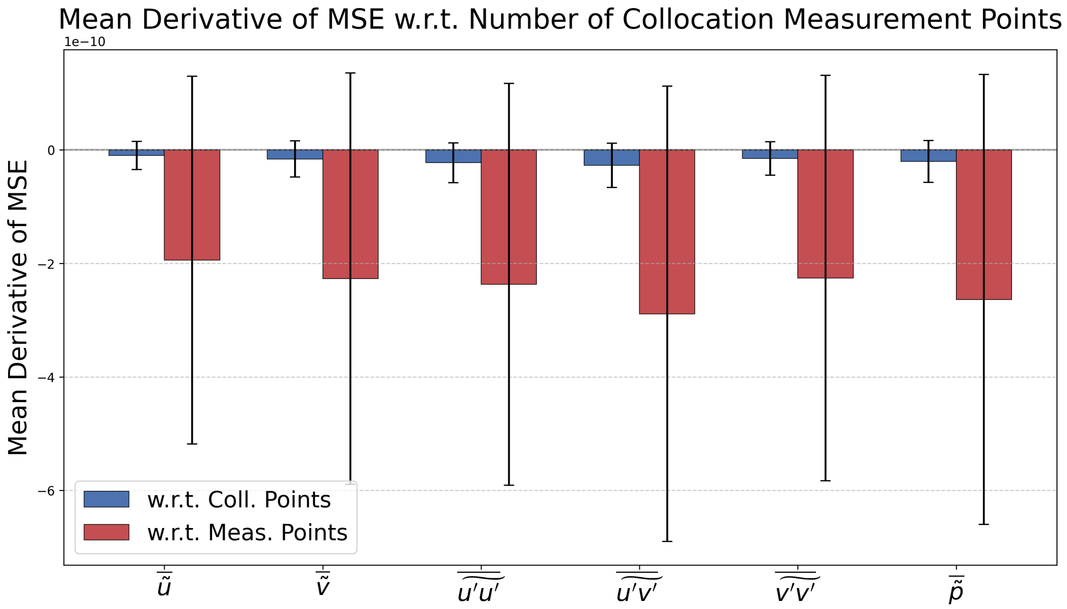


Figure 3.25: Mean variation of MSE for each variable with collocation and measurement points.

To give an example, the derivative of the x-velocity MSE with respect to the number of collocation points corresponding to Tests 1 & 2 from Table 3.9 would be as per Equation 3.13, where the number of measurement points is fixed at $N_{meas} = 0.25 \cdot N_{data}$.

$$\left.\frac{\partial MSE_{\bar{\tilde{u}}}}{\partial N_{coll}}\right|_{\frac{N_{meas}}{N_{data}}=0.25} = \frac{MSE_{\bar{\tilde{u}},2} - MSE_{\bar{\tilde{u}},1}}{N_{coll,2} - N_{coll,1}} \tag{3.13}$$

Delving in the actual results, it is possible to see that, on average, all derivatives take negative values, meaning that adding either type of points tends to reduce the reconstruction errors. Note that this is not always necessarily true, as all the wicks trespass into the positive range. Of special interest is the noticeable difference between the sensitivities to both types of points. In fact, quantifying the above hypotheses on the significantly larger impact of the number of measurement points, one can see that, on average, one additional measurement point is able to reduce the reconstruction error of most variables around $\sim 20\times$ more than adding a collocation point for the present problem and range of points studied.

Finally, the focus of attention is brought to the last of the questions exposed, that is, the addition of computation effort both types of points add. The method followed here to assess this in a practical manner is to measure the mean time it takes to evaluate either type of loss, namely PDE and data, depending on the amount of their respective types of points, the collocation and measurement.

Provided that the above study has been performed with a constant batch size for all loss types equal to one quarter of the total number of measurement points, an alternative approach is here considered to account for the possibility to include different amounts of measurement and collocation points at each iteration. This method is based on measuring the mean time it takes to compute each type of loss for batch sizes equal to one quarter of the collocation points (for the PDE loss) and one quarter of the measurement points (for the data loss). Following this procedure, the mean computation time and corresponding standard deviations for each combination of loss and number of points are included in the bar plot of Figure 3.26, where the data for each bar is obtained from $N = 100$ loss evaluations.

At first glance, a major delta can be observed between the computation times for the PDE loss and those for the data loss. The theoretical explanation for this phenomenon has already been exposed in §1.4.1, where the procedure to obtain the PDE loss is covered in detail. However, it is now possible to determine this difference for the present application in a quantitative manner, with the PDE loss taking approximately one additional order of magnitude in terms of time per computation.

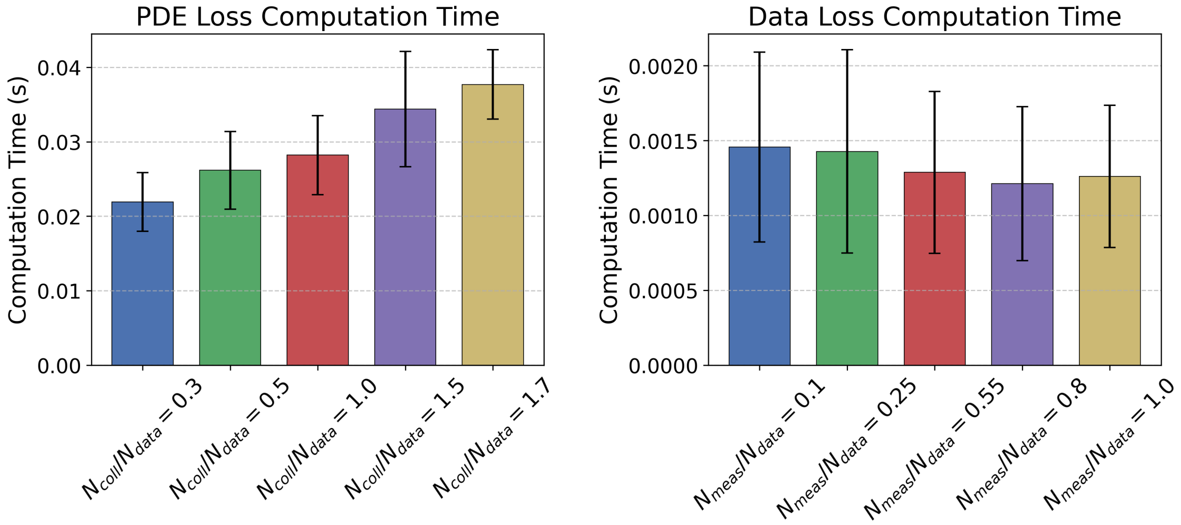


Figure 3.26: Mean time per type loss computation with $N = 100$ samples as a function of the number of points.

Besides, the computational effort for both loss terms exhibits unequal trends when the number of evaluation locations is altered. For the PDE loss, the computation time appears to grow linearly with the number of collocation points - as expected due to the need to compute the derivatives in the N-S equations in additional coordinates. On the contrary, the mean time per data loss computation seems to be insensitive to the addition of measurement points. This behavior is also anticipated, provided that the number of computations remains the same, with only matrix size in the MSE equations changing. In the *PyTorch* framework, where parallel computing is used, this adds barely any computational overhead, only raising memory requirements.

While it is possible to compute the mean additional time per collocation point by simply taking derivatives of the form of Equation 3.14, where index i indicates all values above $N_{coll} = 0.3$, since this is taken as the baseline time. It is noted, however, that this mean value of 0.00132 s per each additional 1000 collocation points is specific to this problem. It can be deduced from the explanation about how to compute the PDE loss that, even for problems using the RANS equations in two dimensions as PDE loss, the computational overhead largely depends on the NN shape, making it problem-dependent.

$$\frac{\partial T}{\partial N_{coll}} = \frac{1}{4}\sum_{i=2}^{4} \frac{T_i - T_1}{N_{coll,i} - N_{coll,1}} = 0.00132 \pm 0.00048\, s/1000\, pts \tag{3.14}$$

As a summary of the above, it becomes apparent that additional data which matches the quality standards of the dataset largely governs the ability to reconstruct all variables across the domain, doing so in a computationally efficient manner. On the other hand, it appears that the amount of collocation points could be less relevant for reconstruction accuracy above a given threshold, thus allowing to cut training time without incurring major performance reductions.

3.3.4. Boundary Condition loss

Introduction

As exposed in §2.2.2, the *PINN* framework makes the enforcement of boundary conditions a straightforward step through the addition of loss components which can take the form of problem-specific or physical constraints, even if these include local derivatives.

In the present work, the physical constraints inherent to viscous flows are enforced at the cylinder surface in the form of no-slip, no through-flow and no-fluctuations boundary conditions. These are defined by Equations 2.24, 2.25 and 2.26 respectively. These allow to add another source of information to guide the optimization process at the principal area of interest in the present study - the solid boundary.

Prior to assessing the effects introduced by the different boundary conditions on the key performance indicators, it is a matter of interest to investigate whether these are satisfied by the reference data, as well as by the baseline *PINN* model. This information is represented in the Figures of 3.27, where the three physical boundary conditions are represented on the cylinder surface. Specifically, these are computed at an uniformly distributed grid of $N_{pts} = 100$, where the reference data has been obtained through linear interpolation and the baseline *PINN* algorithm has been evaluated directly.

On the one hand, Figure 3.27a depicts the mean tangential speed at the cylinder surface, which should take null values for the no-slip condition to be fulfilled. Looking at the reference data in orange first, it can be recognized that the constraint is satisfied along the forward semi-circumference, namely for $\theta \in [0°, 90°] \cup [270°, 0°]$. On the contrary, the region affected by recirculation appears to present a deficit tangential speed in $\theta \in [90°, 180°]$ and an excess of the same in $\theta \in [180°, 270°]$. In the case of the baseline *PINN* prediction in blue, a different picture is observed, with the first and third quadrants containing an excess in angular speed, while the other two contain negative values.

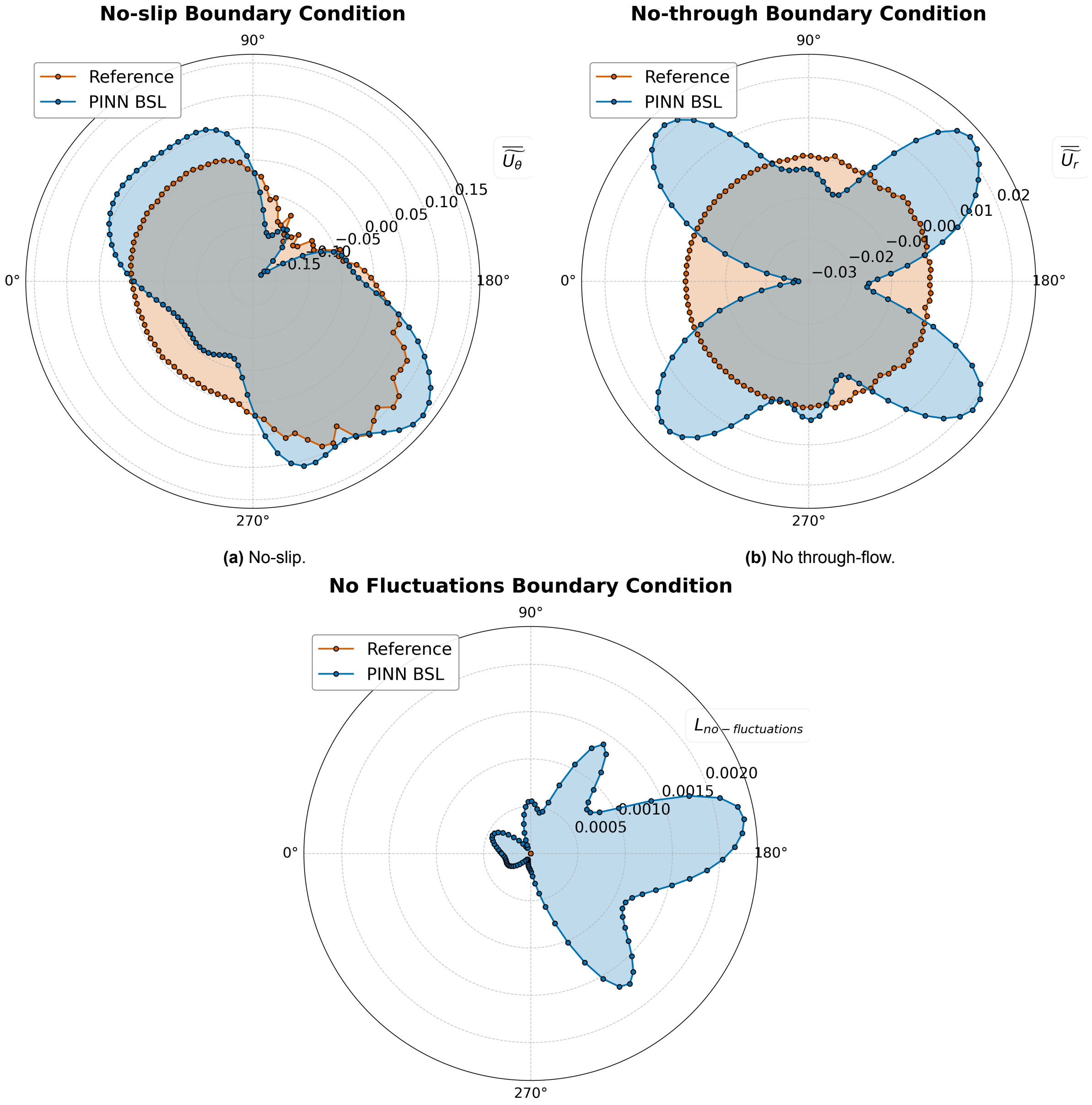
In second place, the mean speed in the radial direction is pictured in Figure 3.27b, where positive values reflect outward flow and negative values represent flow into the cylinder surface. In this instance, the reference data offers a seemingly perfect match, with in/out flows not overcoming $0.002 \times U_\infty$. Moreover, the baseline *PINN* predicts major inflows of up to $0.03 \times U_\infty$ at stagnation and slightly smaller outflows at all four diagonals, depicting a symmetric pattern around $y/D = 0$. Nonetheless, it can be firmly stated that the fulfillment of the no through-flow boundary condition is fulfilled by both versions to a greater extent than the no-slip boundary condition, given that the deviations from zero are nearly and order of magnitude smaller.

Finally, the no-fluctuations boundary condition loss, computed as the sum of the squares of all three Reynolds Stress components is attached in Figure 3.27c, offering a disparate behavior between both data sources. While the reference data presents a null loss value, which is exactly the accomplishment of the boundary condition, the NN model predicts loss values of up to $\mathcal{L}_{no,fluct} \approx 0.002$. Bearing in mind that this value is computed as the square of the sum of all three RS components, it can be considered a significant mis-prediction in absolute terms.

Additionally, prior to introducing the boundary condition loss, it is relevant to understand what the magnitude of the loss components are, especially for the *PINN* baseline results. This is because loss magnitudes which fall severely under the baseline total loss might require a non-unitary weighting to affect the optimization process. Accordingly, Table 3.10 includes all three loss terms for both data sources. Comparison of all three *PINN*-predicted loss values with the total loss for the same model, $\mathcal{L}_{Tot} = 1.18 \times 10^{-4}$, one can check that the no-through and no-fluctuations loss terms are of the same order of magnitude and the no-slip loss is one order of magnitude larger. Therefore, it is expected that an unweighted version of the boundary condition loss suffices to include the constraints in the training process.

Table 3.10: Boundary condition loss components - reference data and *PINN* baseline.

	$\mathcal{L}_{\text{no-slip}}$	$\mathcal{L}_{\text{no-through}}$	$\mathcal{L}_{\text{no-fluct}}$
Reference	2.52×10^{-3}	2.15×10^{-7}	5.63×10^{-9}
BSL *PINN*	6.28×10^{-3}	1.88×10^{-4}	7.17×10^{-4}



(a) No-slip.

(b) No through-flow.

(c) No fluctuations.

Figure 3.27: Physical boundary conditions on the cylinder surface - baseline *PINN* vs Reference.

Missing data close to the boundary

Once the main differences between the reference data and the *PINN* in the fulfillment of surface boundary conditions have been pointed out, it is important to note that the motivation to propose the inclusion of these loss terms becomes more obvious for experimental datasets. The reason is that, while CFD simulations often include higher grid resolutions close to the solid boundaries to properly capture the gradients imposed by the viscous nature of real flows, experimental datasets often have poorer quality or even missing data in regions adjacent to solid walls. This is because, during PIV experiments involving high-power laser beams, primary and secondary reflections abound, often making data acquisition close to solid boundaries extremely challenging. Additionally, simultaneous visibility access for the laser and imaging device can add complexity to near-the-wall acquisition, especially in internal flow applications or external flows in close proximity to the ground, such as various motorsport categories.

In accordance to the above, the present study focuses on studying to which extent enforcing physical behavior at the cylinder surface during the *PINN* training results in an increased ability to reconstruct surface pressure, particularly when data is missing close to the solid wall. To achieve this, various *PINN* models are trained on datasets where the measurement data closer than a progressively bigger radius, the cropping radius, or $\tilde{r}_{crop}$, is removed. In particular, the range of cropping radii that have been studied are shown in Equation 3.15.

$$\tilde{r}_{crop} = \{0.55,\, 0.625,\, 0.75,\, 0.875,\, 1.0\} \tag{3.15}$$

For reference, the measurement locations available for the versions with $\tilde{r}_{crop} = \{0.55,\, 0.75,\, 1.0\}$ are represented in Figure 3.28. This allows the reader to better understand the extent to which data is missing away from the cylinder boundary and associate each cropping radius run with the reason that could cause such a dataset. For example, the $\tilde{r}_{crop} = 0.55$ data could correspond to the effect of reflections, losing data only at $\tilde{r} \leq 0.1 \times \tilde{r}_{cyl}$. On the other hand, for experimental set-ups meeting good quality criteria, datasets such as $\tilde{r}_{crop} = 0.75$ and beyond could only be attained through accessibility issues.

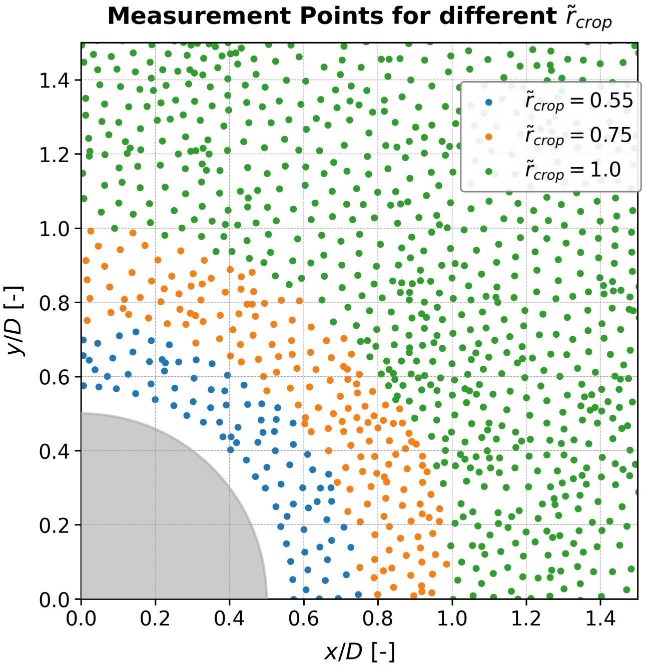


Figure 3.28: Locations where data is available during training as a function of the cropping radius.

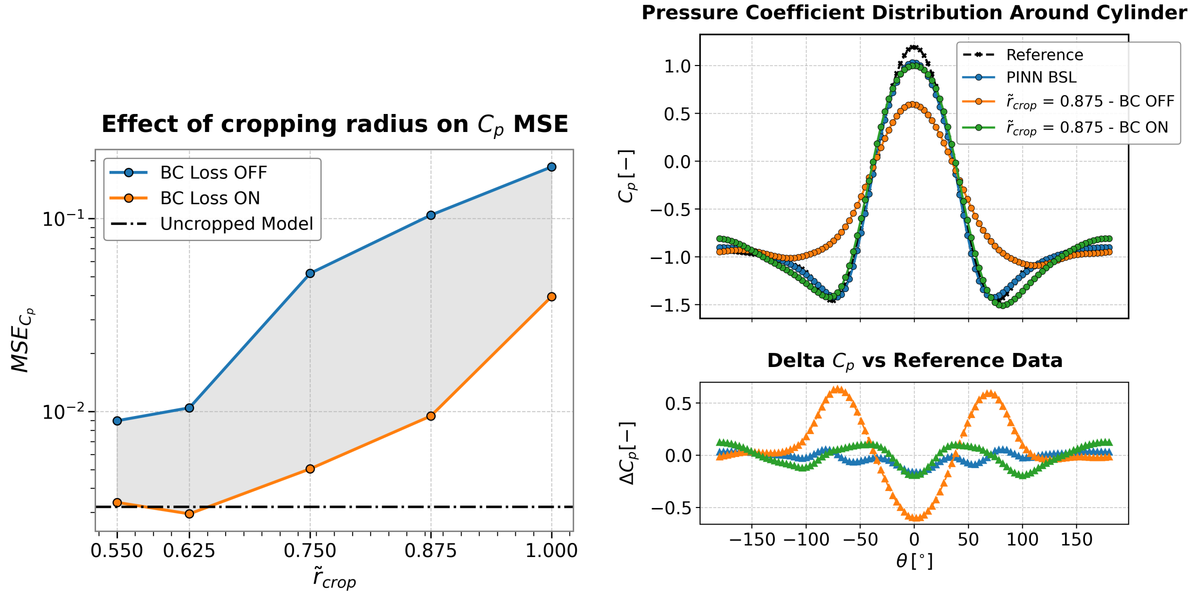
It is relevant to state that, while measurement points are not available at any location closer than the cropping radius, the original data points are kept as collocation points, thus allowing the *PINN* algorithm to reconstruct the data in the cylinder proximity via the N-S equations. In a similar manner, the boundary condition locations take the form of 100 uniformly distributed points on the cylinder surface. Later in the section, an investigation on the effect of the collocation point generation strategy is presented, as it would be required in any experimental dataset with data gaps close to the surface.

Results

The attention is brought first to the potential of reducing the impact of missing data close to the surface by including all three physical constraints. As a result, two *PINN* models, one with the boundary condition loss and the other one without it, are trained on the dataset corresponding to each cropping radius. In Figure 3.29a are shown the MSE values of C_p on the cylinder surface for each of these models, as well as the *PINN* baseline case with no cropped data, as a black dash-dotted line.

From the graph, it can be said that both sets of models follow the trend of increasing pressure coefficient error with respect to the reference data as the cropping radius increases, which is to be expected given the large sensitivity observed to the number of data points in previous studies. However, a consistently better outcome is achieved when the boundary conditions are enforced, achieving average MSE_{C_p} reductions of $\sim$ 80% across the range of radii. Further notice is given to the fact that, for the two smallest cropping radii, adding surface information in the form of physical constraints achieves the same accuracy level as the un-cropped *PINN* model.

Further detail on the sources of error for the larger radii can be found in Figure 3.30b, where the surface C_p distributions of both models corresponding to the $\tilde{r}_{crop} = 0.875$ are plotted alongside the baseline *PINN* and the reference data. While the BC-OFF model incurs errors of $\Delta C_p \sim 0.5$ around stagnation and suction peaks, not achieving to capture the regions with the largest gradients. On the contrary, enforcing the boundary conditions allows the optimization process to converge to the appropriate solution in both regions, highlighting the power of adding surface information with data gaps.



(a) MSE of surface C_p.

(b) Pressure coefficient distribution - $\tilde{r}_{crop} = 0.875$ vs PINN BSL vs Reference.

Figure 3.29: Effect of including the boundary condition loss on surface pressure reconstruction as a function of the cropping radius.

After reviewing the effect of including all boundary conditions on datasets with missing data, it is worth investigating further into the mechanism through which the pressure accuracy is improved. To do so, the boundary condition enforcement in broken down into its components, leading to models trained with the loss term combinations listed below, specifically on the dataset with $\tilde{r}_{crop} = 0.875$.

1. $\mathcal{L}_{no-slip}$
2. $\mathcal{L}_{no-through}$
3. $\mathcal{L}_{no-slip} + \mathcal{L}_{no-through}$
4. $\mathcal{L}_{no-fluctuations}$
5. $\mathcal{L}_{no-slip} + \mathcal{L}_{no-through} + \mathcal{L}_{no-fluctuations}$

In the first place, the boundary condition loss values for all three physical constraints are represented for each of the models in Table 3.11, where the reference data and baseline *PINN* results are kept for reference. Notably, including either of the no-slip and no-through boundary conditions causes both conditions to be met to a similar extent, irrespective of which is enforced, achieving values of $\mathcal{L}_{no-slip} \approx 6 \times 10^{-5}$ and $\mathcal{L}_{no-through} \approx 1 \times 10^{-5}$. However, these appear to be achieved at the expense of $\mathcal{L}_{no-fluctuations}$, which is increased in about two orders of magnitude with respect to the baseline *PINN*.

