Physics-informed neural networks for the reconstruction of high-Reynolds 3D flows over the Gaussian Boeing bump geometry.

Master Thesis

by

Noah van Santen

Student Name	Student Number
Noah van Santen	4838750

Thesis Committee	Function
Dr. W. Baars	Supervisor
J. Melkert	Chair
Dr. N.A.K Doan	External Examiner

External Supervisors	Affiliation
Dr. J. Humml	California Institute of Technology
Dr. M. Gharib	California Institute of Technology
Dr. V. Lee	Boeing Commercial Airplanes
A. Khodadoust	Boeing Research & Technology

Project Duration:	March, 2024 - December, 2024
Faculty:	Faculty of Aerospace Engineering, Delft

Preface

This thesis summarizes the work performed during my 9-month period at Caltech, and concludes the pursuit of my MSc. in Aerospace Engineering from TU Delft. I would like to dedicate this page to those who have supported me during this journey, and the appreciation that I have for them.

First and foremost, I would like to acknowledge my family, who have always encouraged my educational journey and extended me the freedom to pursue the life I wanted for myself. A big thank you to them for supporting this chapter of my life, and especially for allowing me to conclude the last two years in the United States.

I also would not be where I am today without the great friends who I have met at various instances of this journey, who have joined me in California, kept me company in Delft or cheered me on from afar. A special token of appreciation to my girlfriend without whom I would not have had the courage to move to Caltech in an effort to be ”closer” together. The people I have met are by no exception the greatest takeaway this MSc. has provided me with.

Finally, I would like to extend my gratitude to my supervisors, and thesis committee members from Delft, as well as my colleagues at Caltech; I feel exceedingly grateful for the many experiences and lessons I have learned.

Noah van Santen
Delft, February 2025

Summary

This thesis investigates the potential for Physics-Informed Neural Networks (PINNs) to reconstruct spatially dense, steady-state, 3D flow fields from spatially sparse experimental wind tunnel data over the Gaussian Boeing bump geometry. The difficulty in experimentally obtaining flow field data that is simultaneously accurate, 3-dimensional and offering a wide field of view makes PINNs an intriguing tool for flow enhancement due to their ease and simplicity regarding data-acquisition.

This work attempts to reconstruct 3D separated, mean velocity fields from 2D, two-component PIV and static pressure data at Reynolds numbers up to 3 million. Training and validation data is used from previous experimental collaborations investigating the Gaussian Boeing Bump geometry while additional training data is collected in the Lucas Wind Tunnel at Caltech. The choice of the Gaussian Boeing Bump geometry stems from its propensity to generate complex 3D, separated flow fields through the combination of adverse and favorable pressure gradients at modest incoming flow velocities. The documented inability for low-cost CFD methods to accurately model the unique flow features over the geometry and the abundance of publicly available data for the purposes of training and validation proposes an excellent opportunity for the investigation of Physics-Informed Neural Networks to demonstrate their capacity as a novel data-acquisition and enhancement method.

The PINN method is first verified over a 2D, two-component domain against OpenFOAM data produced over the geometry as a result of previous works. This sets the baseline for the eventual expansion to a 3D PINN model with a verified implementation of the Reynolds-Averaged Navier Stokes equations, to be trained on multiple planes of 2D, two-component PIV data. The 3D predicted flow field is validated against 2D, three-component PIV data along an untrained axis, assessing the reconstruction accuracy of all velocity components. The mean velocity and static pressure data used in training the PINN are used in an ablation study to be individually assessed for their contributions to a successful reconstruction of the flow field. The PINN method's inherent approximation of velocity gradients over the domain further allows for a global prediction of surface shear stresses and skin friction coefficients. Results obtained in previous collaborations using Oil Flow Visualization and CFD offer opportunities to qualitatively validate the reconstruction of surface-shear stresses and skin friction coefficient magnitudes over the surface of the geometry.

The PINN's capacity for reconstructing 3D flow features is ultimately curtailed by the accuracy of its predictions, which remain costly and data-driven. The ablation study on training data highlights the importance of relevant input data, limiting the network's capacity to predict the magnitudes of parameters over which it has no innate information. Despite this, qualitative approximation of flow features was shown with considerable promise, lending the possibility to further gains in accuracy from increased computational investment. Furthermore, the contribution of additional training data and its potential for increased reconstruction accuracy cannot be ignored. The relative youth of the PINN methodology compared to contemporary computational methods, and their aptitude for data acquisition encourage additional investigation for their application as flow enhancement tools.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
PINN	Physics-Informed Neural Network
PIML	Physics-Informed Machine Learning
ODIL	Optimizing a DIscrete Loss
CFD	Computational Fluid Dynamics
APG	Adverse Pressure Gradient
FPG	Favorable Pressure Gradient
DNN	Deep Neural Network
MLP	Multi-Layer Perceptron
PIV	Particle Image Velocimetry
SPIV	Stereo Particle Image Velocimetry
PDE	Partial Differential Equation
BC	Boundary Condition
MSE	Mean-Squared Error
HPC	High-Performance Computing
LDV	Laser Doppler Velocimetry
WMLES	Wall-Modelled Large Eddy Simulation
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MPE	Mean Percentage Error