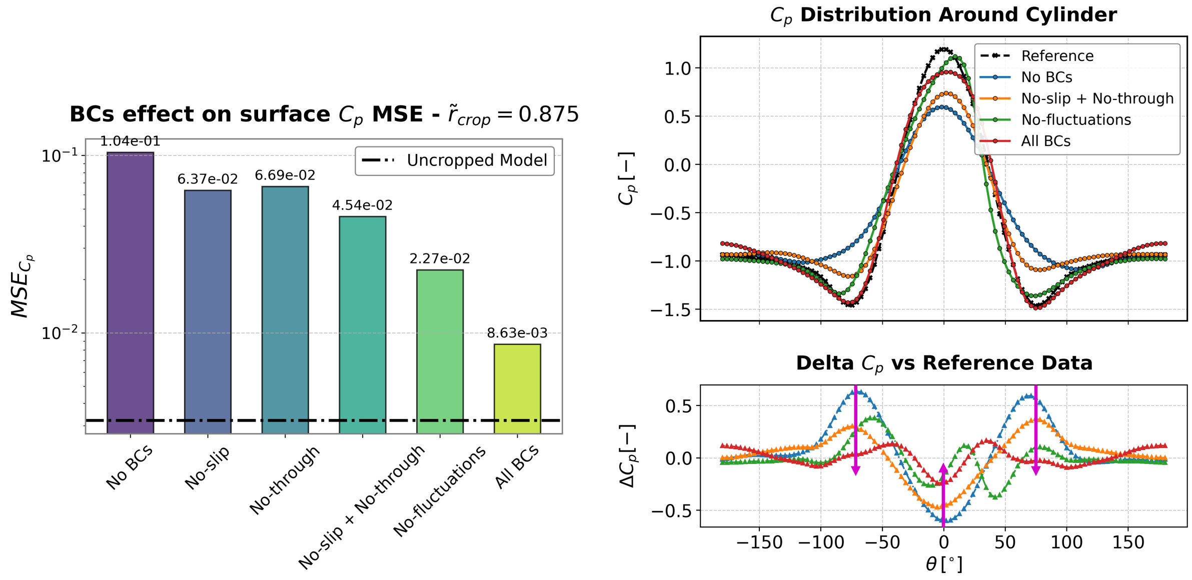
Additionally, employing both conditions simultaneously seems to enable the reduction of the no-slip and no-through constraints further still, not impacting the fluctuations loss.

Besides, the sole imposition of $\mathcal{L}_{no-fluctuations}$ allows its corresponding loss term minimization to achieve a value of 7×10^{-6}, again to the detriment of the un-constrained physical loss terms, which face increments of two orders of magnitude. Finally, imposition of all loss terms appears to find a compromise, the performance of which falls short by an order of magnitude with respect to the independently enforced constraints, with the exception of the no through-flow boundary condition that is reduced to $\mathcal{L}_{no-through} = 7.1 \times 10^{-6}$.

Table 3.11: Boundary condition loss components - effect of BC loss type.

Model	$\mathcal{L}_{\text{no-slip}}$	$\mathcal{L}_{\text{no-through}}$	$\mathcal{L}_{\text{no-fluctuations}}$
Reference	2.5×10^{-3}	2.2×10^{-7}	5.6×10^{-9}
PINN BSL	6.3×10^{-3}	1.9×10^{-4}	7.2×10^{-4}
$\mathcal{L}_{no-slip}$	6.3×10^{-5}	1.3×10^{-5}	2.6×10^{-2}
$\mathcal{L}_{no-through}$	5.8×10^{-5}	1.2×10^{-5}	2.6×10^{-2}
$\mathcal{L}_{no-slip} + \mathcal{L}_{no-through}$	2.8×10^{-5}	2.4×10^{-6}	2.3×10^{-2}
$\mathcal{L}_{no-fluctuations}$	5.0×10^{-1}	2.1×10^{-2}	7.1×10^{-6}
$\mathcal{L}_{no-slip} + \mathcal{L}_{no-through} + \mathcal{L}_{no-fluctuations}$	1.4×10^{-4}	7.0×10^{-6}	2.9×10^{-5}

However, focusing on how this materializes into a reduction of the main key performance indicator of interest, namely the surface C_p, the results in Figure 3.30a are referenced. The bar plot shown reflects the MSE of C_p at the cylinder surface when the different combinations of boundary condition loss terms are enforced on the $\tilde{r}_{crop} = 0.875$ dataset. Analyzing the results from left to right, it may be argued that more complex physical constraints appear to aid in the minimization of the surface pressure reconstruction error. Specifically, the model with no boundary conditions incurs the largest error, with $MSE_{C_p} = 1 \times 10^{-1}$. Nonetheless, the independent addition of the constraints on the mean velocities offer a similar step in accuracy, with the simultaneous addition attaining half the error of the model without boundary conditions. Similarly, preventing fluctuations aids to reduce error by another half, bringing it to $MSE_{C_p} = 2.27 \times 10^{-2}$. Finally, optimizing for all three loss values yields the lowest recorded reconstruction error, attaining an error only twice as large as that of the uncropped model.



(a) MSE of surface C_p.

(b) C_p distribution on the cylinder surface.

Figure 3.30: Effect of different boundary condition loss types - $\tilde{r}_{crop} = 0.875$.

It is relevant to note, however, that MSE is not necessarily always a representative measure of accuracy in the case of the C_p distribution, since a distribution with a crossover with the reference data will likely present a smaller MSE than a different one with a smaller, yet constant offset. Accordingly, it is of relevance to represent the pressure coefficient distributions by each of the models considered, which are plotted in Figure 3.30b.

Observation of the C_p distributions and ΔC_p plots allows to re-inforce the idea that the regions where largest spatial gradients in the wall-normal direction exist are the most susceptible to incur reconstruction errors. Accordingly, these regions see the biggest reductions in local errors when surface data is added in the form of physical constraints. Moreover, as reflected by the magenta arrows, there is a seemingly consistent trend in all three regions (stagnation at $\theta = 0^\circ$ and suction peaks at $\theta \approx \pm 75^\circ$), where the reconstruction error is inversely proportional to the amount of boundary conditions enforced.

These observations allow to conclude that, in principle, there is no specific boundary condition that yields superior results and the minimum error is consistently found for models including the most amount of information at the cylinder surface.

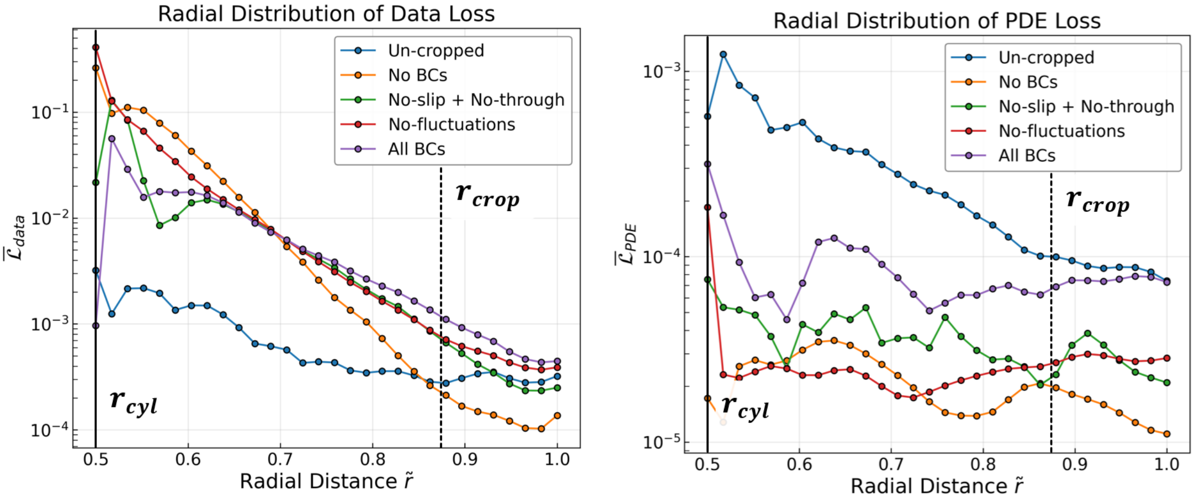
With the aim to comprehend the mechanism through which the additional surface data reduces the MSE_{C_p} on the cylinder, the effect of including the different boundary conditions on $\mathcal{L}_{data}$ and $\mathcal{L}_{PDE}$ in the near-the-wall regions is studied. To do so, the graphs in Figure 3.31 are included, where the evolution of the data and PDE loss components as a function of the radial distance to the cylinder are represented for various models trained on the $\tilde{r}_{crop} = 0.875$ dataset. To yield each point, both loss terms are evaluated at thirty-six evenly distributed locations along $\theta \in [0^\circ, 180^\circ]$, repeating the process for each of the radial locations in the plot.

Placing the focus of attention on the behavior of the data loss in Figure 3.31a, it is witnessed that all models converge to a similar reconstruction error level of $\overline{\mathcal{L}}_{data} \approx 3 \times 10^{-4}$ away from the wall, which coincides with the un-cropped model accuracy. However, within the small variability among the models, it seems that the bigger the amount of constraints on the surface, the larger the data loss at $\tilde{r} = 1.0$. Hypothetically, this could be related to the NN using a portion of its learning capability to store the surface data, limiting the extent to which error can be minimized in other regions. Moving closer to the wall, there appears to be a crossover at $\tilde{r} = 0.7$, as the models with larger complexity achieve progressively better results.

Within the inflation layer, namely $\tilde{r} \leq 0.56$ two distinct trends are identified. On the one hand, the model with no surface information (*No BCs*) keeps an increasing error trend, similarly as in the outer regions, leading to data losses as large as $\overline{\mathcal{L}}_{data} \approx 4 \times 10^{-1}$. An analogous behavior is found with the model where the null RS components are enforced on the surface. This is rather expected since the Reynolds Stresses represent a smaller portion of the local loss than the mean velocity components, the local error of which is largely worsened, as reflected in Table 3.11. On the contrary, the other models with boundary conditions enforced see a step reduction in the data loss, achieving even superior performance than the baseline model for the *All BCs* run.

Moving on to the radial distribution of the PDE Loss, two major observations can be made. Firstly, adding constraints on the data variables causes the PDE loss to grow. This applies not only for constraints in the form of surface data, but also for measurement data, as reflected by the behavior of the un-cropped model. Secondly, the models which performed the best see a sudden spike in the N-S loss near the wall. This suggests that the main mechanism through which the surface boundary conditions improve the reconstruction ability of the *PINN* is mostly tied to properly matching the variables themselves, rather than providing a constraint for the PDE loss that allows to properly reconstruct information from the cropping boundary.

As a final aspect of interest, different strategies are assessed with respect to the generation of the collocation points between the solid boundary and the cropping radius. In particular, two main topics of interest are identified that concern the sampling strategy. On the one hand, it is worth studying whether the progressive propagation of information towards the solid boundary is more effective than simply optimizing for all the collocation points at once. Secondly, studying whether the addition of a pseudo-inflation layer that mimics the grids utilized in wall-resolved CFD simulations has any effect on the reconstruction error.



(a) Data Loss.

(b) PDE Loss.

Figure 3.31: Radial distribution of loss terms with different BC Loss types - $\tilde{r}_{crop} = 0.875$.

Accordingly, the testing strategies reflected in the diagram of Figure 3.32 are compared. Starting from the top, a *PINN* model trained on the cropped data is used as a starting point for both approaches. Focusing on the *progressive front* method to the left first, the intention is to generate collocations points in a progressive manner from the cropping radius towards the solid surface, producing a new ring only after optimizing the loss for the newest one. Accordingly an initial ring with $N_{pts} = 200$ equally distributed points is generated at $\Delta\tilde{r} = -0.1$ from the cropping radius. Subsequently, iterations are carried out to minimize the Navier-Stokes loss at the given ring until a threshold loss value of $\mathcal{L}_{PDE,ring} \leq 5 \times 10^{-5}$ is attained. This loop is repeated until the cylinder surface is reached, always keeping all previous ring points in the optimization batch. It can be deduced that, since this study is carried out for $\tilde{r}_{crop} = 0.75$, the total number of points for the progressive front run is equal to $N_{pts,total} = 5000$, as per Equation 3.16.

$$N_{pts,tot} = N_{pts} \cdot \frac{\tilde{r}_{crop} - \tilde{r}_{cyl}}{\Delta\tilde{r}} = 200 \cdot \frac{0.75 - 0.5}{0.01} = 5000 \tag{3.16}$$

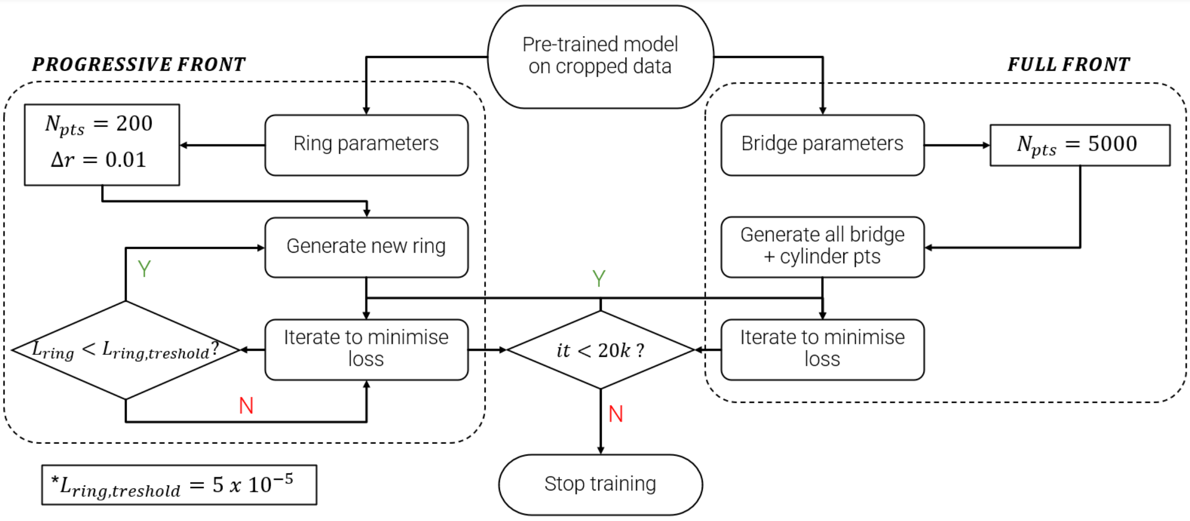
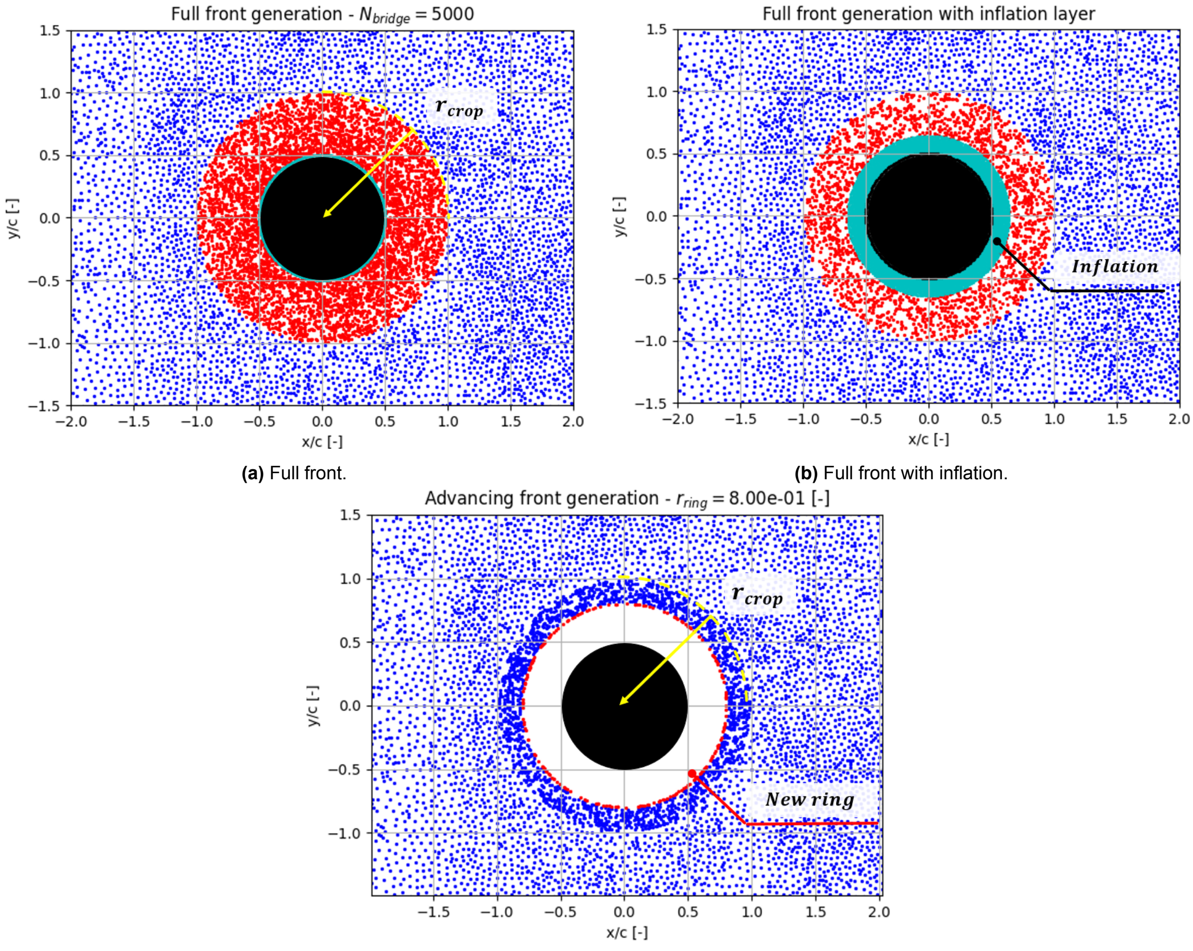


Figure 3.32: Diagram reflecting the procedure to assess the effect of different collocation point generation strategies.

The alternative collocation point generation mechanism, namely the *full front* approach, relies on random sampling in the domain given by $0.5 < \tilde{r} < 0.75$ and $0 \leq \theta \leq 2\pi$ to obtain $N_{pts} = 5000$ locations where the PDE loss is evaluated. Note that, opposed to the progressive algorithm, the entire batch is used at each step, provided that the sampling takes place prior to the training process.

Moreover, the number of $N_{pts} = 5000$ points is selected for both strategies because this is approximately the number of points removed from the original dataset when cropping at $\tilde{r}_{crop} = 0.75$, thus removing the (small) effect of having additional collocation points.

In addition to the above, the effect of adding a pseudo-inflation layer is probed in the full-front version, adding a back-to back comparison for this effect. Instead of attempting to match the inflation mesh generation of the CFD simulation, an even bigger refinement level is imposed with a similar thickness to create a bigger difference with respect to the model with no inflation layer. This results in a radial grid generation with a first layer height of $\delta_1 = 0.005$ and $N_L = 15$ layers with a radial growth rate of $GR = 1.1$. Note that, in angular terms, these nodes are aligned with the cylinder surface grid formed by $N_{pts} = 100$ evenly spaced points.



(a) Full front.

(b) Full front with inflation.

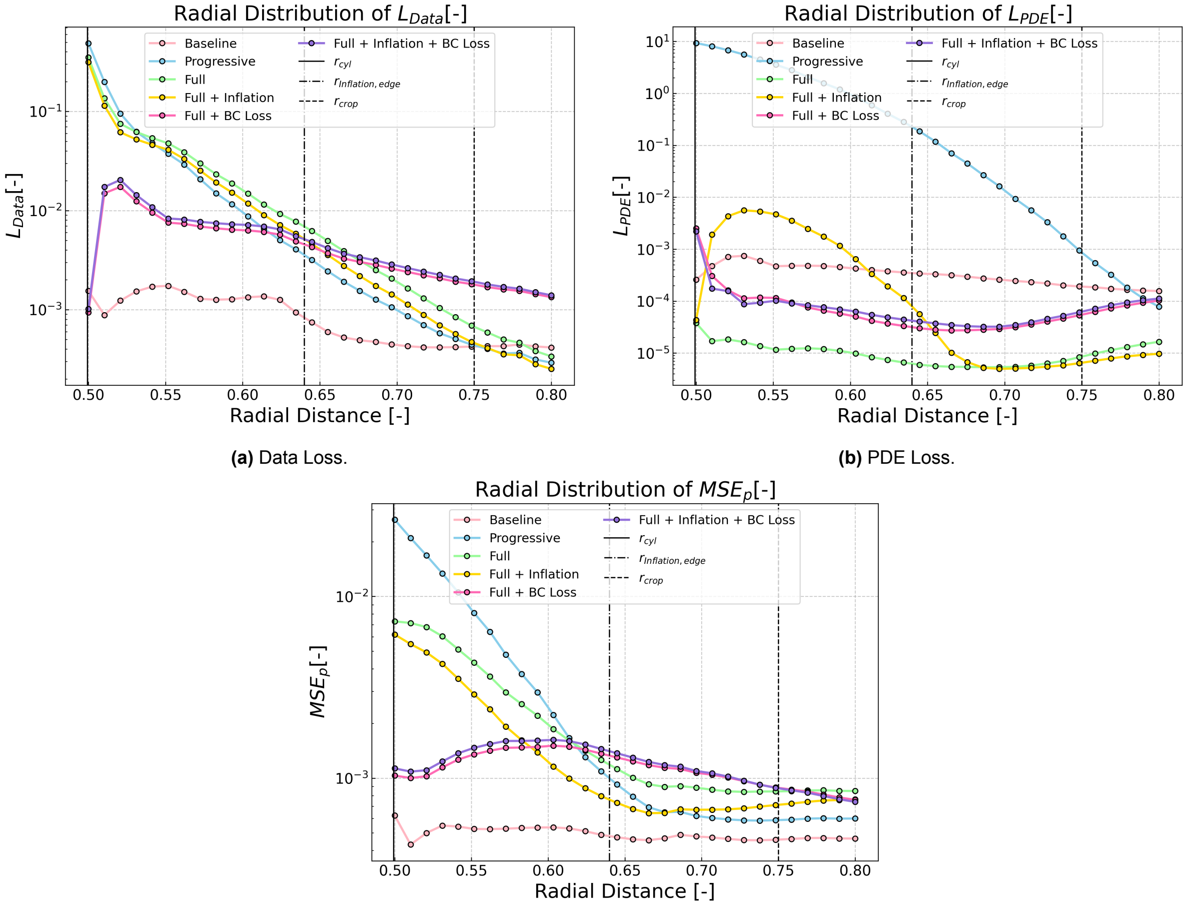
(c) Progressive front.

Figure 3.33: Comparison of collocation points for different bridging strategies.

To provide a visual reference of the different strategies, Figure 3.33 is enclosed, containing all three versions for a cropping radius of $\tilde{r}_{crop} = 1.0$. On the top left, the full-front method is reflected in Figure 3.33a, where the blue points are the original CFD locations, which are used to compute the data and PDE losses. Besides, the locations in red denote the randomly sampled positions that serve as collocation points to bridge the gap between the data and the solid boundaries. Similarly, Figure 3.33b depicts the analogous distribution of collocation locations, however adding the inflation mesh, which visibly includes a significant larger local point density. Note that in this version, the randomly sampled points do not extend to the solid boundary but rather to the outer edge of the inflation mesh. Finally, the progressive front alternative is represented in Figure 3.33, where the newly generated ring is reflected in red.

To assess the effect of the different strategies on the reconstruction accuracy, the mean radial distribution of the two main loss components, as well as that of the pressure MSE are included in Figure 3.34 for the $\tilde{r}_{crop} = 0.75$ case. Note that, additionally to the three methods, the baseline model is represented, as well as both full-front versions including all three boundary conditions at the cylinder wall.

Comparing first the point generation mechanisms with no boundary conditions, all three strategies appear to provide similar error levels across the range of radii studied, as per Figure 3.34a, signaling no major differences in the ability to propagate information from the data boundary towards the surface. Slight differences, however, appear in Figure 3.34b, where the progressive ring generation approach incurs a steadily increasing PDE loss up to $\overline{\mathcal{L}}_{PDE} = 10$. Comparing both full-front models, they exhibit equal behavior outside the inflation boundary, at which the model featuring the inflation points faces a local detriment of the N-S loss minimization, converging to a similar value of $\overline{\mathcal{L}}_{PDE} = 4 \times 10^{-5}$ at the cylinder surface. Finally, these behaviors translate into a marginally better performance of the full-front mechanism when it comes to pressure reconstruction accuracy. Moreover, no significant advantages appear to be achieved by the addition of the inflation layer mesh.



(a) Data Loss.

(b) PDE Loss.

(c) Mean Squared Error of dimensionless pressure.

Figure 3.34: Effect of sample point generation strategy on the radial distribution of KPIs.

Finally, looking at Figure 3.34a and bearing in mind that all models stem from a parent *PINN* that has been pre-trained on the data beyond $\tilde{r}_{crop}$, it is interesting to see that all models tend to the same mean data loss beyond the cropping location, except those where the boundary conditions are enforced. Since these offer better performance at the surface at the expense of increased errors away from the wall, it could be argued that these should be enforced solely where surface data is the main target, rather than blindly applying them. Furthermore, these results further reinforce the idea that imposing the BCs on the wall acts to reduce pressure reconstruction errors via a better match of the other variables, rather than through the minimization of the PDE loss.

Overall, the analysis conducted on the use of the no-slip, no through-flow and no-fluctuations boundary conditions at the surface has lead to insightful findings. On the one hand, it appears that, whenever data is missing close to the wall, their inclusion during training consistently aids in the reduction of the pressure reconstruction error at the solid boundary. Additionally, the no-slip and no through-flow conditions appear to have similar effects, as the optimization of one tends to encompass the other one. However, it is found that the simultaneous use of the three proposed loss terms consistently brings the most accurate results. Finally, it is concluded that the strategy to generate the collocation points that allow to reconstruct data gaps close to the surface is, at least, second to the inclusion of physical constraints at the surface. Nonetheless, marginal gains appear to be found by mimicking the inflation layer of CFD simulations and optimizing for all collocation points from the start, rather than progressively adding them towards the surface.

3.4. Observations from Chapter 3

Prior to assessing the performance of *PINNs* on experimental datasets, it is of interest to highlight some of the key observations in this Chapter.

First and foremost, it has been demonstrated that the selected framework is able to successfully reconstruct global and surface pressure values with accuracies of up to XX% and XX% respectively, with the baseline settings.

Moreover, study of the effect of various parameters on its ability to approximate the flow solution has revealed key insights on the different sensitivities:

- On the one hand, the NN architecture plays a key role on the ability to optimize the model. Since too simple networks do not appear to have sufficient learning power, it is recommended that $N \geq 20$ or $L \geq 4$. Nonetheless, as too complex architectures do not appear to provide further ability to learn the data, it is recommended that the total number of trainable parameters is kept to around one half of the total number of data points.
- Secondly, it appears that the non-dimensionalization of flow variables with physical properties bound to the problem allows to balance the contributions from the PDE and data losses naturally, avoiding the need to use weighting schemes for the main loss components.
- Besides, data resolution in the form of the number of labeled data points appears to be one of the most influential parameters, equally affecting the reconstruction accuracy of the known and hidden variables without incurring additional computational overhead. On the contrary, addition of collocation points via random sampling appears to be of, at least, second order of magnitude in terms of results accuracy, adding an approximate 0.00132 s per iteration every 1000 points (for the given computational server and NN shape).
- Finally, it has been demonstrated that the addition of physical boundary conditions at the solid surface in the form of the no-slip, no-through flow and no-fluctuations constraints allows to consistently reduce the surface pressure reconstruction error when data gaps exist around the body of interest. In this regard, it has been proven that neither the addition of a pseudo-inflation layer nor the technique to add collocation points from the data boundary to the solid object appear to play a relevant role.

4
PINNs performance on experimental results

After studying the ability of the selected *PINN* framework to reconstruct the time-averaged pressure field of the flow around a two-dimensional cylinder from Computational Fluid Dynamics simulation data, the present Chapter presents an extension of the problem to experimental data. Specifically, and as elaborated in Chapter 1, the intent is to compare the performance of *PINNs* with that of state of the art pressure reconstruction techniques, such as the Poisson solver, when compared to conventional experimental direct pressure measurement techniques, like pressure taps. To this end, the experimental datasets of the flow around a smooth cylinder and one fitted with zig-zag strips to trigger turbulent transition are used as test cases.

Accordingly, this Chapter starts with a brief introduction of the procedure that has been followed to produce the experimental data via Particle Image Velocimetry, as well as the Poisson solver method used. Subsequently, following an analogous structure to Chapter 3, the performance of the baseline *PINN* framework is analyzed and compared with the various sources of pressure data for both the smooth and the zig-zag strips cases. Finally, the sensitivities of results to various NN and *PINN*-related parameters are studied, placing the focus on how these may differ from the CFD observations.

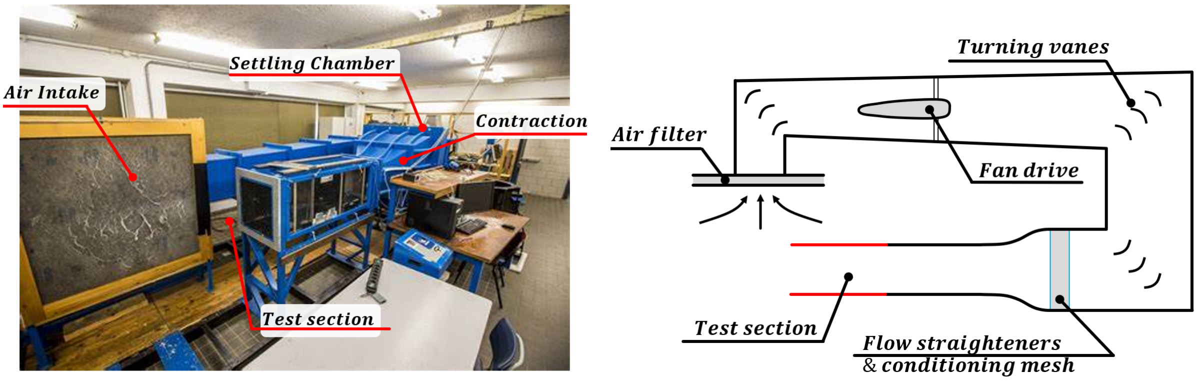
4.1. Dataset Generation

In this section, the details of the experimental set-up to obtain both the smooth and zig-zag strips datasets are exposed. Firstly, a brief description of the wind-tunnel facilities is given to provide context and some of the constraints that limit the testing envelope, resultingi in the cylinder geometry selection. Following, the set of characteristics of the PIV set-up are described and justified according to the target dataset, leading to a brief exposure of the results obtained. Furthermore, the set-up details of the pressure tapping measurement and the Poisson solver are exposed, closing the loop on the data-gathering process with a deeper analysis of the two flow-fields.

4.1.1. Wind-Tunnel facility and cylinder geometry

M Wind-Tunnel

One of the first considerations when planning experimental aerodynamic tests are the characteristics of the wind-tunnel to be used. In this thesis, the M Wind-Tunnel facility in the Low-Speed Lab of TU Delft is used. This facility, first constructed in 1953 [62], is an atmospheric wind tunnel that can operate in open-loop mode, providing a maximum test speed of 35 m/s [19]. In the present text, this is used with a closed test section of $S_{WT} = 400 \times 400\, mm^2$, with reported turbulence intensity levels of $T_I \leq 0.5\%$ within the operating range.



(a) Photo in open test section configuration.

(b) Orientative 2D diagram.

Figure 4.1: M Wind-Tunnel facility.

In Figure 4.1a, an image of the M-tunnel in open-loop and open test section configuration is enclosed, highlighting some of the most representative parts of the tunnel, such as the air intake, the settling chamber or the contraction section. Further insight is given by the diagram in Figure 4.1b, which is not built to scale but rather to enhance the reader's understanding. In the latter, key internal components are highlighted, such as the air filter to limit internal debris accumulation, the fan drive, turning vanes or the components of the settling chamber, which contribute to the flow quality.

Cylinder geometry

Given the availability of a suitable test section from other tests in the University, a cylinder diameter of $D_{cyl} = 50\,mm$ is selected, providing a blockage ratio of 12.5% for the given test section. Accordingly, assuming standard conditions for air properties, the range of Reynolds Numbers that can be tested is of $Re \in [1.7 \times 10^4, 1.2 \times 10^5]$ for testing speeds in the range $U_\infty \in [5, 35]\,m/s$. Taking into consideration the C_d vs. Re plot provided in Figure 3.1 of Chapter 3, it is possible to deduce that the test range is thus constrained to the drag plateau, characterized by the turbulent wake regime, where the boundary layer is laminar at separation and transition takes place in the free shear layer.

Provided that a similar flow regime is expected irrespective of the selected test speed, an intermediate value of $U_\infty = 20\,m/s$ is chosen, ensuring sufficiently high static pressure difference values while not pushing the envelope of the wind tunnel operating regime. Using the air properties as measured during the experimental tests, the nominal test Reynolds number is of $Re = 6.7 \times 10^4$, as shown in Equation 4.1.

$$Re = \frac{\rho \cdot U_\infty \cdot D_{cyl}}{\mu} = \frac{1.18\,\frac{kg}{m^3} \cdot 20\,\frac{m}{s} \cdot 0.05\,m}{1.75 \times 10^{-5}\,Pa \cdot s} = 67430 \tag{4.1}$$

Furthermore, since a different flow regime can be attained easily and in a cost-effective manner, the decision is made to conduct tests with zig-zag strips. Such aerodynamic devices are typically used for drag reduction purposes by tripping the boundary layer of the surface they are glued on. This is achieved locally via the introduction of physical disturbances above the critical roughness. Accordingly, their thickness is typically a compromise between a sufficiently high value to ensure transition and as thin as possible to limit their local pressure drag [74].

In this case, the decision is made to fit them at $\theta = \pm 45°$, as literature suggests it provides the clearest signs of transition for the Reynolds number of this test. This is shown in Figure 4.2, where the effect of the strips azimuthal location on cylinder drag coefficient is reflected for zig-zag strips with $t/D_{cyl} = 0.01$.

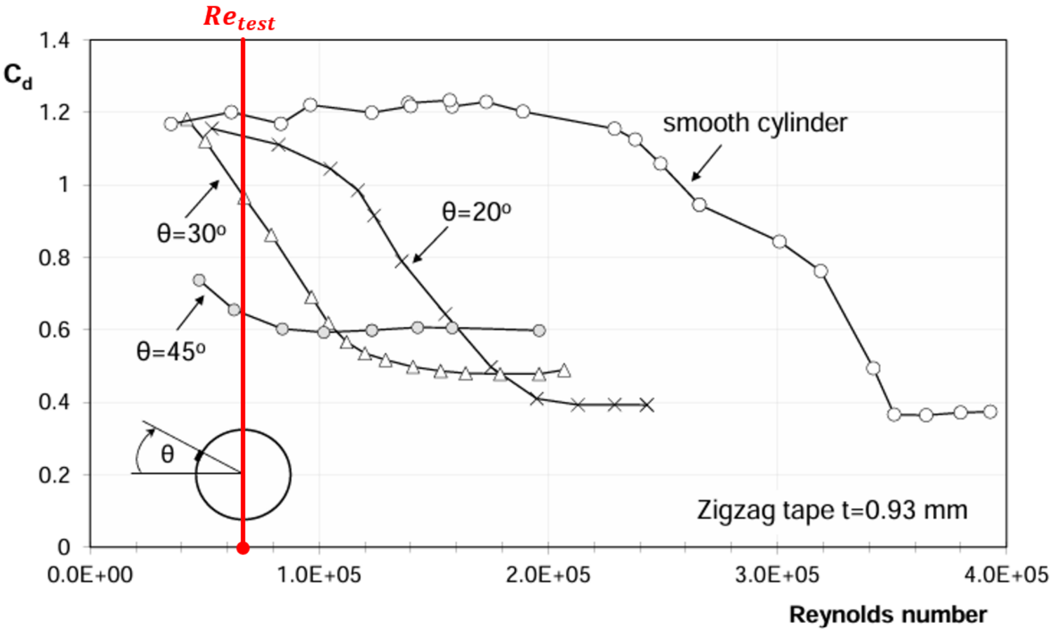
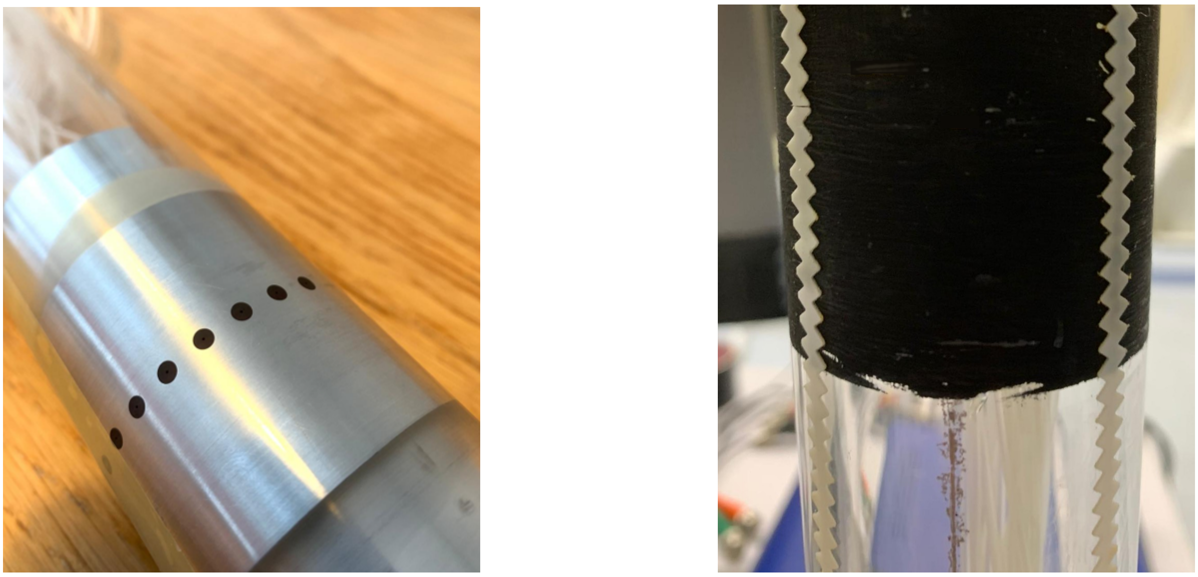


Figure 4.2: Effect of zig-zag strips azimuthal location on C_d- adapted from [74].

Specifically, the zig-zag strips used during the present tests have a thickness of $t = 0.6\,mm$ and a width of $w = 4\ mm$, resulting in a relative thickness of $1.2\,\%$ of the cylinder diameter. These can be seen in white in Figure 4.3b, where a portion of the cylinder has been painted in black Musou paint to limit laser reflections during PIV tests. Additionally, an image of the smooth cylinder configuration is attached in Figure 4.3a, providing a close-up of the static pressure ports around the circumference at the middle of the cylinder.



(a) Smooth cylinder - pressure tap close-up.

(b) Cylinder with zig-zag strips at $\theta = \pm 45^\circ$.

Figure 4.3: Test cylinder model.

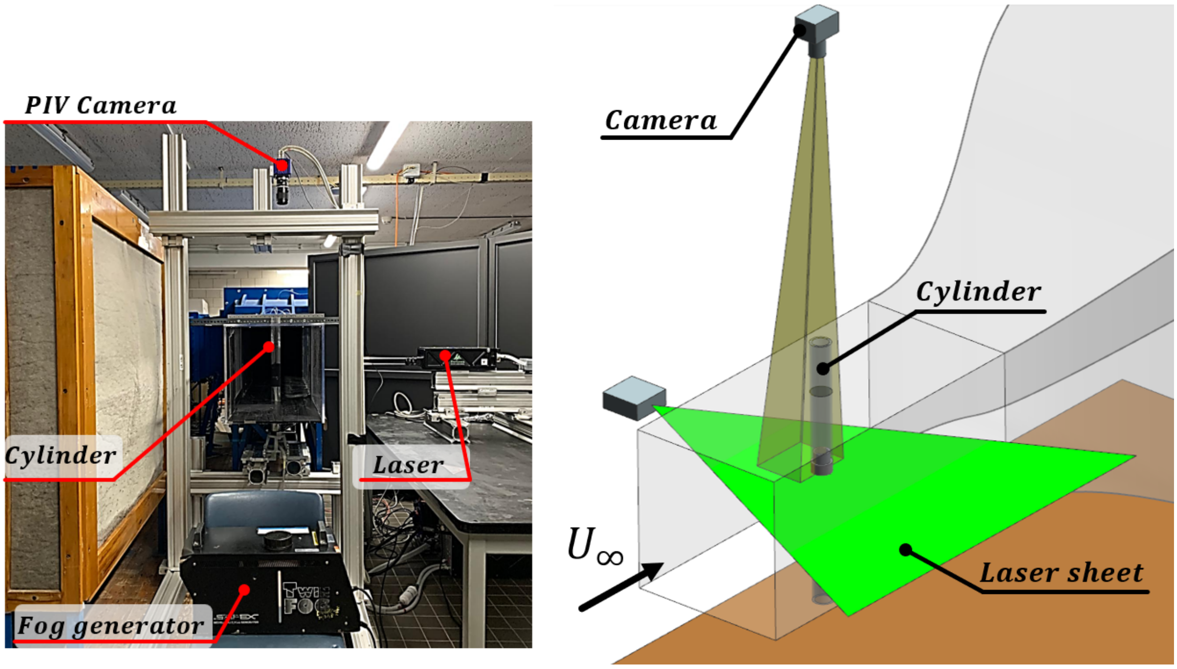
4.1.2. Experiment set-up and data acquisition

Test set-up

Analogously to the CFD simulations, the intent is to obtain the dataset for the two-dimensional flow around a cylinder. For this purpose, the test set-up reflected in Figure 4.4 is used.

On the one hand, the cylinder is placed such that it traverses the test section vertically. Accordingly, in order to capture the two-dimensional flow around it, the laser sheet is placed normal to the cylinder axis, illuminating the tracer particles contained in a planar section of the cylinder, as depicted in the CAD diagram in Figure 4.4b. In particular, the *EverGreen EVG00145* dual pulse laser in Figure 4.4a is used, which operates with $532\,nm$ wavelength light and provides firing frequencies of up to 15 Hz. It is noted that, since the cylinder is painted to avoid laser reflections, the illuminated area is limited to the top half of the cylinder. Nonetheless, assuming negligible non-uniformities in the inflow, this is sufficient as the time-averaged flow-field is symmetric around the $y = 0$ line.

In accordance, the 16 available pressure taps in the cylinder are arranged so as to maximize the information gathered in the same semi-circumference as the PIV acquisition. The layout can be partially observed in Figure 4.3a, and is such that 13 taps are equally spaced ranging from stagnation ($\theta = 0°$) to the back of the cylinder ($\theta = 180°$) at steps of $\Delta\theta = 5°$. The remaining three taps are placed such that the cylinder can be properly aligned with the inflow via comparison with their symmetric counterpart taps. As a result, the last three orifices are located at $\theta = \{185°, 270°, 355°\}$ from stagnation.



(a) Test set-up at the M-tunnel.

(b) CAD diagram of the test set-up.

Figure 4.4: Set-up for experimental tests.

Additionally, given that the flow pattern is assumed to be purely two-dimensional, a stereoscopic PIV set-up is not required, hence making use of a single *LaVision Imager sCMOS* camera placed in alignment with the cylinder axis, as displayed on either of Figures 4.4a and 4.4b. Finally, the PIV test set-up is completed with the SAFEX fog generator, placed adjacent to the inlet and below the test section to ensure smoke particles are diffused inside the room and provide a homogeneous seeding density at the test area.

On the other hand, the pressure acquisition set-up is simpler, as the pressure taps are directly connected to a 16-port 600 Pa gauge pressure scanner, which transfers the time signals to the control room computer at a frequency of 2 kHz.

Note that, with the exception of the fog generator, all controls and data acquisition processes are carried out from the the data acquisition and control computers, the former of which communicates with the camera and laser via a control unit that synchronizes the firings during the tests.

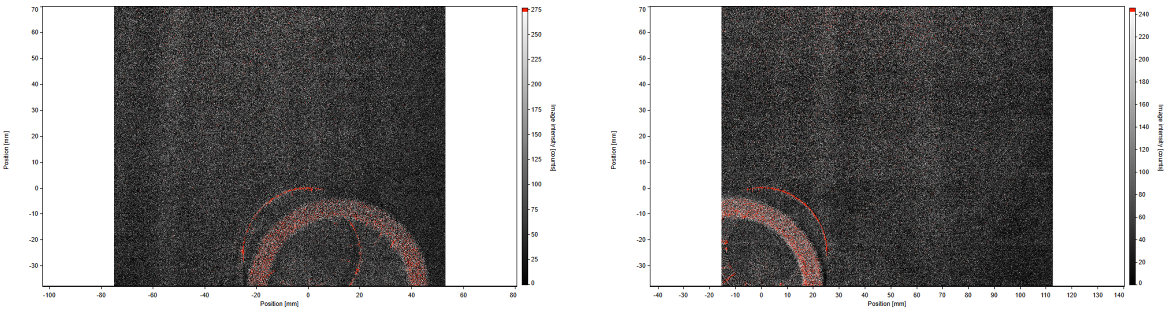
Characteristics of PIV set-up

In order to define optimal settings for PIV data acquisition, an initial target acquisition of three cylinder diameters in X is set, resulting in a $150 \times 127\,mm$ field of view, provided the $2560 \times 2160\,px$ camera resolution. Additionally, taking into account the pixel pitch of $6.5\,\mu m$, an image magnification factor of $M = 0.11$ is obtained, yielding an approximate particle image size of $d_p = 1.1 \times 10^{-1}\mu m$ before considering light diffraction effects.

Furthermore, a camera lens with $f = 105\,mm$ focal length is selected to provide a suitable object distance of $d_{obj} = 1030\,mm$ between the camera and the acquisition plane. Besides, targeting an image particle of $d_\tau \sim 2\,px$, an optimal $f_\#$ is found at 16 in combination with a high-power laser setting, allowing to get a depth of field as large as $\delta z \approx 67\,mm$ that can accommodate tests with an inclined laser sheet.

Finally, to achieve target pixel displacements of approximately $\Delta_{px} \approx 8\,mm$ that meet the one-quarter rule of thumb for a nominal free-stream speed of $U_\infty = 20\,m/s$, a time-step of $\Delta t = 23\,\mu s$ between image pairs is used across the tests.

It is noted, however, that in order to achieve optimal data that aligns with the main thesis intent of reconstructing surface pressure data, the camera is slightly offset laterally and two different arrangements corresponding to up and downstream displacements of the camera are used. This allows to minimize the loss of data close to the cylinder surface at the acquisition section due to the side effects of perspective. An instance of the output data is included in Figure 4.5, where the up and downstream frames after a minimum subtraction filter operation are shown in Figures 4.5a and 4.5b, respectively. These frames are processed separately and their respective time-averaged datasets are merged.



(a) Upstream.

(b) Downstream.

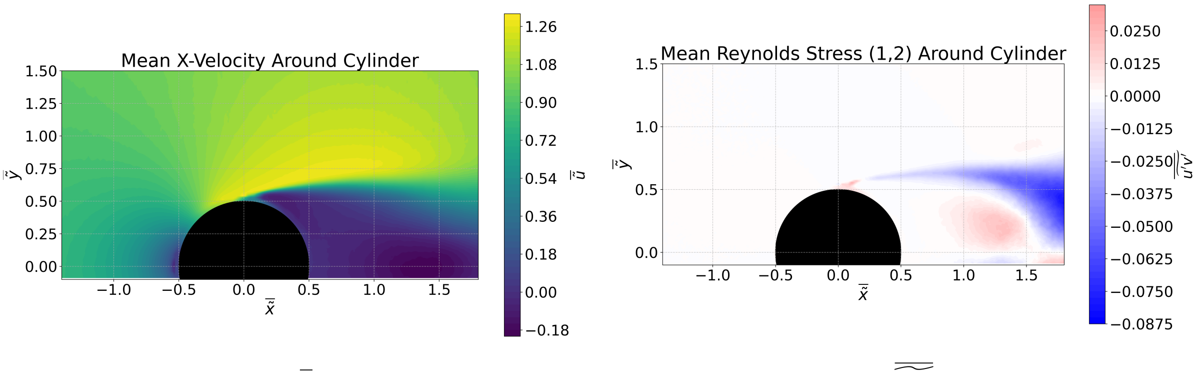
Figure 4.5: Camera configurations used during the tests.

During the tests, $N = 2000$ image pairs are recorded for both the smooth and zig-zag strip cases, leading to an approximate image acquisition time of $T = 130\,s$ at an acquisition frequency of $f = 15Hz$. Subsequently, these images are processed using a multi-pass algorithm with final circular correlation windows of $16 \times 16\,px$ and a 75% overlap. Additionally, an universal outlier detector and vector interpolation post-processing techniques are used to minimize data gaps across the captured domain.

Finally, the time-averaging process is carried out, providing the mean x and y velocity fields, as well as all three Reynolds stress components in two dimensions. Besides, considering the amount of samples taken and the range of the velocity fluctuations, the level of uncertainty in the mean velocity field can be computed via Equation 4.2, assuming independent and identically distributed samples. As observed, using a 3-sigma criterion ($k = 3$), a relative uncertainty of $\epsilon = 0.023 \cdot U\infty$ is estimated.

$$\epsilon = k\,\frac{\sqrt{u'^2_{max}}}{U_\infty \cdot \sqrt{N}} = 3\,\frac{\sqrt{(7\,m/s)^2}}{20\,m/s \cdot \sqrt{2000}} = 0.023 \tag{4.2}$$

To close this section, two instances of the results obtained are displayed in Figure 4.6, where the mean dimensionless x velocity (Fig. 4.6a) and the shear Reynolds Stress component (Fig. 4.6b) are displayed for the smooth cylinder case.



(a) $\overline{u}$.