Symbols

Symbol	Definition	Unit
$\mathbf{U} = (u, v, w)$	Velocity field	[m/s]
$\mathbf{P}$	Pressure field	[Pa]
$\mathbf{X} = (x, y, z)$	Spatial input coordinates	[m]
L	Boeing bump length scale	[m]
Re_L	Reynolds Number	[-]
M	Mach Number	[-]
C_f	Friction Coefficient	[-]
f_{NN}	Neural Network	[-]
N_{layer}	Network layers	[-]
H	Network depth	[-]
σ	Activation function	[-]
$w_{i,j}$	Network layer weight	[-]
b_j	Network layer bias	[-]
$\mathcal{L}$	Loss	[-]
$\mathbf{B}, \mathcal{N}$	Gaussian sampling	[-]
$< \overline{\mathbf{u'u'}} >$	Reynolds stress tensor	[m^2/s^2]
∞	Freestream Condition	[-]
ρ	Density	[kg/m^3]
μ	Dynamic viscosity	[Pa s]
ν	Kinematic viscosity	[m^2/s]

Symbol	Definition	Unit
Θ	Trainable neural network parameters	[-]
Σ	Summation operator	[-]
α	Loss weighting parameter	[-]
ϵ	PDE Residual	[-]
∇	Differential operator	[-]
∇^2	Laplacian operator	[-]
$\gamma(\mathbf{X})$	Random Fourier encoding	[-]
$\sigma_{\mathcal{N}}$	Gaussian standard deviation	[-]

1 Introduction

In recent years, Physics-Informed Neural Networks (PINNs) have emerged in a variety of fields for their ability to model partial differential equation solutions, where the ease of implementation and simple architecture have made PINNs prominent candidates to compete with contemporary numerical solvers. Nevertheless, machine learning (ML) techniques remain data-driven methods, and in large part, the success of a PINN is dependent on the quantity and quality of the data it is trained on.

Simultaneously, the field of experimental aerodynamics suffers from its inherent difficulty in obtaining flow field data that is simultaneously accurate, time-resolved and 3-dimensional. Additionally, for all of these parameters, typically only a restricted field of view (FOV) is available, limiting the ability to capture a global flow field. Complex and turbulent flow fields only amplify this problem, introducing flow separation, out-of-plane motion and in the case of sufficiently high velocities, shock-wave phenomena. Spatially dense, time-resolved data is therefore also necessary to adequately represent the turbulence spectra and observe the dissipation of eddy cascades. For the limited number of wind tunnels that are capable of obtaining representative flow information, the heavy demand for utilization accompanies long waiting lists and expensive costs.

Jointly combining these two disciplines, this thesis will investigate to what extent PINNs can be applied to spatially-sparse experimental wind-tunnel results to reconstruct spatially dense flow information. Additional mean velocity measurement data, beyond the existing benchmark data for the PINN to use in training will be obtained over the Gaussian Boeing speed bump geometry installed at Caltech's Lucas Wind Tunnel. Despite being a simple geometry, the Boeing bump produces a great variety of 3D flow features and separation phenomena that current CFD models struggle to accurately resolve. Thus, their inability to model mean velocities of complex flows continues to limit their capacity to act as iterative and cost-effective design tools. The Boeing bump geometry has been previously studied both experimentally and numerically at the University of Washington, the University of Notre Dame, NASA Langley Research center, offering a variety of opportunities for verification, and validation. In addition to the abundance of data, Boeing's inherent interest and support of this project motivate the geometry selection.

The current experimental setup in the Lucas Wind Tunnel allows for PIV and smoke-based imaging in 2D, and the PINN will be assessed in its ability to reconstruct the 3D, steady-state velocity and pressure fields from spatially-sparse experimental data. Eventually, the PINN will be assessed for whether it can accurately interpret mean velocity and pressure gradients within the boundary layer by predicting the surface shear stresses over the bump surface. A successful campaign will consist of the PINN resolving the complex flow phenomena from coarse experimental measurements and extrapolating 3D flow fields from 2D input data. The ability to adequately model a 3D separated flow field will establish PINNs as a competitive alternative to state of the art numerical methods, and an intuitive supplement to experimentation. Furthermore, an ability to model surface shear stress topologies at Reynolds numbers of $Re \sim 10^6$ will break new ground in the domain of PINNs being applied to fluid mechanics where high-Reynolds flows of this magnitude have not before been successfully modeled.