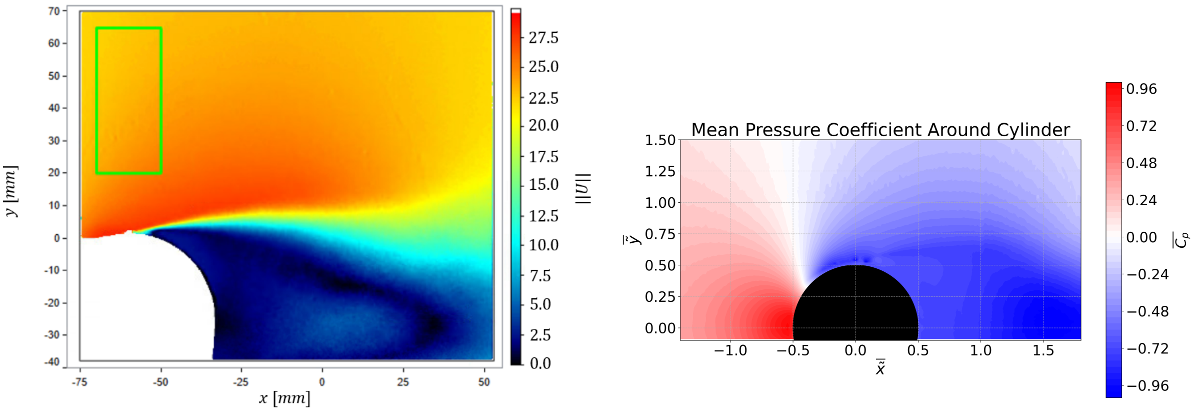
(b) $\overline{u'v'}$.

Figure 4.6: Time-averaged fields for the smooth cylinder with $N = 2000$ samples.

Poisson Solver

With the aim to provide a state-of-the-art PIV-based pressure reconstruction solution that provides a benchmark to assess *PINN* performance, the Poisson Solver built in DaVis 11.0 is used. This solver, based on the principles described in [53], reconstructs the time-averaged pressure field via the resolution of the Poisson pressure equation in the discrete mesh, taking Neumann boundary conditions at the outer edges of the domain.

In addition, in this case a Dirichlet boundary condition for pressure is set to be calculated via the Bernoulli equation of total pressure conservation. Accordingly, the selected regions for the up and downstream datasets comprise a region of $S = 20 \times 45\,mm^2$ outside the main wake and shear layer, where the inviscid flow assumption holds. In fact, Figure 4.7 contains an instance of the region where the boundary condition is set for pressure reconstruction in the downstream dataset. As observed, this is placed away from the cylinder wall and upstream of any shear layer, hence aiming for a region where viscous effects are kept to a minimum.



(a) Contours of velocity magnitude - region for Dirichlet condition in the downstream dataset.

(b) Poisson-based pressure coefficient field around the smooth cylinder.

Figure 4.7: Settings and outputs of the Poisson solver.

Finally, an example of the Poisson-based reconstructed pressure field is displayed in Figure 4.7b. Observing the values obtained, it can be said that the pressure field looks sensible from a theoretical standpoint, provided that a $C_p = 1.0$ is attained at stagnation and the sudden change in C_p right after the suction peak, indicating separation with a static pressure coefficient value close to zero in the recirculation region at the back of the cylinder. However, the error level in the Poisson-based pressure reconstruction will be covered further in-depth in the following chapters, taking as a reference direct pressure measurements and analytical solutions.

Static pressure measurement

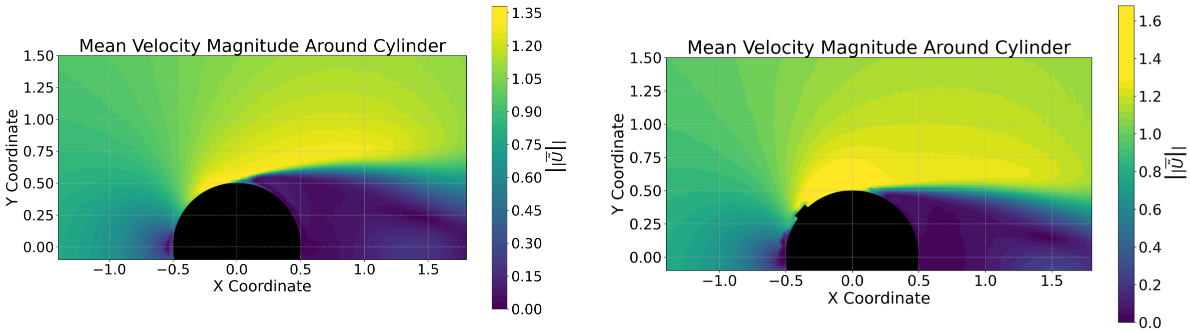
As a closing remark to the present subsection, the specifics of the direct static pressure measurements are briefly exposed. As mentioned in §4.1.2, 16 orifices are arranged throughout the spanwise cylinder symmetry plane, where pressure is measured. Internally, these orifices are assembled with flush-mounted pipes at a $90°$ angle, which have an internal diameter of $0.5\,mm$. In turn, soft tubes with an approximate length of $l \approx 1.2\,m$ connect the pressure taps directly to the differential pressure scanner, which has its reference taps open to the test room.

For simplicity, the measurement process is kept to time-averaged pressures, measured at a frequency of $2\,kHz$ during $T = 120\,s$, ensuring no additional time is needed with respect to the image acquisition. This results in time-averaged pressure values computed from $N = f \times T = 2000\,Hz \times 120\,s = 2.4 \times 10^5$ samples for each of the test cases. Note that, even if pressure measurements are taken simultaneously to the up and downstream image acquisition tests, all values are drawn from the upstream case for simplicity. Finally, since the zig-zag strips partially or totally cover the pressure taps at $\theta = 45°$, its values are not relevant in the cases with zig-zag tape on the model.

4.1.3. Flow field

Prior to analyzing the results provided by the baseline *PINN* algorithm in terms of off-body and surface pressure reconstruction, as well as in overall terms, it is of interest to provide a brief description of the flow field as obtained during the experiments for both cylinder configurations.

In the first place, contours of mean dimensionless velocity magnitude are displayed for the smooth cylinder and zig-zag strips cases in Figures 4.8a and 4.8b respectively. Note that, instead of the nominal freestream speed, a corrected value of $U_\infty = 21.3\,m/s$ is used for normalization, thus accounting for solid and wake blockage in the closed section arrangement. Furthermore, attending to the major differences between the two cases, a delta in maximum velocity magnitude of $\Delta||U|| \approx 0.3\,[-]$ can be located around both suction peaks, which are slightly offset between both cases. This appears to be a consequence of a later flow separation in the case with zig-zag strips fitted, where the detachment point seems to be past the cylinder apex, contrary to the smooth cylinder case. Beyond the separation point, both wakes evolve differently likely due not only to the difference in flow departure angle at separation but also because of the variation in the energy contained in the integral turbulent scales between both cases. Further downstream, this results in a wider wake in the smooth cylinder case, where the total height of the wake appears to be ~ 0.12 diameters taller at the $x/D = 0.5$ station.



(a) Smooth cylinder.

(b) Cylinder with zig-zag strips.

Figure 4.8: Contours of mean dimensionless velocity magnitude - PIV data.

In terms of data quality, several spurious data points can be located close to the cylinder surface in either cases, these likely deriving from local laser reflections and concentrating especially around the stagnation point.

Further supporting evidence for the change in flow topology between the smooth and zig-zag strips cases is provided in the pressure coefficient plot of Figure 4.9, where the represented data is obtained via the static pressure taps on the cylinder surface. As noted, both distributions are matched around stagnation and up to $\theta = 30°$. Beyond this angle, the geometry with zig-zag strips displays a larger favorable pressure gradient, to reach a larger suction peak in magnitude, with an approximate $\Delta C_{p,min} \approx 0.62$, as highlighted in black. The whole picture appears to be dominated by the delay in flow separation of $\Delta\theta_{sep} \approx 15°$, which increases the effective curvature of the flow in the near-wall region, thus causing the commented delta in suction peak magnitude. Note that even if both separation points are identified by the inflection points highlighted in green, the lack of spatial resolution means these are only approximate values.

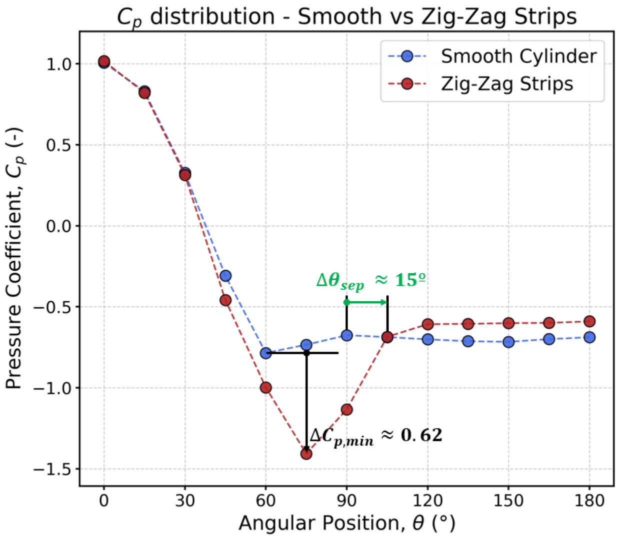
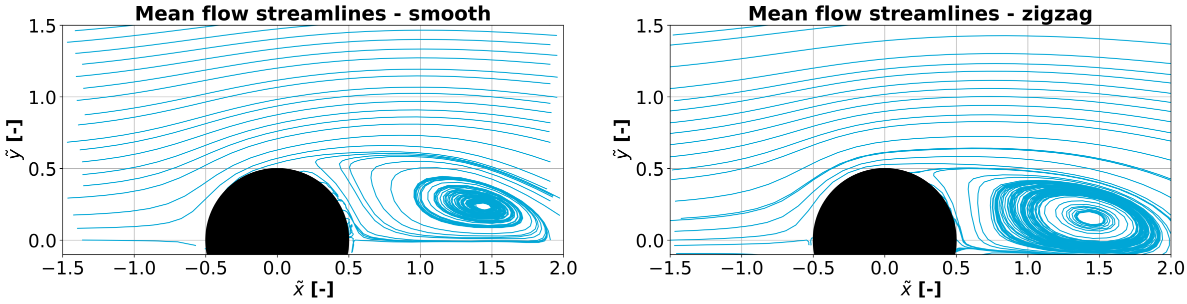


Figure 4.9: Comparison of $\overline{C_p}$ distributions for smooth cylinder and with zig-zag strips @$45°$ - pressure tap data.

Finally, both flow streamline patterns are displayed in Figures 4.10a and 4.10b. Building on the above discussion, one of the central differences lies in the streamline curvature in the $\theta \sim 60° - 90°$ region. As a result, the effective shape of the cylinder changes between cases, particularly in terms of frontal area and local loads, thus impacting the drag coefficient. In accordance, both models appear to differ in the re-circulation region, where a lack of symmetry can be spotted in the case of the cylinder fitted with zig-zag strips.



(a) Smooth cylinder.

(b) Cylinder with zig-zag strips.

Figure 4.10: Flow streamlines from PIV data.

Overall, even if both problems are extremely similar in nature, key differences arise between both flow patterns. Specifically, all observations point towards an adequate effect introduced by the zig-zag strips, as they seem to promote earlier boundary layer transition, effectively delaying separation and thus changing the flow field and local loads on the cylinder surface. Accordingly, it is concluded that hints on the ability of *PINNs* to reconstruct surface pressure in unequal cases can be provided.

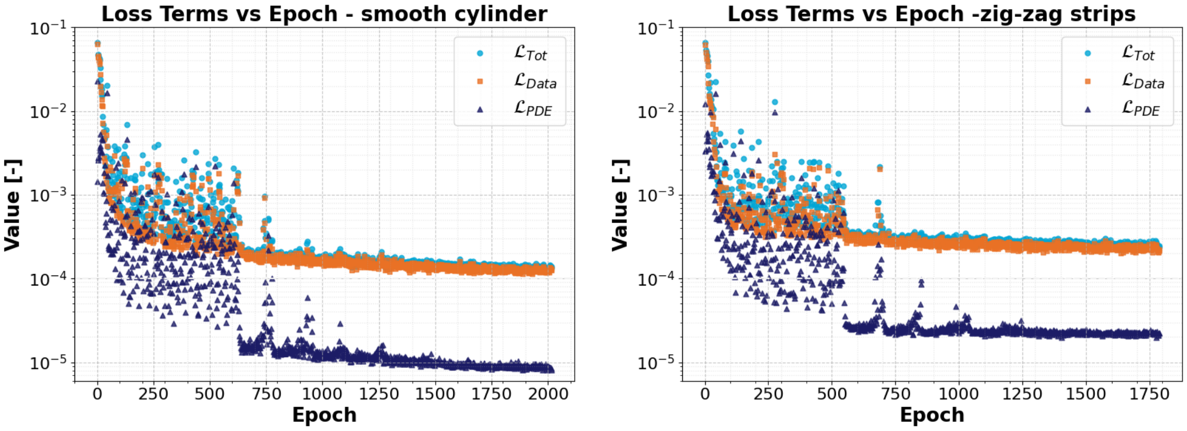
4.2. PINNs vs Poisson Solver (state of the art) - baseline comparison

Once a base idea of the flow field that describes both cases has been provided, an analysis of the baseline *PINN* performance is given for both flow fields. In order to assess the accuracy of the *PINN* models, reference pressure values from pressure tap measurements on the surface are used. Besides, an in-depth comparison to the pressure field provided by the Poisson solver is included, this representing the current state-of-the-art pressure reconstruction technique.

4.2.1. Numerical Results

Following an analogous structure to our analysis in Chapter 3, an overview of the key performance indicators are given for both converged models.

Thus, in the first place, Figure 4.11 is referenced, which shows the evolution of the two main loss components, as well as that of the total loss, for both the smooth (Fig. 4.11a) and zig-zag strips (Fig. 4.11b) cases. As observed, both plots display similar learning dynamics, with hints of an excess learning rate in the initial iterations leading to a noisy response up to $epoch \approx 600$. In terms of absolute values, it appears that the smooth cylinder case is able to find a lower optimum, with signs of optimization persisting even at $it = 20000$, contrary to the premature stop in the zig-zag strips case due to no optimization for over 500 epochs.



(a) Smooth cylinder.

(b) Cylinder with zig-zag strips.

Figure 4.11: Evolution of main loss components during training.

In addition, Table 4.1 includes the loss values for both models after convergence. Visibly, the delta when evaluating the loss at the full batch is unequal for both models, with the zig-zag model incurring twice the PDE loss, with $\mathcal{L}_{PDE} = 4.2 \times 10^{-5}$, which represents at least one order of magnitude less than the data loss for either case. Accordingly, the total loss is only marginally higher than the data loss, resulting in converged values of $\mathcal{L}_{tot} = 6.7 \times 10^{-4}$ and $\mathcal{L}_{tot} = 9.1 \times 10^{-4}$ for the zig-zag and smooth cases respectively.

Furthermore, it is of interest to observe what the distribution of errors with respect to the reference data is among the different variables. These values are included in Tables 4.2 and 4.3, which feature the MSE for each of the labeled variables, as well as the surface pressure MSE when compared to the pressure tap data. Note that, in addition to the *PINN* results, the surface C_p reconstruction errors incurred by the Poisson solver are attached too.

Table 4.1: Loss components for *PINN* baseline - smooth and zig-zag strips cases.

Model	$\mathcal{L}_{\text{Total}}$	$\mathcal{L}_{\text{Data}}$	$\mathcal{L}_{\text{PDE}}$	$\mathcal{L}_{\text{Prior}}$	$\mathcal{L}_{\text{p, BC}}$
Smooth	9.08×10^{-4}	8.87×10^{-4}	2.11×10^{-5}	5.83×10^{-8}	1.25×10^{-8}
Zig-zag	6.71×10^{-4}	6.29×10^{-4}	4.16×10^{-5}	6.28×10^{-8}	1.25×10^{-8}

As may be witnessed, both cases show different responses, particularly in the x-velocity error level. While this is kept to a similar value to the y-velocity MSE in the zig-zag strips case, with $MSE_{\bar{\tilde{u}}} \approx MSE_{\bar{\tilde{v}}} = 2.9 \times 19^{-4}$, an unbalanced response is seen in the smooth cylinder case, with $MSE_{\bar{\tilde{u}}} = 7.1 \times 10^{-4}$. While this is inconclusive proof, the disparate response across cases possibly reflects a mismatch between the optimal solutions in terms of PDE and data losses.

Table 4.2: Mean squared error per predicted variable - flow around smooth cylinder.

Source	$\bar{\tilde{\boldsymbol{u}}}$	$\bar{\tilde{\boldsymbol{v}}}$	$\overline{\widetilde{\boldsymbol{u'u'}}}$	$\overline{\widetilde{\boldsymbol{u'v'}}}$	$\overline{\widetilde{\boldsymbol{v'v'}}}$	$\bar{\tilde{\boldsymbol{p}}}$
PINN	7.13×10^{-4}	1.39×10^{-4}	1.24×10^{-5}	2.05×10^{-5}	2.87×10^{-6}	4.63×10^{-3}
Poisson	–	–	–	–	–	4.67×10^{-3}

Table 4.3: Mean squared error per predicted variable- cylinder with zig-zag strips.

Source	$\bar{\tilde{\boldsymbol{u}}}$	$\bar{\tilde{\boldsymbol{v}}}$	$\overline{\widetilde{\boldsymbol{u'u'}}}$	$\overline{\widetilde{\boldsymbol{u'v'}}}$	$\overline{\widetilde{\boldsymbol{v'v'}}}$	$\bar{\tilde{\boldsymbol{p}}}$
PINN	2.93×10^{-4}	2.90×10^{-4}	1.68×10^{-5}	2.03×10^{-5}	9.27×10^{-6}	3.54×10^{-3}
Poisson	–	–	–	–	–	4.50×10^{-3}

In addition to the above, certain insight is already given in the last column on the comparative assessment of both pressure reconstruction algorithms, namely *PINN* and the Poisson solver, when compared to pressure tap data on the surface. Interestingly, both methods show comparable results in the smooth case, with approximately $MSE_{\bar{\tilde{p}}} \approx 4.6 \times 10^{-3}$. On the other hand, a -25% reduction in error is seen for the *PINN* prediction with respect to the Poisson solver in the case with zig-zag strips fitted. As a closing note, it is possible to realize that the values of surface MSE are at least an order of magnitude larger than the rest. Presumably, this is not only due to the fact that pressure is the hidden state, but also due to a larger concentration of gradients close to the cylinder walls posing a bigger challenge in terms of reconstruction accuracy, as $\bar{\tilde{p}}$ is only evaluated at the surface.

Table 4.4: PDE residuals at convergence.

Model	$\mathcal{R}_{\text{continuity}}$	$\mathcal{R}_{\text{x-momentum}}$	$\mathcal{R}_{\text{y-momentum}}$
Smooth	1.15×10^{-5}	5.87×10^{-6}	3.72×10^{-6}
Zig-zag strips	2.88×10^{-5}	7.22×10^{-6}	5.61×10^{-6}

Finally, the values of the residuals at convergence are enclosed in Table 4.4. While these do not appear to offer additional information other than the previous observation on the larger $\mathcal{L}_{PDE}$ in the zig-zag strips case, it can now be stated that the increment in loss does not appear to be biased towards any of the three residuals.

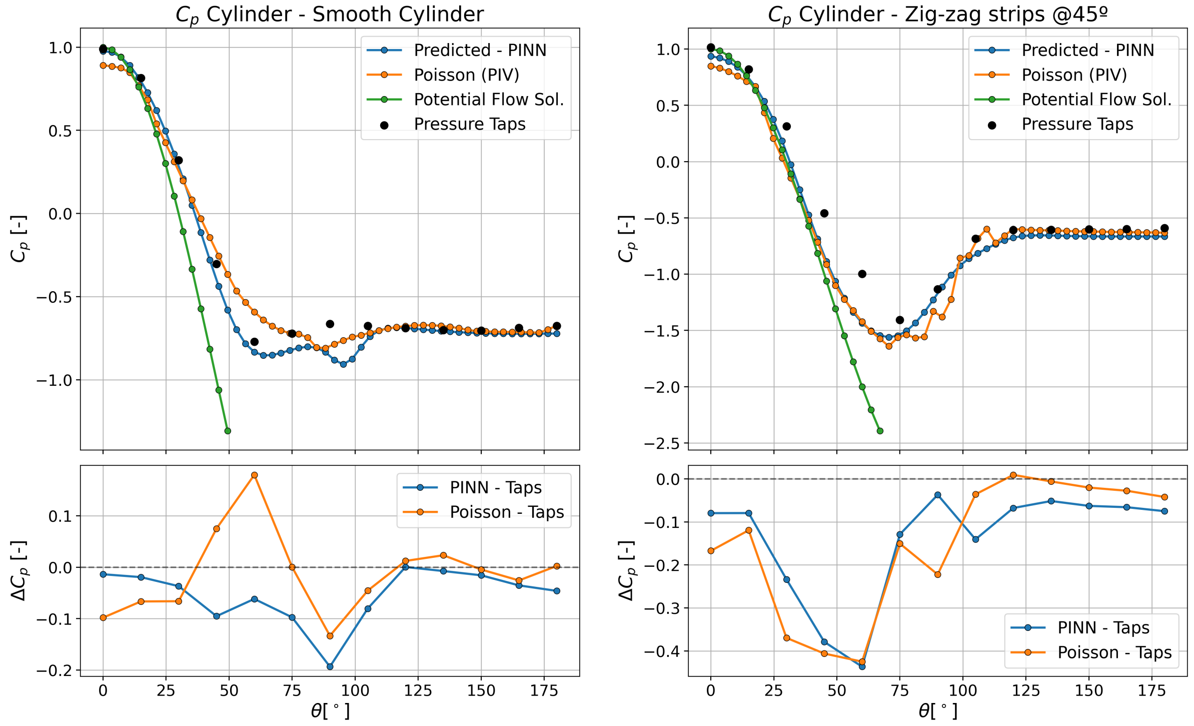
4.2.2. Spatial distribution of errors

Moving on to the distribution of reconstruction errors in space, the pressure coefficient distributions of both case studies are reflected in Figure 4.12, where the time-averaged pressure tap data and the

Poisson-based solution are included. Moreover, the potential flow solution for the flow around a two-dimensional cylinder is plotted too, providing a reference for what the inviscid and irrotational solution would be.

Elaborating on the smooth case results of Figure 4.12a first, an initial deviation at $\theta = 0°$ is incurred by the Poisson solver, under-predicting C_p at stagnation by $\Delta C_p = 0.1$. While this difference may derive from data, it requires further investigation. Advancing to higher angles of up to $\theta = 90°$, all three sources of data predict milder favorable pressure gradients, reflecting non-negligible viscous effects that deem the theoretical model useless for $\theta \geq 30°$.

In terms of overall suction peak prediction, while the *PINN* slightly over-predicts it by $\Delta C_{p,min} = -0.07$, the Poisson solution again provides a milder peak value, incurring an error equal to $\Delta C_{p,min} = +0.19$. Besides, distinct behaviors take place in the $\theta \in [75°, 105°]$ range. First, the pressure taps follow a regular pressure distribution with separation taking place possibly around $\theta = 90°$ as per the inflection point. In contrast, both the *PINN* and Poisson-reconstructed fields show signs of local suction at the cylinder apogee. While this could be seen as a trace of a re-circulation bubble, this seems unlikely and thus requires further investigation not only due to the lack of re-attachment, but also because the C_p is far from remaining constant, as it forms a local suction peak.



(a) Smooth cylinder.

(b) Cylinder with zig-zag strips.

Figure 4.12: Pressure coefficient distribution on the cylinder surface.

Shifting to the zig-zag strips case, a similar trace is found at $\theta = 0°$, though now it affects both the Poisson model, with $\Delta C_p = -0.16$, and the *PINN* solution to a lesser extent, with $\Delta C_p = -0.08$. Focusing on the favorable pressure gradient region, there appears to be a mismatch between the pressure tap data and the rest of information sources. Interestingly, both Poisson and *PINN* distributions appear to follow the potential flow solution and match each other closely at the $\theta = \{45°, 60°\}$ stations. It is noted that this area can be deemed of less reliability with regards to pressure tap information, as the zig-zag strips are placed at around $\theta = 45°$ and their $4\,mm$ width causes them to extend over an angle of $\Delta\theta \approx 10°$. Accordingly, it can be assumed that the neighboring taps, namely the $\theta = 30°$ and $\theta = 60°$, may be affected by local instances of back-pressuring or re-attachment at the near-wall scale.

Moreover, with regards to predicting the magnitude of the suction peak location, both algorithms fall within $\Delta\theta_{C_p,min} \leq 15°$, the Poisson solver representing the least accurate with an additional $\Delta C_p = 0.02$. Further downstream, the main delta between both reconstruction models appears to be a smoother transition into separation of the *PINN* solution, as compared to the erratic C_p response seen in Poisson.

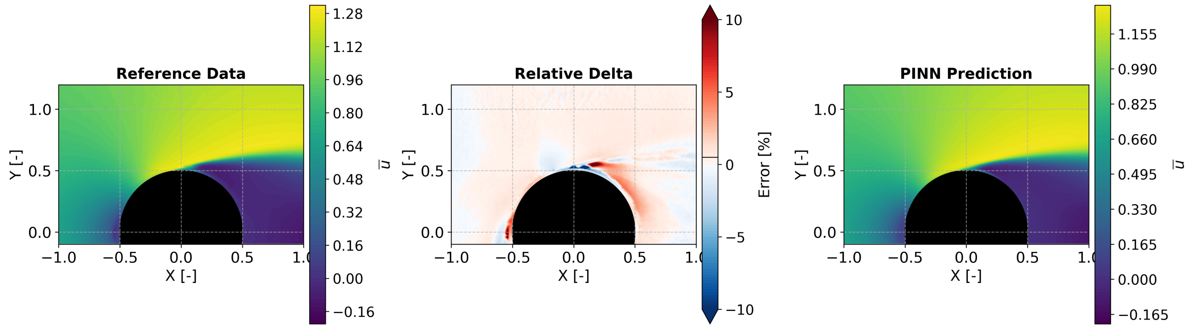


Figure 4.13: Contours of relative $\overline{u}$ delta with respect to smooth cylinder PIV - $\frac{X_{PINN}-X_{Ref}}{Y_\infty}$

With the aim to deepen the understanding on the modes of error for both models and identify what the root causes are, a deeper investigation is presented. Firstly, Figure 4.13 contains the relative delta of mean $\tilde{u}$ around the cylinder Note that this delta reflects the deviation of the *PINN* output with respect to the reference data, as a fraction of the free-stream values, analogous to §3.2.2.

In the first place, one of the main deviations takes place close to stagnation, at $\{x, y\} = \{-0.5, 0\}$, where the reference data includes spots of low velocity that the *PINN* effectively smoothens out. Indeed, this trace in the raw data appears to be an artifact of reflections close to the cylinder boundary, as per the representation in Figure 4.14, where the evolution of $\overline{\tilde{u}}$ and $\overline{C_p}$ are shown along the stagnation streamline. In this plot, it can be seen that the reference x-velocity distribution contains invalid data in the range $x \in [-0.56, -0.5]$, hence reflecting the ability of *PINNs* to filter out noisy data. Besides, coupling it with the pressure prediction, one can conclude that the error around stagnation observed in the Poisson solution for the smooth cylinder is induced by the propagation of noise from the source data into the hidden state.

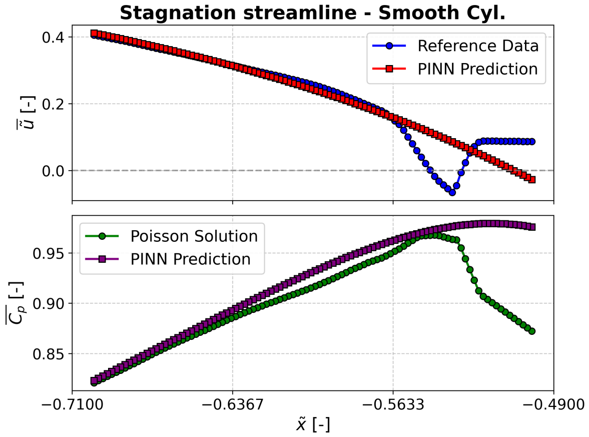
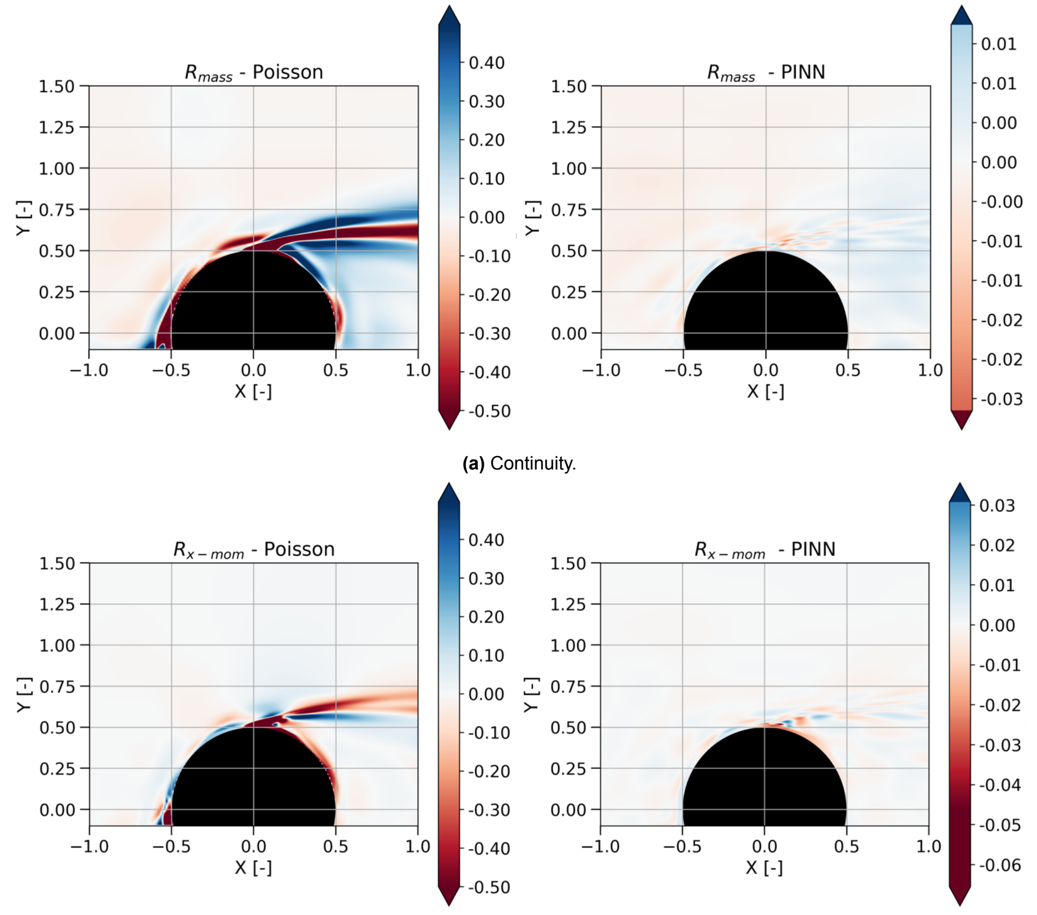


Figure 4.14: Comparison of $\overline{\tilde{u}}$ and $\overline{C_p}$ along stagnation streamline - smooth cylinder

Secondly, the region around the $\theta \in [90° - 100°]$ concentrates the biggest deviation between the *PINN*-predicted and reference velocity fields, with local errors of $|\varepsilon| \geq 10\%$. Similarly, inspection of the region in the absolute reference data contours showcase a lack of smoothness at the start of the shear layer. However, this region is particularly susceptible to time-averaging-induced error, as the local separation makes it a strong point of unsteadiness, thus making it challenging to classify it as an un-physical behavior.

Since this particular region matches the small secondary suction peak observed in the surface pressure distribution, further insight is searched via the analysis of the continuity and x-momentum residuals attained by both models, which are displayed in Figures 4.15a and 4.15b respectively. Note that the Poisson-based residuals are obtained by training the same NN architecture as the *PINN*, but removing the PDE loss and learning the pressure field directly from the Poisson solution, as if it were another term in the data loss function. Additionally, the colorbar limits for both models are different, with those of the Poisson solver being one order of magnitude above.

From the fields shown, one can conclude that the near-wall region and the wake edge concentrate the majority of non-zero residuals in the Poisson case, while a virtually constant distribution is seen for the *PINN*-optimized solution. Even if the cause for regions of larger x-momentum residuals can be challenging to decipher, there are limited reasons as to why the conservation of mass should not hold locally. While at stagnation the source of error might be due to the local noise highlighted earlier, the region around the separation point and wake edge could indeed be areas where three-dimensional effects abound due to the presence of larger coherent turbulent structures. Accordingly, one of the strongest hypothesis to explain the local error seen by the *PINN* and Poisson solvers could be due to the two-dimensional simplification of the flow-field.

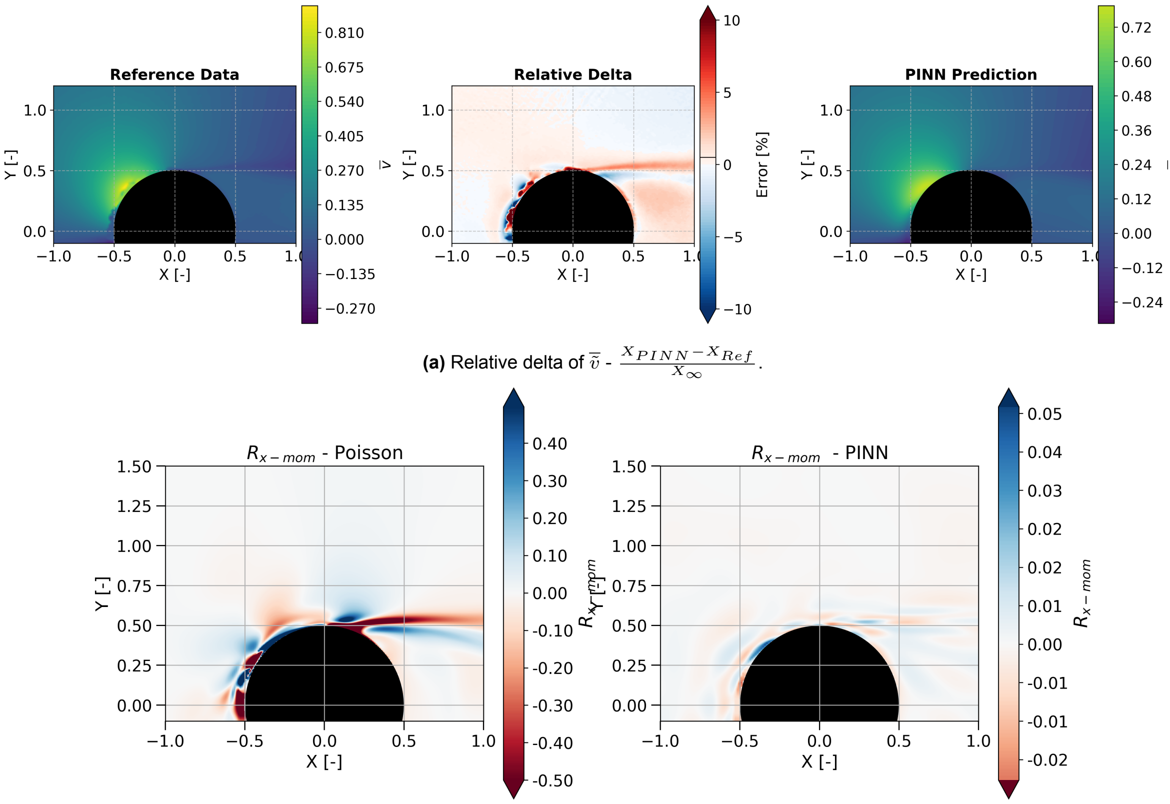


(a) Continuity.

(b) X-momentum.

Figure 4.15: Contours of residuals for smooth cylinder - PIV with Poisson (left) and PINN(right).

Performing a similar analysis on the zig-zag strips case, it appears that the same trace in y-velocity error shows around stagnation and beyond in Figure 4.16a, potentially deriving from reflections off the cylinder surface. Yet again, this behavior is suppressed in the *PINN* output, showing consistency in noise robustness across cases. Furthermore, despite the x-momentum residuals in Figure 4.16b are of the same order of magnitude to the smooth case, their optimization does not seem to introduce noticeable discrepancies between the Poisson and *PINN*-based pressure fields.



(a) Relative delta of $\overline{v}$ - $\frac{X_{PINN} - X_{Ref}}{X_\infty}$.

(b) Residual of x-momentum - PIV with Poisson (left) and PINN(right).

Figure 4.16: Contours of zig-zag strips case.

Overall, it appears that, while both models have comparable responses, in agreement with the absolute surface $MSE_{\overline{p}}$ values of Tables 4.2 and 4.3, their modes of failure are distinct. On the one hand, Poisson consistently fails to achieve the correct stagnation pressure through propagation of errors from the PIV data. Additionally, the fact that both models rely on two-dimensional formulations of the conservation equations to reconstruct the pressure field appears to cause deviations from the tap values. In general, it appears that the noise robustness of *PINNs* makes it more consistent across cases.

4.3. PINNs Sensitivity studies

After studying how the baseline *PINN* algorithm performs and comparing to state of the art techniques for pressure reconstruction from PIV data, the attention is shifted to understanding how various *PINN* parameters affect its accuracy. To this end, an approach similar to that in Chapter 3 is carried out, probing sensitivities to the NN shape, the number of data and collocation points, the relative weighting between losses and the effect of enforcing physical boundary conditions at the surface, especially when there are data gaps.

Nonetheless, an additional test is carried out to understand the sensitivity to incorporating pressure tap data as a labeled variable during the NN training process, thus providing reference data for the hidden state at the surface.

Finally, it is noted that all tests in the present section are carried out with a reduced data version of the smooth cylinder dataset, where the domain has been cropped to include the rectangle enclosed in $\{x/D, y/D\} \in [-1:1, -0.1:1]$. This is done to aid in training efficiency and overall RAM usage, reducing the total amount of data and collocation points to $N_{meas} \approx 100.000$ while keeping the same point density close to the surface.

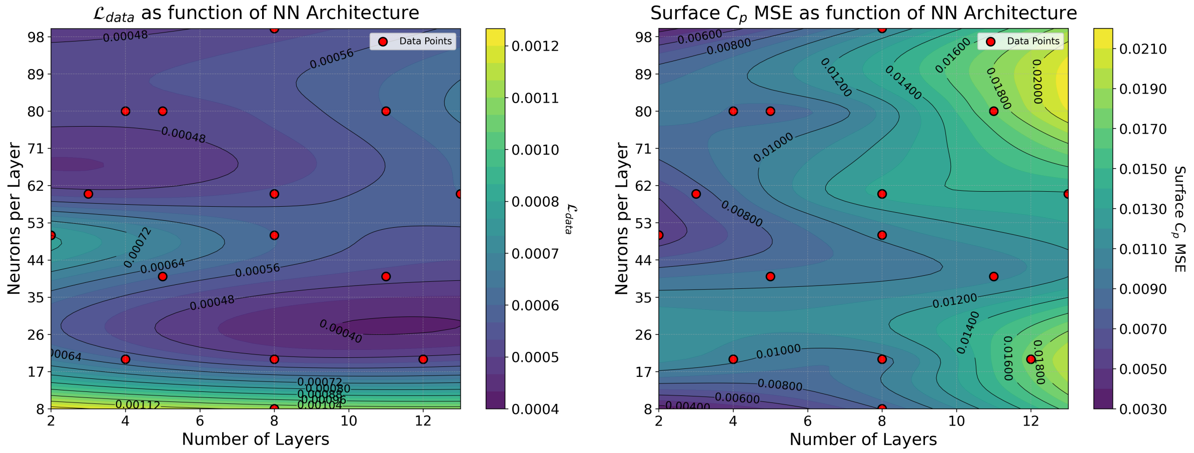
4.3.1. NN shape

Looking at the effect of changing the neural network architecture first, an initial test matrix was laid out to replicate the same ratios of learnable parameters to data points as in the previous chapter, resulting in the test models from $\#1$ through $\#9$. However, given the little sensitivity observed, which is discussed next, additional points are added to fill the testing space, particularly towards thinner and shallower networks.

Table 4.5: NN Shape/Size study

Test #	L		N		$\boldsymbol{N_{params}}$
	Level	**Value**	**Level**	**Value**	
1	-1	5	-1	40	8566
2	-1	5	1	80	33 126
3	1	11	-1	40	18 406
4	1	11	1	80	72 006
5	0	8	0	60	29 826
6	1.4	13	0	60	48 126
7	-1.4	3	0	60	11 526
8	0	8	1.4	20	3546
9	0	8	-1.4	100	81 706
10	–	2	–	50	5556
11	–	8	–	8	654
12	–	12	–	20	5226
13	–	4	–	20	1866
14	–	4	–	80	26 646
15	–	8	–	50	20 856

Proceeding with the results of the test matrix, contours of data loss and pressure reconstruction error at the surface are displayed in Figures 4.17a and 4.17b respectively. Interestingly, both key performance indicators display substantially unequal sensitivities to the number of layers or neurons. Firstly, the response of $\mathcal{L}_{data}$ is overall similar in trends to the observations made on CFD data, penalizing the models where either the N or L parameters take extreme values at the lower end.



(a) $\mathcal{L}_{Data}$.

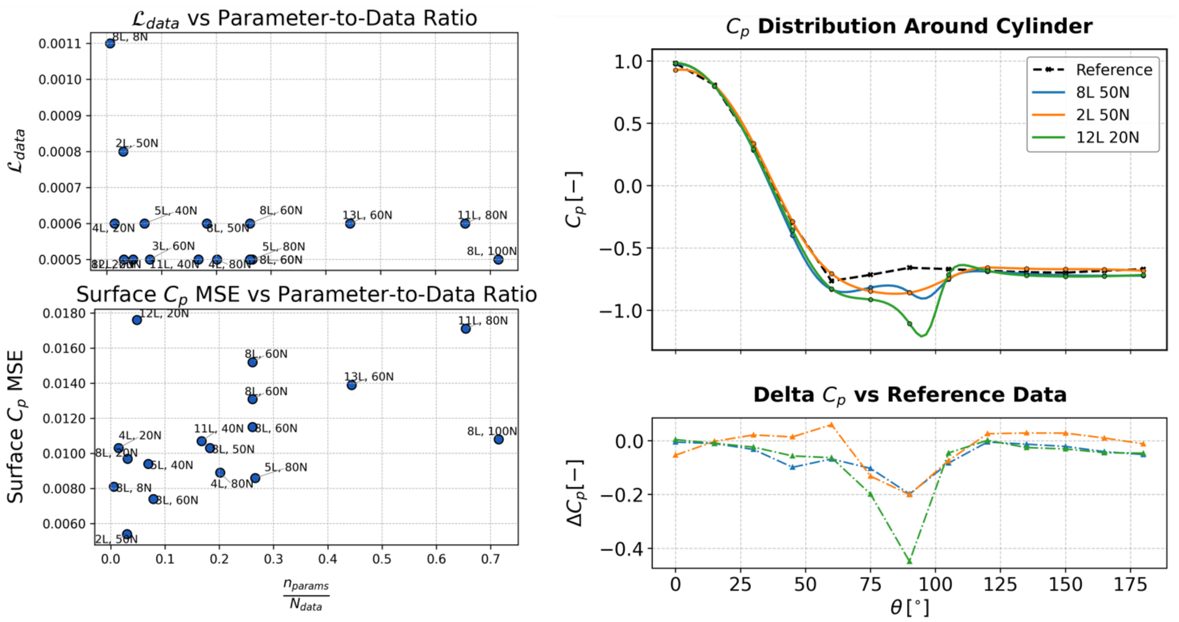
(b) .

Figure 4.17: Contours of loss components with NN shape parameters.

Conversely, the $MSE_{\overline{C_p}}$ contours show only a major trend, where deeper models incur the largest pressure reconstruction errors at the surface. Presumably, it could be that overly complex models that can accommodate spatial variations of higher order attempt to fine-tune data reconstruction, hence reproducing part of the data noise. This hypothesis not only builds on the contours below, but also on the observations made earlier with respect to the ratio of the data and PDE losses at convergence, which reflect that $\mathcal{L}_{data}$ optimization is the main limiting factor to reduce the overall loss and thus any additional degrees of freedom would be directed towards its minimization.

One of the key takeaways from the analogous study in §3.3.3 was the dependency of performance indicators to the ratio of learnable parameters and total number of data points. Performing an analogous analysis on the experimental dataset leads to the scatter plots in Figure 4.18a, where the evolution of the data loss and the $MSE_{\overline{C_p}}$ at the surface are shown with respect to that particular ratio. Assessing the top figure first, one cannot really discern a particular trend, other than the two extreme models provide the least ability to reconstruct the labeled variables, leading to values of $\mathcal{L}_{data} = 1.1 \times 10^{-3}$ and $\mathcal{L}_{data} = 8 \times 10^{-4}$ for the thinnest and shallowest models respectively. Alternatively, the rest of models appear to provide comparable reconstruction errors, with $\mathcal{L}_{data} \in [5, 6] \times 10^{-3}$.

On the other hand, the lower graph depicts the error when reconstructing the hidden state at the surface, when compared with pressure tap data. Interestingly, the two simplest models are among the three most accurate, achieving an error as low as $MSE_{\overline{C_p}} = 5.7 \times 10^{-3}$ with the $2\,L, 50\,N$ architecture. However, it must be noticed that the level of dispersion due to the stochasticity in the training process is only half an order of magnitude lower than the entire range of errors, given that $|\mathcal{L}_{max} - \mathcal{L}_{min}|_{8L,60N} = 4 \times 10^{-3}$. This signals that, if present, the dependencies are certainly more muted than in the CFD investigation, where all KPIs showed the same consistent trend of reduced error with increased model complexity.



(a) $\mathcal{L}_{data}$ and surface MSE_{C_p} vs learnable parameters.

(b) Surface C_p distribution for different NN shape models.

Figure 4.18: Results of NN shape study.

As a final source of information to identify the underlying trend, Figure 4.18b includes the absolute and delta $\overline{C_p}$ distributions on the surface for three different NN architectures. On this occasion, the main remark concerns the $\theta \in \{60^\circ, 105^\circ\}$ region, where instances of data noise or unsteadiness have been identified in the source data. Visibly, the shallow model with $2\,L$, in orange, follows a smoothed out version of the distribution provided by the baseline architecture, in blue. This allows the simpler model to achieve an overall better reconstruction, which would be accentuated should a finer spatial resolution of taps be present, given the oscillations present in the baseline. Conversely, the deep model with $12\,L$ appears to worsen such oscillatory response, leading to local errors as large as $\Delta\overline{C_p} = 0.045$ at $\theta = 90^\circ$.

While the trends are not as conclusive as those observed in CFD, a main hypothesis is built that justifies the differences present in the experimental case. This theory assumes that more complex architectures, especially deeper ones, can accommodate a wider space of function reconstructions thanks to the nesting of both non-linear and linear operators. Accordingly, deeper models potentially devote the additional degrees of freedom to minimize the data loss further, which appears to be the limiting factor for overall loss minimization in the baseline *PINN* model. As a result, deeper models appear to display more oscillatory pressure distributions at the surface, incurring substantial reconstruction errors when compared to the pressure tap data.

4.3.2. Data loss weighting

Building on the same discussion, an interesting test to carry out is to perturb the data loss weighting, which can provide further insight into how its optimization affects the surface pressure reconstruction accuracy. Supposedly, should a further minimization of the data loss lead to a similar failure mode as the deep models, it would strengthen the hypothesis.

In order to produce this information, an analogous test as the one carried out in §3.3.2 is executed, probing both a static weighting on $\mathcal{L}_{data}$ ($\alpha = \{0.01, 0.1, 1, 10, 100, 1000\}$) and a dynamic weighting scheme, β on $\mathcal{L}_{PDE}$.

To begin with, Figure 4.19 depicts the evolution of the data, PDE and total losses (both weighted and un-weighted) versus the data loss weighting. As witnessed, the PDE loss in red follows a direct and virtually constant relation with α. Just as was concluded in Chapter 3, this is to be expected, given that its contribution to the total loss is inversely proportional to α.

Of bigger interest, however is the trend observed in the unweighted total loss, as well as in $\mathcal{L}_{data}$. For downscaling data weights, namely $\alpha < 1$, the optimization remains stable around $\mathcal{L}_{data} \approx 10^{-3}$, while biasing the weights towards the minimization of $\mathcal{L}_{data}$ worsens the actual, unweighted, loss function, as seen for $\alpha \geq 10$.

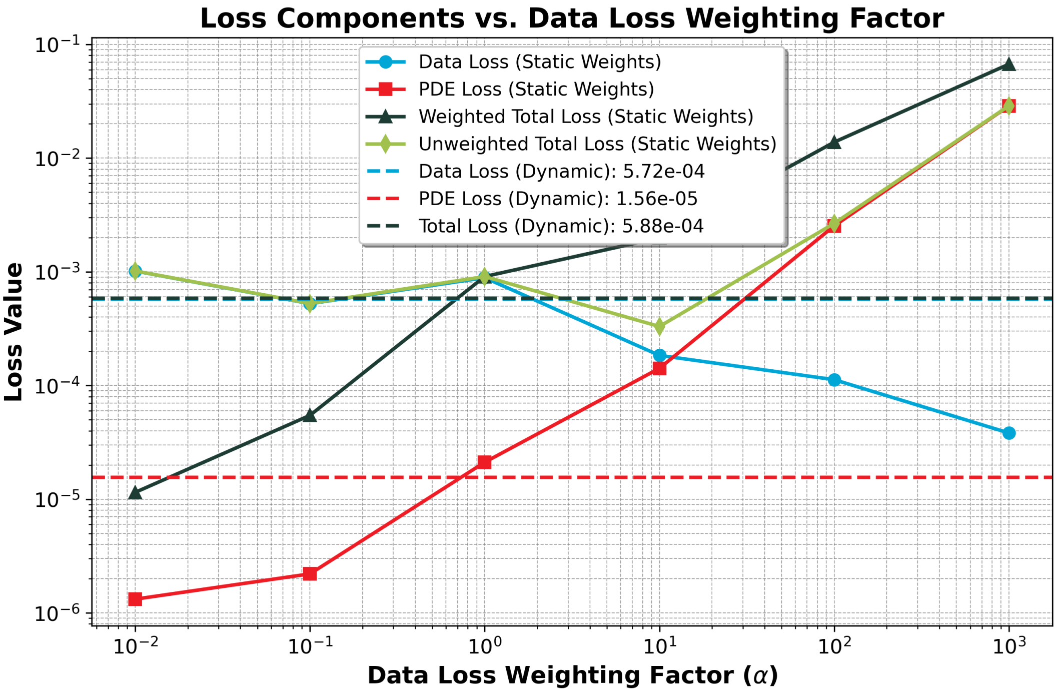


Figure 4.19: Loss components as a function of data loss weighting.

A possible interpretation for this is that an optimum value for the data loss weight lies in the range $\alpha \in [1, 10]$, where all relevant features in the data are learned. This would explain that, when $\alpha < 1$, a further PDE loss minimization does not violate those key features in the data. On the other hand, the consequence of this optimum for $\alpha \geq 10$ is that, even if the data loss is further optimized, the rate at which it decays, namely $\partial\mathcal{L}_{data}/\partial\alpha$ is smaller than the rate at which the PDE loss grows, namely $\partial\mathcal{L}_{PDE}/\partial\alpha$. The interpretation is that any efforts of the NN to learn the data better than the level for $\alpha = 10$ require for the N-S equations to be unfulfilled to a bigger extent in absolute terms.

Moving on to how these behaviors impact the accuracy of pressure reconstruction at the surface, Figure 4.20 shows the $\overline{C_p}$ distributions for the $\alpha = \{0.01, 1, 100\}$ cases, as well as that of the dynamic loss case. Similarly to the observations made in the CFD case, biasing the loss towards minimizing the N-S equation residuals leads to an overly smooth pressure distribution, incurring errors as large as $\Delta\overline{C_p} = -0.26$ at stagnation or $\Delta\overline{C_p} = +0.23$ at the suction peak. On the contrary, a different behavior to the study with CFD data is seen for the $\alpha = 100$ case, which in this case reflects oscillations in the $\theta \in \{75°, 120°\}$ region, leading to errors of $\Delta\overline{C_p} = -0.36$ when compared to pressure tap data. Finally, the dynamic model offers similar results to those of the unitary weighting but reducing oscillations further still, which aligns with the close loss values between both models, as per Figure 4.19.

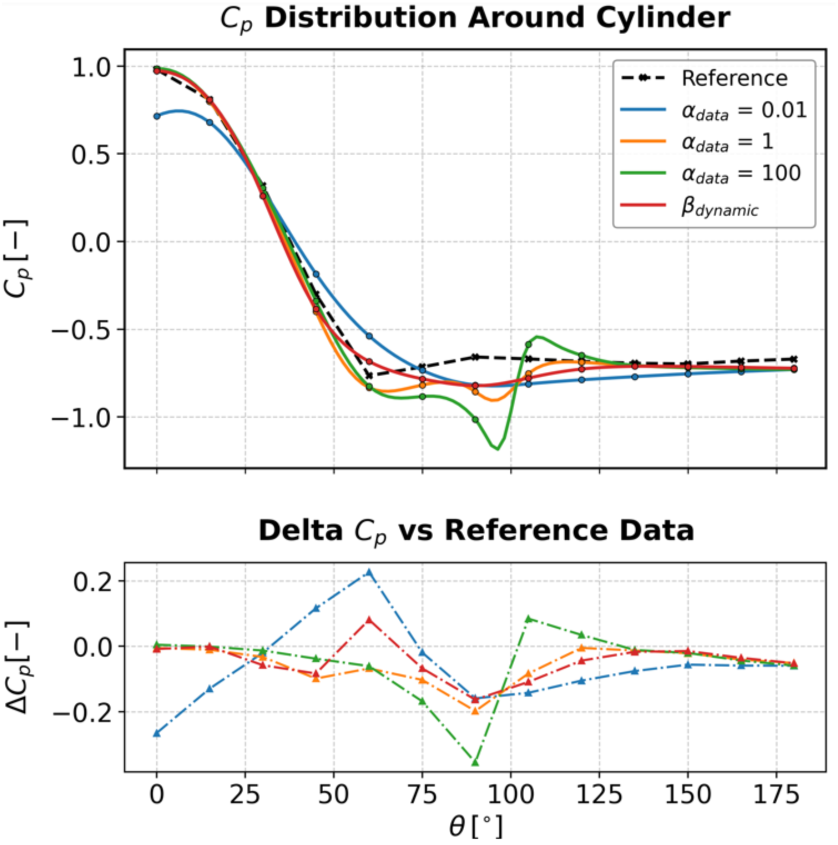


Figure 4.20: Surface C_p distribution for various $\mathcal{L}_{data}$ loss weights.

As a general outline, it appears that the results obtained in this study support the hypothesis discussed earlier. Specifically, steps that result in optimizations of the data loss function beyond baseline levels lead to increased surface pressure reconstruction errors. This is a by-product of the baseline *PINN* framework capturing the relevant data features while achieving a balance between the data and N-S losses such that a physical meaning is ensured. Hence, attempts to learn data further only result in deviations from the local optimum achieved for the PDE loss minimization, which seems to provide the most accurate pressure reconstruction at the surface.

4.3.3. Data and collocation points

Following with the investigation to understand whether the various aspects of *PINNs* equally affect performance on CFD and experimental data, an analogous study is carried out to to the one in §3.3.3, seeking to measure the effect of the number of measurement and collocation points on pressure reconstruction. To avoid repetition, the reader is referenced to Table 3.9, where the tested ratios of both types of points to the total number of data locations are displayed.

Initially, Figure 4.21 is referenced, where the derivatives of the MSE of each variable with respect to the number of measurement and data points are represented as red and blue bars respectively. Note that, analogously to the previous chapter, the derivatives to either type of points are computed by keeping the opposite number of points constant, thus isolating the effect. Moreover, in this case, the reference is set to be the PIV data, except for the pressure field, which is only measured at the pressure tap locations to be compared with the corresponding data.

Attending to the various values, the derivatives of the x-velocity and pressure reconstruction errors stand out when compared to those of the rest of variables, which are at least two orders of magnitude smaller in absolute terms. While the dispersion of results is significantly larger in magnitude that the averages themselves, there appear to be unequal responses for both variables. In the case of the x-velocity, the addition of both types of points appears to result in an increase in the reconstruction error. On the other hand, the accuracy of the surface pressure field appears to benefit from the addition of collocation points, while raising the number of measurement points during training seems to worsen the reconstruction.

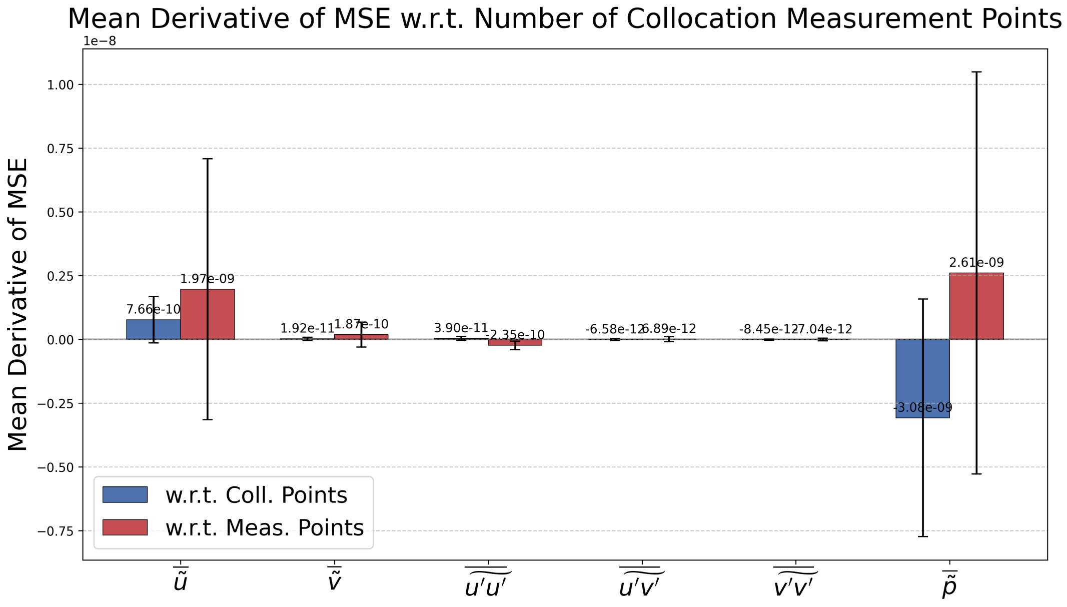
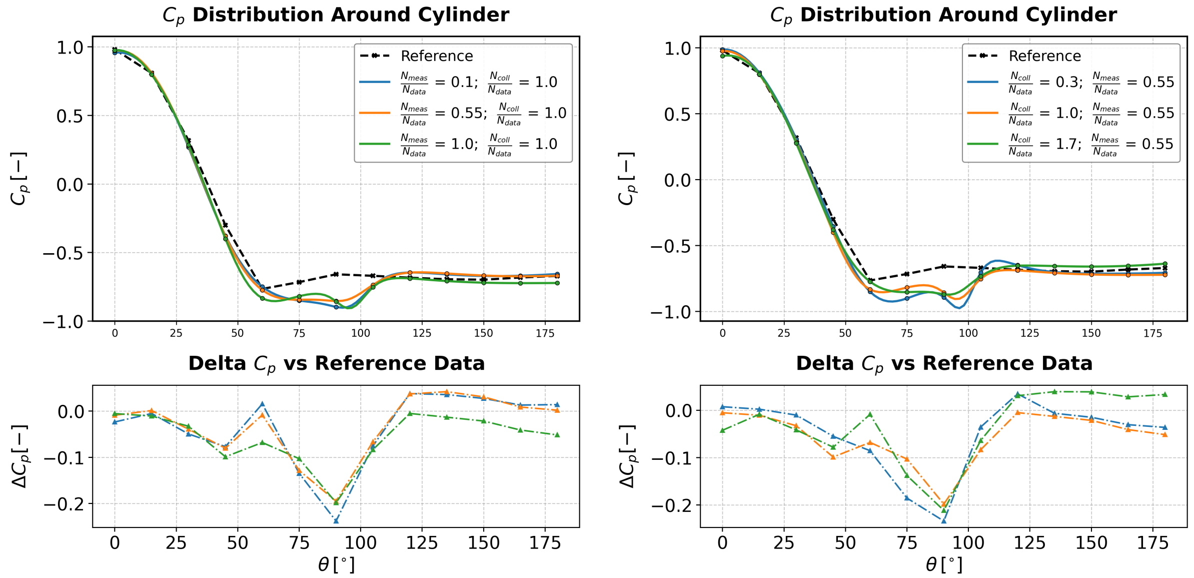


Figure 4.21: Mean variation of MSE for each variable with collocation and measurement points.

In order to better understand the latter trend, which is notably different from the observations in the CFD case study, Figures 4.22a and 4.22b next respectively show the surface pressure distributions for various numbers of measurement and of collocation points.

Visibly, the main region affected by both parameters is that contained in $\theta \in [60°, 105°]$, where the concentration of error with respect to the pressure tap data takes place too. In this regard, data suggests that the model trained with all data points, namely $N_{meas}/N_{data} = 1.0$, produces the largest oscillations, thus producing a positive derivative of the error with respect to the number of measurement points. Alternatively, the opposite behavior is reflected in the plot to the right, where augmenting the quantity of PDE evaluation points reduces the oscillations in the pressure distribution, and hence the reconstruction error.



(a) Dependency to number of measurement points.

(b) Dependency to number of collocation points.

Figure 4.22: Effect of measurement and collocation points on the surface pressure distribution.

Even though these observations significantly differ from the results obtained in §3.3.3, a possible justification can be given along the same lines as the previous sections. Potentially, these findings support the idea that the main deviations observed with respect to the pressure tap data derive from local noise in the PIV data close to the cylinder boundary. Accordingly, removing data points could reduce the number of noisy data locations, hence lessening the constraint that forces the reproduction of this erroneous data. Similarly, increasing the density of collocation points helps to add local regularization in areas affected by noise, thus improving the physical meaning of data.

Note, however, that this appears to hold only because the data point density is sufficiently high that the removal of up to 90% of these labeled data points still contains sufficient information about the flow to permit an accurate reconstruction of it. In fact, based on the observations in Chapter 3, a lower limit would exist that removes key flow features, causing more harm through data scarcity than good through noise reduction.

4.3.4. Boundary Condition Loss

Building on the sensitivity that was observed to the addition of physical laws on the surface when operating the *PINN* on CFD data, an analogous test is carried out on the experimental dataset of the smooth cylinder. Accordingly, the effect of enforcing the no-slip, no through-flow and no-fluctuations boundary conditions on the cylinder surface are studied - both for the original dataset and with artificial data gaps, removing the labeled data from the cylinder surface to the pre-specified cropping radius. It is worth mentioning that, given the lack of sensitivity observed in §3.3.4 to the collocation point generation strategy, the original PIV data locations are kept to evaluate the PDE loss when the data is cropped.

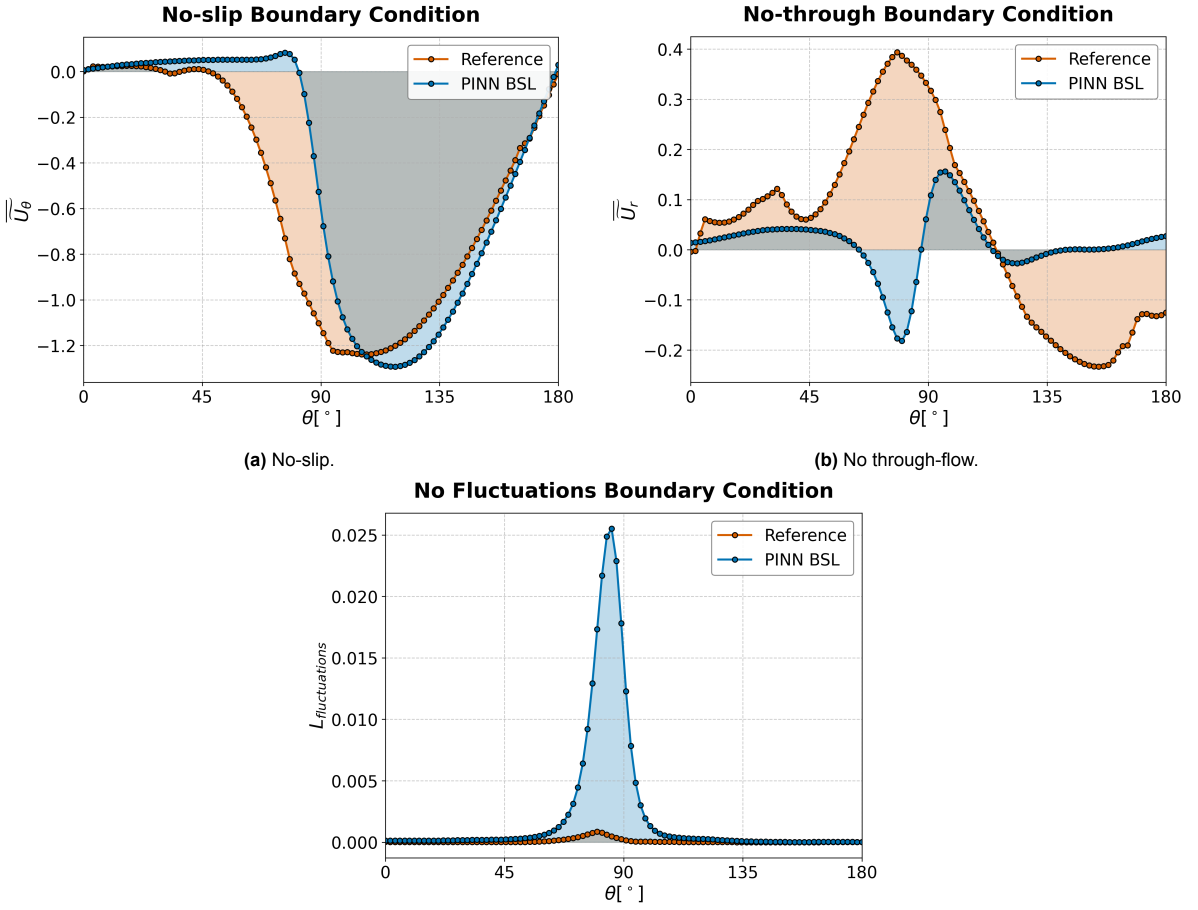
Previous to the exposition of the results from the study, the extent to which the boundary conditions are fulfilled in the reference data as well as by the baseline *PINN* is reflected in Figure 4.23. Note that the values corresponding to the source data have been obtained via linear extrapolation normal to the surface.

Starting from the top-left, Figure 4.23a depicts the slip velocity at the cylinder surface. It can be seen that both the source data and the baseline *PINN* offer a similar distribution, having a virtually null slip velocity from stagnation to $\theta = 45^\circ$ in the reference data and to $\theta = 90^\circ$ for the *PINN*. At $\theta = 90^\circ$, the slip velocity largely increases to around $U_\theta \approx -1.2 \cdot U_\infty$, progressively recovering towards zero at the back of the cylinder.

On the other hand, the lines in Figure 4.23b show the radial velocity at the cylinder boundary. In this case, the differences between both data sources are more evident, with the reference data reaching values of up to $U_r = +0.4 \cdot U_\infty$ right before separation and negative values in the recirculation region. On the contrary, the $PINN$ output results in less error with $|U_r| \leq 0.2 \cdot U_\infty$, concentrating the error around the separation point.

Finally, the no-fluctuations boundary condition in Figure 4.23c offers disparate responses, where the label data presents a null value throughout the surface and the $PINN$ incurs errors of up to 2% of U_∞^2 right before the cylinder apogee.

While the effects of these deviations at a local level are obvious in terms of data loss, the extent to which they affect the pressure distribution at the surface is less obvious, as this takes place through the N-S equations. Nonetheless, the subsequent results shed more light on this topic.



(a) No-slip.

(b) No through-flow.

(c) No fluctuations.

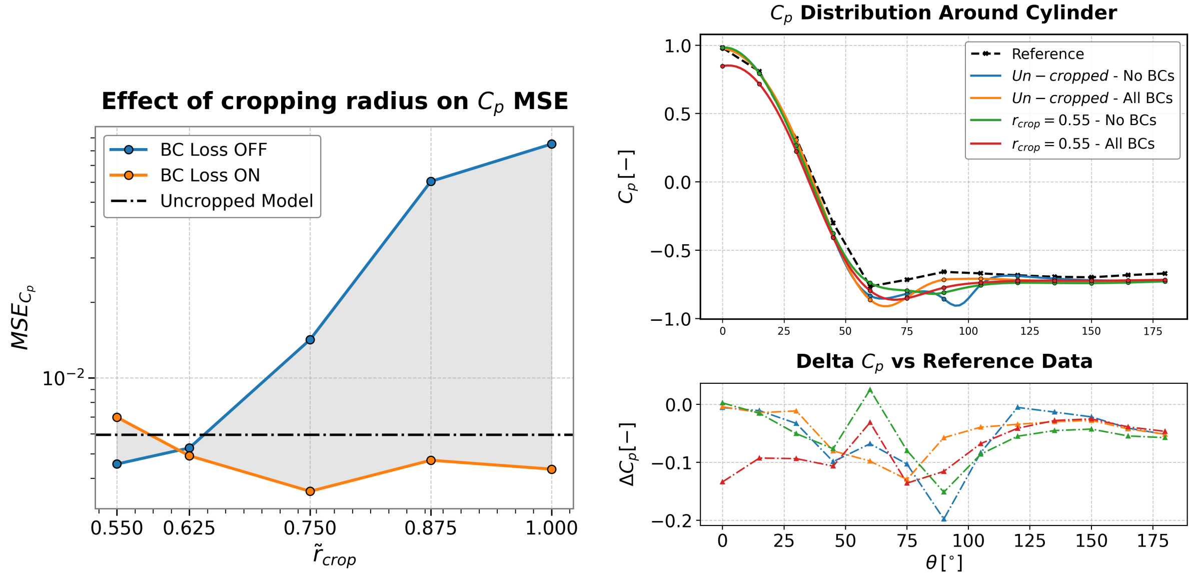
Figure 4.23: Physical boundary conditions on the cylinder surface - baseline *PINN* vs Reference.

As a first step to understanding the interplay between the imposition of the various boundary conditions, Table 4.6 reflects the extent to which these are satisfied for the *PINN* models trained with combinations of all three physical constraints, including the reference and baseline *PINN* results for completeness. To begin with, it can be seen that optimizing for the no-slip and no-through conditions independently or jointly have similar effects, reducing $\mathcal{L}_{no-slip}$ and $\mathcal{L}_{no-through}$ by approximately six and four orders of magnitude with respect to the baseline or reference data levels, without particularly affecting the fluctuations loss. In a comparable, but opposite way, enforcing the lack of fluctuations at the surface brings that particular loss term to $\mathcal{L}_{no-fluctuations} = 5.7 \times 10^{-7}$, barely affecting the two other contributions.

Table 4.6: Boundary condition loss components - effect of BC loss type.

Model	$\mathcal{L}_{\text{no-slip}}$	$\mathcal{L}_{\text{no-through}}$	$\mathcal{L}_{\text{no-fluctuations}}$
Reference	5.2×10^{-1}	3.7×10^{-2}	8.4×10^{-5}
PINN BSL	4.7×10^{-1}	3.7×10^{-3}	2.2×10^{-3}
$\mathcal{L}_{\text{no-slip}}$	1.5×10^{-7}	1.5×10^{-7}	3.0×10^{-3}
$\mathcal{L}_{\text{no-through}}$	6.2×10^{-7}	7.4×10^{-7}	5.1×10^{-3}
$\mathcal{L}_{\text{no-slip}} + \mathcal{L}_{\text{no-through}}$	2.1×10^{-7}	1.7×10^{-7}	8.4×10^{-3}
$\mathcal{L}_{\text{no-fluctuations}}$	3.8×10^{-1}	4.7×10^{-3}	5.7×10^{-7}
$\mathcal{L}_{\text{no-slip}} + \mathcal{L}_{\text{no-through}} + \mathcal{L}_{\text{no-fluctuations}}$	5.5×10^{-7}	2.1×10^{-7}	3.9×10^{-7}

Before investigating how the enforcement of the three physical constraints affects the reconstruction of pressure, Figure 4.24a reflects the evolution of the surface $MSE_{\overline{C_p}}$ as more and more data points are removed from the proximity of the boundary, as well as the effect of including all three boundary condition loss terms in the training process. In the plot, it is seen that the baseline (un-cropped) model realizes an error of $MSE_{\overline{C_p}} = 4.6 \times 10^{-3}$, as previously mentioned. Of considerable remark is that, not only the $\tilde{r}_{crop} = 0.625$ model without boundary conditions is able to overcome this level of accuracy by achieving an $MSE_{\overline{C_p}}$ of 4×10^{-3}, but also most of the models trained with the boundary conditions loss are able to do so as well. Once again, this shows a consistent benefit of including any physical behavior at the surface as a means to improve robustness to data gaps.



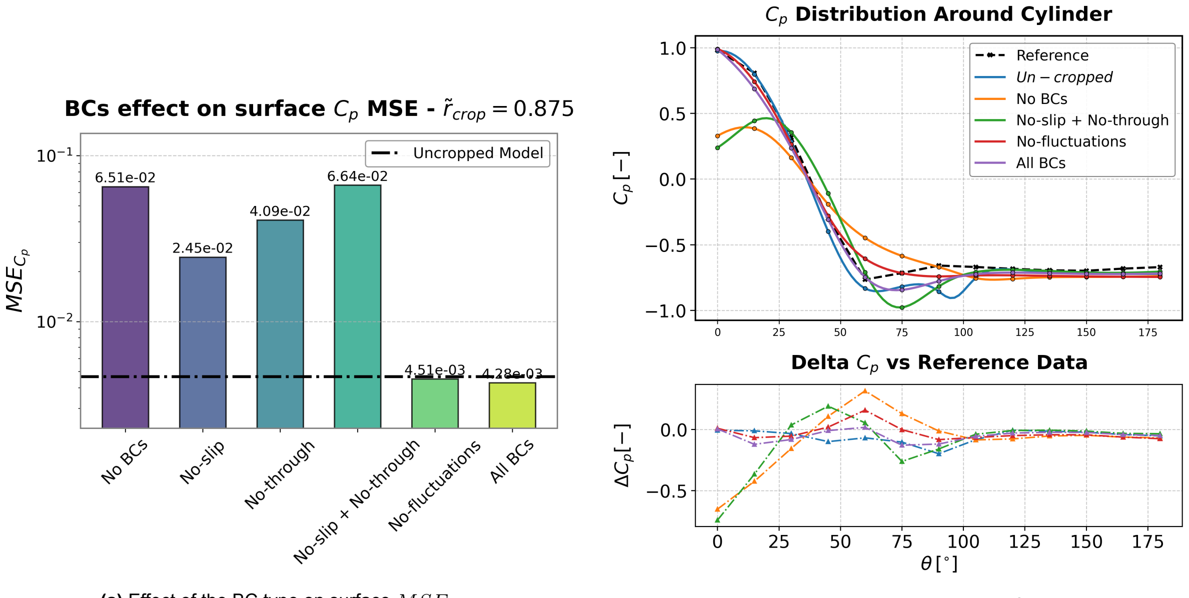
(a) Effect of $\tilde{r}_{crop}$ and $\mathcal{L}_{bc,surf}$ on surface $MSE_{\overline{C_p}}$.

(b) $\overline{C_p}$ distribution on cylinder surface.

Figure 4.24: Effect of data gaps and boundary conditions on surface pressure reconstruction.

However, one can see that the model with $\tilde{r}_{crop} = 0.55$ and boundary conditions appears to break the trend. In order to identify the source of error, Figure 4.24b includes the pressure distributions for both the un-cropped and $\tilde{r}_{crop} = 0.55$ datasets, featuring the variants including and neglecting the surface boundary conditions. While the main point of deviation found in the $\tilde{r}_{crop} = 0.55$ model with boundary conditions is at stagnation, two remarkable findings are made by observing the $\theta \in [90°, 100°]$ region, where consistent deviations have been found along the studies in this section. The first of these is that the un-cropped model where the physical constraints are enforced at the surface, in orange, removes the local secondary suction peak, highlighting the power of including $\mathcal{L}_{bc,surf}$ during training, even with no data gaps. Secondly, the model where $\tilde{r}_{crop} = 0.55$ and with no $\mathcal{L}_{bc,surf}$ also removes this local error, supporting its relation to noisy data close to the cylinder boundary.

As a next step to better understand the effect of the different boundary conditions on the reconstruction accuracy, the mean squared root error of $\overline{C_p}$ at the cylinder surface is displayed in Figure 4.25a for all the combinations in Table 4.6 when trained to the $\tilde{r}_{crop} = 0.875$ dataset. From the bar-plot, it is seen that the cropped model with no boundary conditions enforced presents an error of $MSE_{\overline{C_p}} = 6.5 \times 10^{-2}$, in line with the plot in Figure 4.24a. Notably, introducing the no-slip, no-through or both loss terms during training appear to give distinct error reduction levels, with the latter barely introducing any benefit. This seems to be tied to an inability of these three models to properly reconstruct the stagnation pressure, as per Figure 4.25b.



(a) Effect of the BC type on surface $MSE_{\overline{C_p}}$.

(b) C_p distribution on cylinder surface.

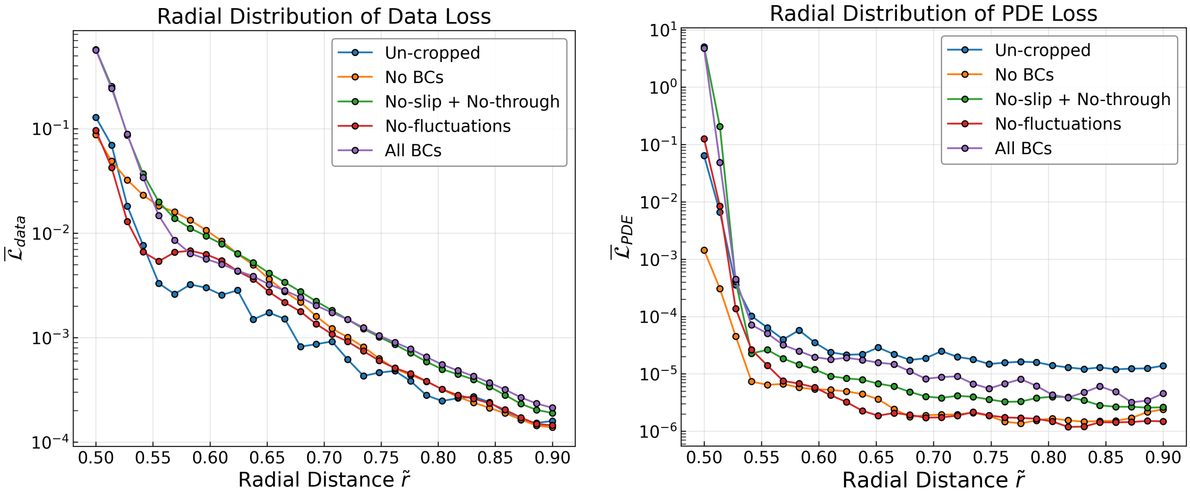
Figure 4.25: Effect of $\mathcal{L}_{BC,surf}$ type on surface pressure reconstruction.

Besides, it appears that the other area of variability is the one around the suction peak and separation point, where the largest wall-normal gradients occur. In this case, the cropped model with no boundary conditions only achieves a smoothed out version of the distribution, resulting in $\Delta C_p = 0.4$ at $\theta = 60^{\circ}$. On the other hand, it is difficult to establish a clear trend in how the various boundary conditions affect accuracy, as the better-posed model is the one trained with all boundary conditions, which achieves a comparable surface pressure error as the un-cropped model. This balanced impact of the various boundary conditions could be related to the fact that the noise not only affects the first statistical moments, but also the fluctuating components, as all emanate from the same temporal data. Accordingly, adding regularization terms for the mean velocities and for the Reynolds Stresses at the surface appear to add comparable steps in accuracy for that particular region.

Finally, and in a similar procedure to the previous chapter, the mean radial distribution of the two main loss components is shown for the various models assessed. Paying close attention to Figure 4.26a first, it can be stated that there are minimal differences across the various models at the data boundary, namely $\tilde{r} = 0.875$. Approaching the surface, the reconstruction error of all models decays at a similar pace, except for that of the uncropped model, which shows the best reconstruction error, as it would be expected. Nonetheless, as the cylinder surface is reached, all models display mean data losses of $\overline{\mathcal{L}}_{data} \approx 0.1$, while the NN trained with all boundary conditions scales all the way to $\overline{\mathcal{L}}_{data} \approx 0.5$, hinting a conflict between the labeled data and the fulfillment of the boundary conditions.

On the other hand, Figure 4.26b contains the analogous representation for the PDE loss. In this case, the model with the labeled data constraint shows a consistently inferior performance across the radii to those which are only focused on the minimization of the PDE loss. Additionally, at the cylinder surface, there's a clear trend that speaks of a conflict between the fulfilment of the physical constraints and the minimization of the RANS residuals. This can be deduced as the model with no BCs is able to minimize

$\overline{\mathcal{L}}_{PDE}$ to around 1×10^{-3}. As more BCs are enforced, this value is increased to $\overline{\mathcal{L}}_{PDE} \approx 0.1$ for the model without fluctuations and to $\overline{\mathcal{L}}_{PDE} = 6$ for the models with both mean flow constraints (in green) and all three boundary conditions enforced (in purple).



(a) Data Loss.

(b) PDE Loss.

Figure 4.26: Radial distribution of loss terms with different BC Loss types - $\tilde{r}_{crop} = 0.875$.

Summarizing, valuable insights can be extracted from this section. Firstly, it has been verified that the elimination of data within $10\% \, \tilde{r}_{cyl}$ of the surface can be positive for overall reconstruction accuracy, suggesting poor data quality due to reflections. Additionally, further support is given to the idea that the inclusion of the no-slip, no through-flow and no-fluctuations conditions at the body surface can consistently reduce the surface pressure reconstruction error, especially when data gaps are present close to the surface. Also, the addition of the boundary condition loss term to the un-cropped model reinforces the possibility to overcome reconstruction errors associated with noisy source data. Moreover, it has been proven that forcing of all three physical constraints provides the most accurate $\overline{C_p}$ distribution, highlighting the role of each term.

4.3.5. Surface pressure data

As a final test, the effect of including pressure tap data as another data loss term during the *PINN* training process is investigated. The aim of this study is to understand whether feeding low resolution pressure tap data information to the *PINN* allows to achieve reconstruction accuracy levels that would otherwise be unfeasible. The reasoning behind this is that experimental set-ups to measure pressure directly often include rather sparse pressure tap measurements, given some of the reasons discussed in Chapter 1, such as intrusion or the cost of manufacturing. Accordingly, it would be of interest to identify whether including such a layout in combination with a *PINN*-based pressure reconstruction from PIV would elevate its performance.

To achieve this, the baseline *PINN* framework is trained adding the MSE of pressure at the cylinder boundary locations corresponding to $\theta = \{0°, 90°, 180°\}$. These locations are chosen to be equally spaced and as far away as possible from one another and it is reckoned that a linear interpolation between their values would not provide a representative pressure distribution, hence deeming it an insufficient resolution on its own.

In addition to the above, a complementary test is carried out by training another *PINN* model which includes all pressure tap values as labeled data. The idea behind this is that training the *PINN* with this information can provide further insight as to what the causes are that produce the persistent deviations found in this Chapter, especially around the suction peak region.

Since the Mean Squared Error with respect to the pressure taps is not a representative metric to compare these two models with respect to the baseline *PINN* due to the direct enforcement of the pressure tap data, the distributions of the output variables are analyzed directly.

In Figure 4.27, the mean pressure coefficient distributions on the cylinder surface are shown for the baseline *PINN* and both pressure data-fed models, as well as their respective deltas to the pressure tap data. Along the same lines as with previous sections, the area which concentrates the largest spread among models is $\theta \in [60°, 105°]$. Remarkably, both models where pressure data is fed appear to chase the un-constrained solution, presenting oscillations that tend towards the baseline distribution in between the pressure data locations. An obvious instance of this is identified at $\theta \approx 95°$, where both the orange and green lines display a peaky oscillation towards that secondary suction peak mentioned in previous sections. This can be taken as tentative proof that there is a mismatch between the flow solution that fulfills the N-S while reproducing the PIV data and the one that matches the pressure tap data while following the PDE loss.

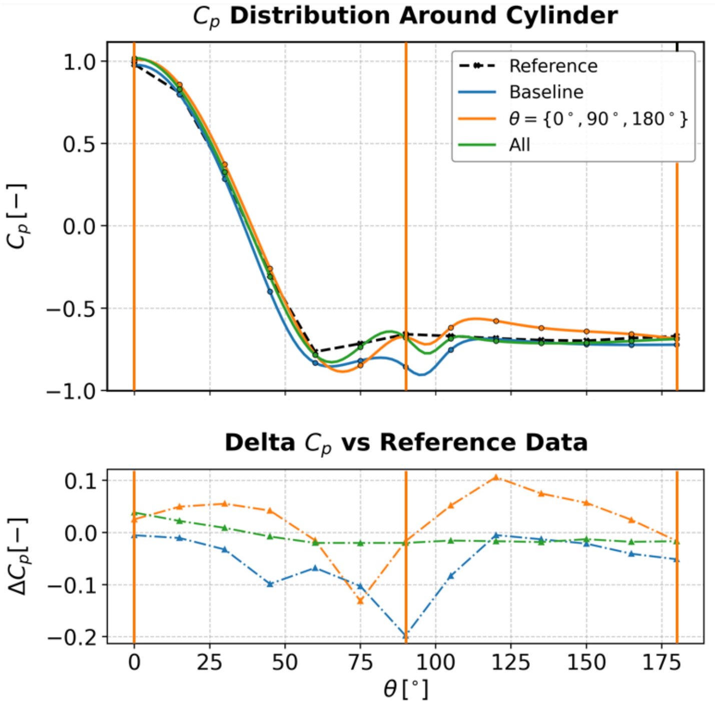


Figure 4.27: Mean pressure coefficient distribution - effect of introducing surface pressure data.

In order to ensure this is the case, the velocity field output by the *PINN* model trained with all pressure data points is plotted against the PIV data in Figure 4.28. As per the delta relative to the free-stream, it becomes apparent that the imposition of the pressure data at the surface can only be accomplished with a comparable level of PDE loss minimization via the increment of the x-velocity data loss.

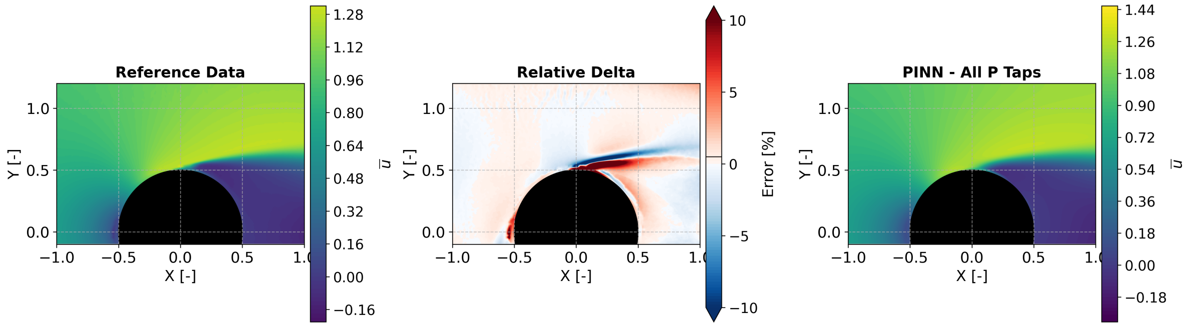


Figure 4.28: Contours of mean x-velocity - effect of introducing all pressure tap data.

Finally, by assessing the distribution of the residuals around the cylinder surface, it is possible to identify which areas pose the main barrier for the PDE loss minimization given that the PIV data and the pressure tap data are learned. An instance of this is displayed in Figure 4.29, where the distribution of the residual corresponding to the x-momentum conservation equation is shown close to the surface for both models.

Markedly, a different behavior is observed with respect to the apparently random distribution of residuals shown in Figure 4.15b for the baseline *PINN* model. Specifically, in this case both models show a concentration of higher absolute residuals close to the surface and around the separation point at $\theta = 90°$. In alignment with the observations made above, this means that particular area requires a penalty in terms of N-S loss minimization when accommodating the pressure tap and PIV data at the same time.

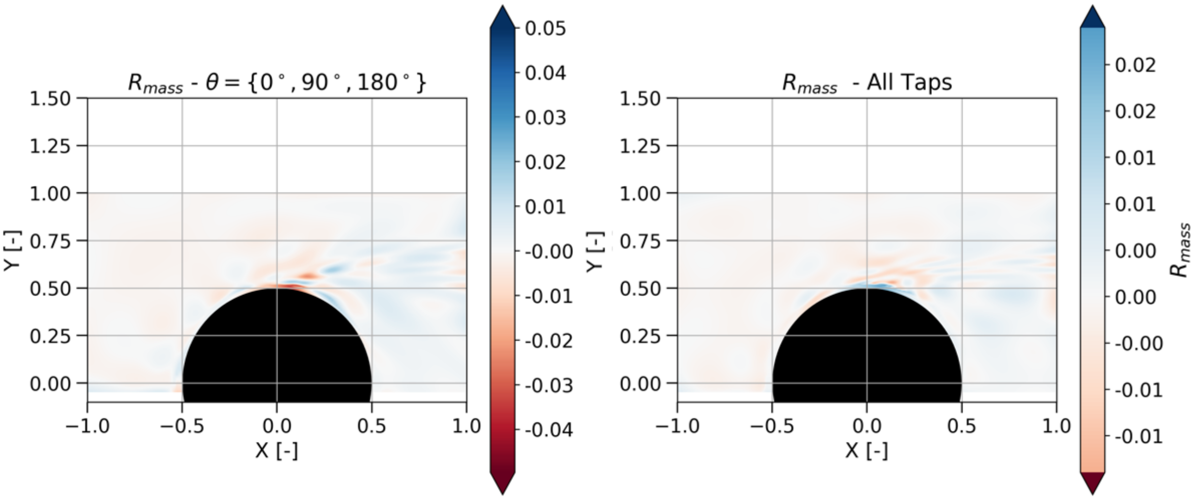


Figure 4.29: Effect of introducing surface pressure tap data on converged x-momentum residual distribution.

Overall, two main conclusions may be drawn from this study. On the one hand, it appears that the enforcement of sparse pressure data values at the surface is able to minimize $\overline{C_p}$ reconstruction error only locally. This appears to be the result of the data-induced solution prevailing locally where only PIV data is available. In fact, the inclusion of such tap data may introduce non-physical oscillations where both data sources disagree, as shown by the local increase of the PDE residuals, thus constituting the second observation.

4.4. Observations from Chapter 4

Prior to attempting to answer the main research questions posed in this Thesis, it is of interest to highlight some of the key findings made in this Chapter, regarding the use of the *PINNs* framework in the reconstruction of surface pressure from experimental time-averaged PIV data.

In the first place, it has been demonstrated that the baseline *PINN* framework is able to reconstruct surface pressure for two flow cases, namely the two-dimensional flow around a smooth cylinder and with zig-zag strips fitted at $\theta = 45°$. In particular, the algorithm achieves respective reductions of 1% and 21% in pressure MSE with respect to the state-of-the-art Poisson solver when comparing to evenly-spaced pressure tap measurements at thirteen angular locations.

Furthermore, deeper investigation of the main sources of discrepancy between both solutions reveals that the regularization added by the RANS equations in the form of the physics-loss is responsible, given the local concentration of high-residuals around the suction peak area in the Poisson-solver output data when compared to the *PINN* solution.

In addition, extensive analysis is made on the different sensitivities offered by the various *PINN* and training parameters on pressure reconstruction accuracy at the surface, particularizing for the smooth cylinder case.

In that regard, it has been shown that simpler NN shapes with dimensions as small as $L = 2$ are able to reconstruct surface pressure even beyond the baseline architecture, contrary to the observations in Chapter 3. On the one hand, this is presumably possible due to the larger point density and lesser information to be learned, given only the top half of the flow field is being fed. On the other hand, the additional performance with respect to the baseline framework initially hints at the additional degrees of freedom being aimed at learning the spatially correlated noise in the data close to the surface, this effect being reduced for simpler NN architectures.

Secondly, the effect of static and dynamic weighting schemes on the data loss is shown to have only a minor effect over the ability of the *PINN* to reconstruct surface pressure. However, weightings that bias the total loss towards the data loss component progressively deviate from the target pressure distribution in the same suction peak area, supporting the hypothesis of data-associated reconstruction errors.

In the third place, the main findings from studying the effect of the number of measurement and collocation points are along the same lines. While the removal of data points subtly reduced some of the oscillations in the baseline pressure distribution, the addition of collocation points in a random sampling approach aided to mute the un-physical response of the algorithm in the areas prone to errors. Thus, this supports the idea that the PDE loss can aid to regularize areas affected by noise.

Subsequently, the effect of enforcing physical constraints at the solid boundary in the form of the no-slip, no through-flow and no-fluctuations boundary conditions has been exposed, placing especial emphasis on their role when labeled data gaps exist around the body of interest. In this case, the removal of data within the boundary layer thickness reduces the reconstruction error in the suction peak area, further evidencing the existence of noisy data. Furthermore, the enforcement of all three physical constraints appears to provide consistent error reductions with or without data gaps, also aiding to reduce fluctuations caused by reflections.

Finally, the addition of sparse surface pressure information from the pressure taps as labeled data during training shows that this is only beneficial locally, given the tendency of the model to divert towards the same non-physical oscillations in the areas where the PIV data is noisy. In addition, study of the residuals distribution around the cylinder for the *PINN* model trained with data from all pressure ports has been used to conclude that the areas where the predictions differ from the pressure tap data in more than $\Delta C_p = 0.1$ show higher residuals, pointing to a mismatch between the physical solutions that describe the C_p reconstructions from the (noisy) PIV data and the distribution given by the pressure taps.

5 Conclusions

Once concluded the detailed discussion of the *PINN* results from both CFD simulations and experimental measurements, it is of interest to provide concise answers to the main and secondary research questions posed in §1.5.2 based on the observations and results obtained.

5.1. Main research question

How does the performance of *PINNs'* quantitatively and qualitatively compare to a conventional PIV-based pressure reconstruction technique like the Poisson Solver in the inference of surface pressure information when compared to pressure tap data?

In Chapter 4, surface pressure reconstruction has been carried out from the PIV data of the flow around a two-dimensional cylinder in two configurations, namely a smooth one and with zig-zag strips fitted at $\theta = 45°$. Based on the comparison of the *PINN* and the Poisson solver outputs with pressure tap data, it has been proven that the baseline *PINN* framework can achieve reductions of 1% and 21% in surface pressure MSE with respect to the Poisson solver for both cases, respectively. Additionally, fine tuning of the *PINN* model for the smooth case via the inclusion of the no-slip, no through-flow and no-fluctuations boundary conditions has been proven to provide surface pressure MSE reductions of up to 50% when compared to the state of the art (Poisson) solver.

Additional relevant quantitative comparison between both methods concerns the resolution time. In this case, while the Poisson solver is instantaneous, the *PINN* training time can fluctuate depending on several variables like the NN architecture or the number of collocation points. For reference, the average physical time to converge to the final solution of the baseline architecture is of around 20 minutes when considering a batch size of ten-thousand collocation points and the workstation specified in §2.1.

On the qualitative side, it has been proven throughout both Chapters that *PINNs* constitute a flexible framework that allows to introduce further prior information into the optimization process. Examples of this are the boundary condition loss terms or the 'prior loss' term to enforce positive normal Reynolds Stress components. Besides, part of such adjustability derives from the freedom to include collocation points using different techniques, which has been proven to provide local regularization in areas affected by correlated noise, such as those where primary and secondary reflections exist.

Along these lines, the nature of the algorithm allows to get rid of discretization errors, thanks to the use of the principle of automatic differentiation. This facilitates the computation of all first and second order derivative terms in the PDE loss via the chain rule of differentiation of the NN outputs with respect to the inputs.

Finally, it has also been shown that *PINNs* enable users to conduct thorough debugging steps, allowing to identify the root causes of the main sources of error and respond accordingly. A clear instance of this is the possibility to train the *PINN* feeding the validation data, thus showing whether this entails a localized worsening of the PDE residuals, for example.

5.2. Secondary research questions

How do data gaps close to the solid boundary affect the accuracy of surface pressure reconstruction accuracy for *PINNs*?

One of the sensitivity studies most relevant to experimental studies concerns the effect that missing data can have on the surface pressure reconstruction accuracy. Since missing data can derive from a number of reasons, such as reflections, shaded areas or lack of visual access, a wide range of data gap sizes has been assessed, covering gaps from 10% of $\tilde{r}_{cyl}$ to 100% of $\tilde{r}_{cyl}$ in the radial direction.

Results on the experimental dataset show that the error progressively increases with the radial data gap size. Specifically, while the pressure MSE for a data gap of 10% $\tilde{r}_{cyl}$ is of 0.0046, this value increases to 0.014 or 0.083 for gaps with sizes $50\% \, \tilde{r}_{cyl}$ and $100\% \, r_{cyl}$, respectively.

These results have been obtained retaining the original data points as collocation points, even though it has been demonstrated that the collocation point generation around the data gap is of, at least, second order of relevance when it comes to accuracy.

Nonetheless, it has been found that the penalty in surface pressure reconstruction accuracy introduced by such gaps can be mitigated via the inclusion of information about the flow at the surface, for instance, via the inclusion of physical boundary conditions, leading to the next secondary research question.

To what extent does the inclusion of physics-based boundary conditions on the surface (e.g. the no-slip condition) affect the accuracy of surface pressure reconstruction?

In order to answer this research question, physical boundary conditions of the Dirichlet type in the form of the no-slip, the no-penetration and no-fluctuations constraints have been enforced at the cylinder surface in various contexts.

In the first place, and in line with the answer to the previous secondary research question, the effect of their combined enforcement has been assessed at each of the data gap sizes, consistently obtaining error reductions for both the CFD and the experimental datasets. Examples of this are the MSE reductions of -76% and -92% achieved on the experimental data set when including the boundary conditions for data gaps with sizes $50\% \, \tilde{r}_{cyl}$ and $75\% \, \tilde{r}_{cyl}$, respectively.

Not only this, but their use has also been shown to allow to mitigate the effects of poor data quality close to the surface. In particular, the baseline *PINN* algorithm trained on the un-cropped dataset saw a -50% reduction in overall surface pressure MSE. More importantly, local fluctuations in the areas identified to be affected by lower quality data are successfully removed via the inclusion of such physical constraints.

Finally, the independent addition of all three boundary conditions and their respective combinations supported that the combined effect of all three yields consistently better results than their respective parts, both regarding CFD and experimental applications.

How much does the provision of reference surface pressure data from pressure taps positively affect the pressure reconstruction accuracy of *PINNs*?

Yet another sensitivity study has been carried out with regards to the experimental dataset, aiming to assess the benefit, if any, introduced by the enforcement of known surface pressure values on the overall surface pressure reconstruction.

While the direct enforcement of surface pressure makes it challenging to quantify this effect via the MSE at all pressure port locations, visual inspection of the deltas for the baseline *PINN* and an analogous model fed with port data at $\theta = \{0°, 90°180°\}$ has allowed to conclude that the error minimization is only materialized locally, rather than propagated along the entire surface. In particular, it has been witnessed that the un-physical oscillations around the noisy data regions are, rather than reduced, accentuated as the model attempts to learn both the PIV and the labeled surface pressure data.

The findings in this study assume, however, that reference pressure values at certain locations in the domain can be established, for instance, via the Bernoulli equation for pressure of incompressible, inviscid flows.

To which extent is the performance of *PINNs* on CFD simulation data representative of its ability to reconstruct pressure from experimental data?

While difficult to quantify numerically, analogous sensitivity studies have been carried out for both types of datasets, enabling to highlight some of the main similarities and differences.

While some of the studies such as the effect of the NN shape or the sensitivity to the number of data and collocation points have resulted in utterly dissimilar results for the two types of datasets, the sample size and and nature of each dataset (spatial resolution, flow topology, number of samples in the time averaging, etc) make it difficult to judge whether the source of dissimilarity is the type of data source.

Nonetheless, key findings have been extracted from the results on both data-sets, proving not only the difficulty in establishing an universal law for all *PINN* applications, but also the fact that both sources of information can be valuable to make informed decisions.

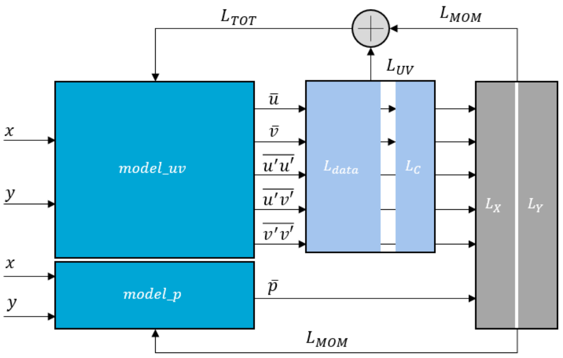
Note, however, that it has been proven that, while there are tools within the *PINN* framework to damp their impact on accuracy, such as collocation points or boundary conditions, it has been proven that correlated noise can and does affect the pressure reconstruction accuracy, perhaps to a greater extent than random Gaussian noise. As a result, it becomes obvious why further research based on real flow applications and experimental tests are required to better quantify the extent of such impact and ways to mitigate it.

5.3. Future work

Based on the research carried out, further advancements are proposed next that can allow to comprehend and set limitations on the use of *PINNs* for (surface) pressure reconstruction:

- study more in-depth the effect of reflections - for instance, is it worthy removing noisy data, effectively converting it into a data gap or keeping it during training?
- deepen study on the relevance of the NN architecture size, especially due to its relevance on training time. Questions of interest: is the performance drop-off for very small model dimensions related to the 'amount of information' contained on each labeled data point or rather on the number of data points? This arises given the disparate response in the same study on the CFD and experimental datasets, which can suggest the first aspect to be determinant. If so, how can one identify what a suitable NN size could be for a given case without requiring to do a full sweep of sizes?
- given the observations made in the experimental section of the collocation points sensitivity study, where non-physical fluctuations are reduced as more collocation points are included, can the local addition of collocation points in noisy areas be sufficient to override its impact in a more effective way?
- if surface pressure data were to be the main target of a given study, what is the minimum amount of data around the surface that is required to achieve optimal results? (as long as a reference pressure value is known).
- what is the impact of having non-two-dimensional measurement planes (i.e. tilted laser with respect to the onset) on the surface pressure reconstruction accuracy? How does this compare to the analogous effect on the Poisson solver? Can one improve accuracy in such cases by removing the continuity equation from the loss function?
- can alternative types of architectures be laid out to improve efficiency? For instance, I initially studied the use of a 'Frankenstein´layout (as per Figure 5.1), which effectively splits the labeled

and hidden output variables into two models, which can take a huge advantage of initially applying zero PDE loss to the known data, greatly reducing the initial optimum discovery phase given the shape of the 'data-only' optimization space.



PROs

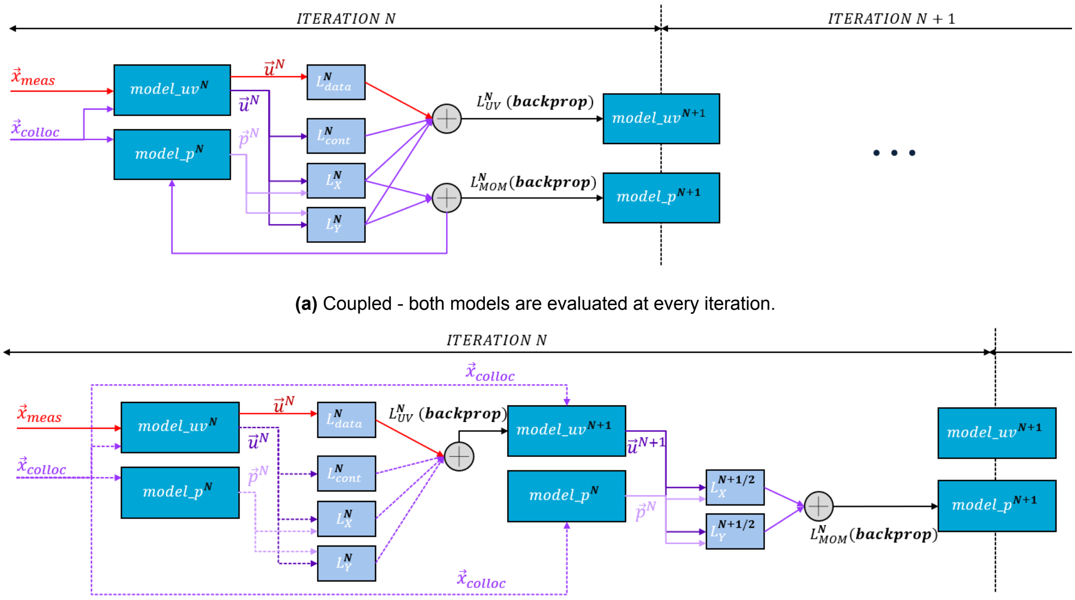
- Possible to increase L_{data} weighting independently of model_p training
- Flexibility of NN hyperparameters (interesting since data-based training appears to converge faster)
- Ease of debugging

CONs

- Training time
- Complexity of implementation

Figure 5.1: 'Frankenstein' architecture with one model to reconstruct labeled data (can use feedback from data and/or PDE losses) and a different model to reconstruct pressure data (using only the PDE loss component).

- along the same lines, can different training data flows be studied to accelerate convergence? For instance, using the 'Frankenstein' architecture could allow one to do 'coupled' (Figure 5.2a) and 'staggered' (Figure 5.2b) iterative steps, where the latter could offer the advantage of reducing the training time via the computation of the PDE loss only after advancing the labeled data model.



(a) Coupled - both models are evaluated at every iteration.

(b) Staggered - both models can be evaluated sequentially, where the output of the advanced labeled data model is used in the PDE loss calculation to optimize the pressure reconstruction model.

Figure 5.2: Examples of iterative layouts that can be accommodated by the Frankenstin architecture.

- since most of the literature supports the idea that *PINNs* display a noticeable robustness to noise propagation, can this framework accommodate larger spatial PIV resolutions even if at the cost of certain noise?
- further test cases on real applications, such as the flying-V TU Delft project, where pressure tapping might be challenging due to space requirements.

References

[1] Elmar Achenbach. “Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to Re = 5 × 10^6”. In: *Journal of Fluid Mechanics* 34.4 (1968), pp. 625–639. DOI: 10.1017/S0022112068002120.

[2] R. J. Adrian and J. Westerweel. “Particle Image Velocimetry”. In: *The Aeronautical Journal* 116.1176 (2012), pp. 219–220. DOI: 10.1017/S0001924000006734.

[3] Airbus. *A320neo | Airbus Aircraft*. Accessed: 2025-03-15. 2025. URL: https://aircraft.airbus.com/en/aircraft/a320-family/a320neo.

[4] John D. Anderson. *Fundamentals of Aerodynamics*. 5th. New York: McGraw-Hill Education, 2010. ISBN: 978-0073398105.

[5] Genick Bar-Meir. *Basics of Fluid Mechanics*. 2nd. USA, 2009. URL: https://www.potto.org.

[6] B. C. Basu and G. J. Hancock. “The unsteady motion of a two-dimensional aerofoil in incompressible inviscid flow”. In: *Journal of Fluid Mechanics* 87.1 (1978), pp. 159–178. DOI: 10.1017/S0022112078002980.

[7] George K. Batchelor. *An Introduction to Fluid Dynamics*. 1st. Cambridge: Cambridge University Press, 1967.

[8] T. Baur and J. Köngeter. “PIV with high temporal resolution for the determination of local pressure reductions from coherent turbulence phenomena”. In: *Proceedings of the 3rd International Workshop on Particle Image Velocimetry*. Santa Barbara, CA, Sept. 1999, pp. 101–106.

[9] Atilim Gunes Baydin et al. “Automatic differentiation in machine learning: a survey”. In: *Journal of Machine Learning Research* 18.153 (2018), pp. 1–43.

[10] G Berkooz, PJ Holmes, and John Lumley. “The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows”. In: *Annual Review of Fluid Mechanics* 25 (Nov. 2003), pp. 539–575. DOI: 10.1146/annurev.fl.25.010193.002543.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[12] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. “Machine Learning for Fluid Mechanics”. In: *Annual Review of Fluid Mechanics* 52 (2020), pp. 477–508. DOI: 10.1146/annurev-fluid-010719-060214. URL: https://doi.org/10.1146/annurev-fluid-010719-060214.

[13] Shengze Cai et al. “Physics-informed neural networks (PINNs) for fluid mechanics: a review”. In: *Acta Mechanica Sinica* 37.12 (2021), pp. 1727–1738. DOI: 10.1007/s10409-021-01148-1.

[14] Shengze Cai et al. “Physics-informed neural networks (PINNs) for fluid mechanics: a review”. In: *Acta Mechanica Sinica* 37.12 (2022), pp. 1727–1738. DOI: 10.1007/s10409-021-01148-1.

[15] Michael A. Calicchia et al. “Reconstructing the pressure field around swimming fish using a physics-informed neural network”. In: *Journal of Experimental Biology* 226 (2023), jeb244983. DOI: 10.1242/jeb.244983.

[16] John Charonko et al. “Assessment of pressure field calculations from particle image velocimetry measurements”. In: *Measurement Science and Technology* 21 (Aug. 2010), p. 105401. DOI: 10.1088/0957-0233/21/10/105401.

[17] S.H. Chue. “Pressure probes for fluid measurement”. In: *Progress in Aerospace Sciences* 16.2 (1975), pp. 147–223. ISSN: 0376-0421. DOI: https://doi.org/10.1016/0376-0421(75)90014-7. URL: https://www.sciencedirect.com/science/article/pii/0376042175900147.

[18] John O. Dabiri et al. “An algorithm to estimate unsteady and quasi-steady pressure fields from velocity field measurements”. In: *Journal of Experimental Biology* 217.3 (2014), pp. 331–336. DOI: 10.1242/jeb.092767.

[19] Delft University of Technology. *M-Tunnel*. https://www.tudelft.nl/en/ae/organisation/departments/flow-physics-and-technology/facilities/low-speed-wind-tunnels/m-tunnel. Accessed: 2025-05-06.

[20] Zhiwen (□□□) Deng et al. "Time-resolved turbulent velocity field reconstruction using a long short-term memory (LSTM)-based artificial intelligence framework". In: *Physics of Fluids* 31.7 (July 2019), p. 075108. ISSN: 1070-6631. DOI: 10.1063/1.5111558. eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/1.5111558/14067062/075108_1_online.pdf. URL: https://doi.org/10.1063/1.5111558.

[21] G. E. Elsinga et al. "Tomographic particle image velocimetry". In: *Experiments in Fluids* 41.6 (2006), pp. 933–947. DOI: 10.1007/s00348-006-0212-z. URL: https://doi.org/10.1007/s00348-006-0212-z.

[22] Di Fan et al. "Comparative assessment for pressure field reconstruction based on physics-informed neural network". In: *Physics of Fluids* 35.7 (2023), p. 077116. DOI: 10.1063/5.0157753.

[23] Gary A. Flandro, Howard M. McMahon, and Robert L. Roach. *Basic Aerodynamics: Incompressible Flow*. Cambridge Aerospace Series. Cambridge University Press, 2011.

[24] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010, pp. 249–256.

[25] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[27] Gazi Hasanuzzaman et al. "Enhancement of PIV measurements via physics-informed neural networks". In: *Measurement Science and Technology* 34.4 (Jan. 2023), p. 044002. DOI: 10.1088/1361-6501/aca9eb.

[28] E. L. Houghton et al. *Aerodynamics for Engineering Students*. 7th. Oxford, UK: Butterworth-Heinemann, 2016. ISBN: 978-0081001943.

[29] ANSYS Inc. *Reynolds Stress Models in ANSYS CFX*. Accessed: 2025-03-20. 2024. URL: https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v242/en/cfx_mod/i1346049.html.

[30] Oxford Instruments. *Guide to PIV Mode for iStar sCMOS Camera*. Accessed: 2025-03-20. 2024. URL: https://andor.oxinst.com/learning/view/article/piv-mode-for-istar-scmos.

[31] Xiaowei Jin et al. "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations". In: *Journal of Computational Physics* 426 (2021). DOI: 10.1016/j.jcp.2020.109951.

[32] R. de Kat and B. W. van Oudheusden. "Instantaneous planar pressure determination from PIV in turbulent flow". In: *Experiments in Fluids* 52 (2012), pp. 1089–1106. DOI: 10.1007/s00348-011-1237-5. URL: https://doi.org/10.1007/s00348-011-1237-5.

[33] Nikolaos D. Katopodes. "Chapter 5 - Viscous Fluid Flow". In: *Free-Surface Flow*. Butterworth-Heinemann, 2019, pp. 324–426. ISBN: 978-0-12-815489-2. DOI: https://doi.org/10.1016/B978-0-12-815489-2.00005-8. URL: https://www.sciencedirect.com/science/article/pii/B9780128154892000058.

[34] Joseph Katz. *Race Car Aerodynamics: Designing for Speed*. 1st. Cambridge, MA: Bentley Publishers, 1995. ISBN: 978-0837601427.

[35] Joseph Katz and Allen Plotkin. *Low-Speed Aerodynamics*. 2nd. Cambridge University Press, 2001. ISBN: 978-0521665520.

[36] Richard Keane and Ronald Adrian. "Optimization of particle image velocimeters. I. Double pulsed systems". In: *Measurement Science and Technology* 1 (Jan. 1999), p. 1202. DOI: 10.1088/0957-0233/1/11/013.

[37] Saša Kenjereš and Kemal Hanjalić. “Tackling complex turbulent flows with transient RANS”. In: *Fluid Dynamics Research* 41.1 (Jan. 2009), p. 012201. DOI: 10.1088/0169-5983/41/1/012201. URL: https://dx.doi.org/10.1088/0169-5983/41/1/012201.

[38] Matthew S. Kuester, Aurelien Borgoltz, and William Devenport. “Pressure Tap Effects on the Lift Measurement of an Airfoil Section”. In: *32nd AIAA Aerodynamic Measurement Technology and Ground Testing Conference*. Washington, D.C.: AIAA, June 2016. DOI: 10.2514/6.2016-3654.

[39] Bozhen Lai, Yingzheng Liu, and Xin Wen. “Temporal and spatial flow field reconstruction from low-resolution PIV data and pressure probes using physics-informed neural networks”. In: *Measurement Science and Technology* 35.6 (2024), p. 065304. DOI: 10.1088/1361-6501/ad3307.

[40] Changhoon Lee et al. “Application of neural networks to turbulence control for drag reduction”. In: *Physics of Fluids* 9.6 (1997), pp. 1740–1747. DOI: 10.1063/1.869290.

[41] John H Lienhard et al. *Synopsis of lift, drag, and vortex frequency data for rigid circular cylinders*. Vol. 300. Technical Extension Service, Washington State University Pullman, WA, 1966.

[42] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance”. In: *Journal of Fluid Mechanics* 807 (2016), pp. 155–166. DOI: 10.1017/jfm.2016.615.

[43] G.R. Liu and Y.T. Gu. “Meshless Methods: A Review and Recent Developments”. In: *Archives of Computational Methods in Engineering* 17.3 (2010), pp. 171–198. DOI: 10.1007/s11831-010-9040-7.

[44] Tianshu Liu, M. Guille, and J. P. Sullivan. “Accuracy of Pressure-Sensitive Paint”. In: *AIAA Journal* 39.1 (Jan. 2001).

[45] Xiaofeng Liu, Joseph Katz, and Shridhar Gopalan. “Measurements of Pressure Distribution by Integrating the Material Acceleration”. In: *Proceedings of the ASME Heat Transfer/Fluids Engineering Summer Conference 2004, HT/FED 2004* 3 (Jan. 2004). DOI: 10.1115/HT-FED2004-56373.

[46] Xiaohua Liu and Joseph Katz. “Instantaneous pressure and material acceleration measurements using a four-exposure PIV system”. In: *Experiments in Fluids* 41 (2006), pp. 227–240. DOI: 10.1007/s00348-006-0152-7. URL: https://doi.org/10.1007/s00348-006-0152-7.

[47] R. Manceau and K. Hanjalić. “Elliptic blending model: A new near-wall Reynolds-stress turbulence closure”. In: *Physics of Fluids* 14.2 (2002), pp. 744–754.

[48] Beverley McKeon and Rolf Engler. “Pressure Measurement Systems”. In: *Springer Handbook of Experimental Fluid Mechanics*. Ed. by Cameron Tropea, Alexander L. Yarin, and John F. Foss. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 179–214. ISBN: 978-3-540-30299-5. DOI: 10.1007/978-3-540-30299-5_4. URL: https://doi.org/10.1007/978-3-540-30299-5_4.

[49] B.G. McLachlan and J.H. Bell. “Pressure-sensitive paint in aerodynamic testing”. In: *Experimental Thermal and Fluid Science* 10.4 (1995). Experimental methods in Thermal and Fluid Science, pp. 470–485. ISSN: 0894-1777. DOI: https://doi.org/10.1016/0894-1777(94)00123-P. URL: https://www.sciencedirect.com/science/article/pii/089417779400123P.

[50] A. Melling. “Tracer particles and seeding for particle image velocimetry”. In: *Measurement Science and Technology* 8.12 (1997), pp. 1406–1416. DOI: 10.1088/0957-0233/8/12/005.

[51] Álvaro Moreno Soto, Alejandro Güemes, and Stefano Discetti. “Complete flow characterization from snapshot PIV, fast probes and physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 419 (2024), p. 116652. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2023.116652. URL: https://www.sciencedirect.com/science/article/pii/S0045782523007752.

[52] Yuichi Murai et al. “Particle tracking velocimetry applied to estimate the pressure field around a Savonius turbine”. In: *Measurement Science and Technology* 18.8 (July 2007), p. 2491. DOI: 10.1088/0957-0233/18/8/026.

[53] B. W. van Oudheusden. “PIV-based pressure measurement”. In: *Measurement Science and Technology* 24.3 (2013). Received 11 September 2012, in final form 19 October 2012, published 24 January 2013, p. 032001. DOI: 10.1088/0957-0233/24/3/032001. URL: https://iopscience.iop.org/article/10.1088/0957-0233/24/3/032001.

[54] B. W. van Oudheusden, F. Scarano, E. W. M. Roosenboom, et al. “Evaluation of integral forces and pressure fields from planar velocimetry data for incompressible and compressible flows”. In: *Experiments in Fluids* 43 (2007), pp. 153–162. DOI: 10.1007/s00348-007-0261-y. URL: https://doi.org/10.1007/s00348-007-0261-y.

[55] Hamamatsu Photonics. *A Visual Guide to CCD vs. EM-CCD vs. CMOS*. Accessed: 2025-03-20. 2024. URL: https://camera.hamamatsu.com/jp/en/learn/technical_information/thechnical_guide/visual_guide.html.

[56] Stephen B. Pope. *Turbulent Flows*. Cambridge, UK: Cambridge University Press, 2000. DOI: 10.1017/CBO9780511840531.

[57] Markus Raffel et al. *Particle Image Velocimetry: A Practical Guide*. 3rd. Springer, 2018. ISBN: 978-3-319-68851-0. DOI: 10.1007/978-3-319-68852-7. URL: https://doi.org/10.1007/978-3-319-68852-7.

[58] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[59] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations”. In: *Science* 367.6481 (2020), pp. 1026–1030. DOI: 10.1126/science.aaw4741. URL: https://www.science.org/doi/abs/10.1126/science.aaw4741.

[60] Prajit Ramachandran, Barret Zoph, and Quoc V Le. “Searching for activation functions”. In: *arXiv preprint arXiv:1710.05941* (2017).

[61] S. Rambaldi et al. “Accuracy analysis of a spectral Poisson solver”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 561.2 (2006). Proceedings of the Workshop on High Intensity Beam Dynamics, pp. 223–229. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2006.01.019. URL: https://www.sciencedirect.com/science/article/pii/S0168900206000362.

[62] Gillian N. Saunders-Smits, Joris A. Melkert, and Michiel J. Schuurman. *80 Years of Aerospace Engineering Education in the Netherlands*. Delft, The Netherlands: TU Delft Open, 2020. ISBN: 978-94-6366-338-0. URL: https://research.tudelft.nl/files/68329341/6.2020_0643.pdf.

[63] Fulvio Scarano et al. “On the use of helium-filled soap bubbles for large-scale tomographic PIV in wind tunnel experiments”. In: *Experiments in Fluids* 56.2 (2015), p. 42. DOI: 10.1007/s00348-015-1909-7. URL: https://link.springer.com/article/10.1007/s00348-015-1909-7.

[64] Hermann Schlichting and Klaus Gersten. *Boundary-Layer Theory*. 9th. Springer, 2017. ISBN: 978-3662529171.

[65] Andrea Sciacchitano, Richard P. Dwight, and Fulvio Scarano. “Navier–Stokes simulations in gappy PIV data”. In: *Experiments in Fluids* 53 (Sept. 2012), pp. 1421–1435. DOI: 10.1007/s00348-012-1366-5. URL: https://doi.org/10.1007/s00348-012-1366-5.

[66] -Ing M. Seidel and Ir. F. Jaarsma. “The German-Dutch low speed wind tunnel DNW”. In: *The Aeronautical Journal* 82.808 (1978), pp. 167–173. DOI: 10.1017/S0001924000094987.

[67] Daniel M. Sforza, Christopher M. Putman, and Juan R. Cebral. “Hemodynamics of Cerebral Aneurysms”. In: *Annual Review of Fluid Mechanics* 41 (Jan. 2009), pp. 91–107. DOI: 10.1146/annurev.fluid.40.111406.102126.

[68] Sat Sharma, Muhammad Fahad Hashmi, and Bharath Patel. *Airway Resistance*. Last updated August 14, 2023. Treasure Island, FL: StatPearls Publishing, 2023. URL: https://www.ncbi.nlm.nih.gov/books/NBK554401/.

[69] J. Sola and J. Sevilla. “Importance of input data normalization for the application of neural networks to complex industrial problems”. In: *IEEE Transactions on Nuclear Science* 44.3 (1997), pp. 1464–1468. DOI: 10.1109/23.589532.

[70] Alberto Solera-Rico et al. “β-Variational autoencoders and transformers for reduced-order modelling of fluid flows”. In: *Nature Communications* (2024). DOI: 10.1038/s41467-024-45578-4. URL: https://doi.org/10.1038/s41467-024-45578-4.

[71] James Stewart. *Calculus: Early Transcendentals*. 7th. Brooks Cole, 2012.

[72] PyTorch Documentation Team. *ReduceLROnPlateau — PyTorch 2.6 documentation*. https://pytorch.org/docs/2.6/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html. [Accessed April 12, 2025]. 2021.

[73] John Tencer. *Cylinder in Crossflow*. Version V1. 2020. DOI: 10.7910/DVN/G5MNYF. URL: https://doi.org/10.7910/DVN/G5MNYF.

[74] Nando Timmer and Leo Veldhuis. “The Impact of Skinsuit Zigzag Tape Turbulators on Speed Skating Performance”. In: *Applied Sciences* 11.3 (2021). ISSN: 2076-3417. DOI: 10.3390/app11030988. URL: https://www.mdpi.com/2076-3417/11/3/988.

[75] Roelof Vos and Saeed Farokhi. *Introduction to Transonic Aerodynamics*. Vol. 108. Fluid Mechanics and Its Applications. New York, NY: Springer, 2015. ISBN: 978-1-4939-1314-0. DOI: 10.1007/978-1-4939-1315-7.

[76] Hongping Wang, Yi Liu, and Shizhao Wang. “Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network”. In: *Physics of Fluids* 34.1 (2022). DOI: 10.1063/5.0078143. URL: https://doi.org/10.1063/5.0078143.

[77] Frank M. White. *Viscous Fluid Flow*. 3rd. McGraw-Hill, 2006. ISBN: 978-0072402315.

[78] Fujio Yamamoto and Masaaki Ishikawa. “A Review of the Recent PIV Studies—From the Basics to the Hybridization with CFD”. In: *Journal of Flow Control, Measurement & Visualization* 10.4 (2022), pp. 117–147. DOI: 10.4236/jfcmv.2022.104008. URL: https://www.scirp.org/journal/paperinformation?paperid=120644.