1.1. Research Proposal

The proposal for this research, and the questions it accompanies are outlined as follows:

To what extent can Physics-Informed Machine Learning methods enhance 2D wind tunnel data for turbulent, 3D flow modeling and aerospace design?

To assist in answering this high-level research question, several sub-questions are proposed, divided in the following categories:

Low-fidelity experimental data enhancement:

- *To what extent can the Physics-Informed Machine Learning methods reconstruct 3D flows from 2D experimental flow information?*
- *In pursuit of the previous research question, what is the error obtained in reconstruction of 3D mean velocity and static pressure?*

Flow field characterization and prediction of turbulent phenomena:

- *What degree of data-resolution is necessary for Physics-Informed Machine Learning methods to predict the locations of flow separation and reattachment over the Boeing-bump geometry at Mach 0.2?*
- *To what extent do mean velocity and static pressure data contribute to the reconstruction of 3D separated flow?*
- *To what extent can reconstructed flow fields support the prediction of surface shear stress topologies over the Boeing-bump geometry?*
- *To what Reynolds number can Physics-Informed Machine Learning Methods accurately model flow separation, reattachment and surface shear topology over the Boeing-bump geometry?*

These sub-questions stem naturally from the pursuit of the greater research question. The first two sub-questions will be answered by validating the predicted flow field with unseen experimental data. The additional two sub-questions can be answered through an ablation study of various levels of data-resolution. Sub-question 5 directs itself to practical applications of this research, where the ultimate goal is to come closer to accurately modeling surface shear stresses for the purposes of aerospace design. In pursuit of the previous sub-questions, a discussion will follow on the maximum Reynolds number at which these results are capable of being produced, answering the final sub-question.

The following report will be structured as follows, divided into 3 parts. Part 1 discusses the background of the investigation, beginning with the characteristics of the Gaussian Boeing bump (chapter 2), and Physics-Informed Neural Networks (chapter 3). Subsequently, the second part dives into the methodology (chapter 4) of this study, beginning with the experimental methodology, and followed by the machine learning methodology and results (chapter 5). Finally, part 3 discusses the closure of the study and contains the conclusion (chapter 6) and recommendations for future work (chapter 7).

Part I

Background

2

Characteristics of the Gaussian Boeing Bump

This chapter will serve to outline the recent advancements made on the Gaussian Boeing bump across various research institutions. An understanding of the experimental studies performed previously will allow this work to contribute to the wealth of experimental data with its in-house measurements, and aim to develop novel flow enhancement techniques. Additionally, understanding where previous computational campaigns have failed to model the separated flow phenomena over the geometry will highlight the candidacy for the PINN methodology.

2.1. Gaussian Boeing Bump Geometry

Despite the simplicity of the geometry, the Gaussian Boeing Bump remains an elusive problem for its propensity to generate complex 3D flow phenomena. Defined by the product of the tapered error function in the spanwise (z) direction, and the Gaussian function in the streamwise direction (x), the geometry takes the following form, shown in Equation 2.1:

$$y(x,z) = h\frac{1 + erf\left(\left(L/2 - 2z_0 - |z|\right)/z_0\right)}{2}\exp\left(-\left(\frac{x}{x_0}\right)^2\right) \tag{2.1}$$

with ratios defined as: $x_0/L = 0.195$, $y_0/L = 0.060$, and $h/L = 0.085$. A standardized width of $L = 0.914$ [m] has been chosen across the 3 experimental setups that have been commissioned, namely at the California Institute of Technology, University of Notre Dame, and University of Washington, whose previous results on investigation of the geometry encourage the work of this thesis [1, 2, 3]. The adherence to a unified scale of the geometry is important due to the fact that bump width, length and distance-to-ceiling all remain tune-able parameters that affect the outcome of the turbulent boundary layer and onset of separation.

The geometry of the bump can be visualized in Figure 2.1, where it is shown from a frontal angle pointing downstream, as well as a side view.

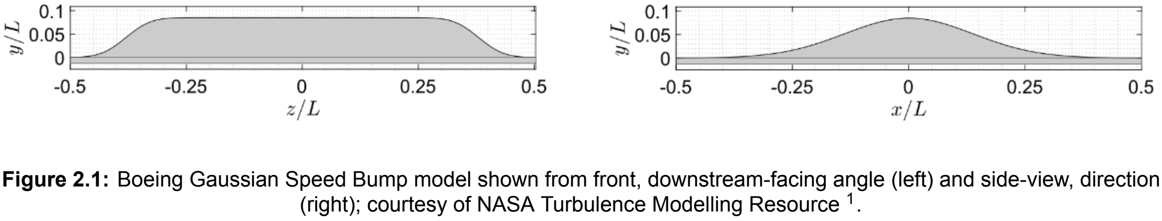


Figure 2.1: Boeing Gaussian Speed Bump model shown from front, downstream-facing angle (left) and side-view, direction (right); courtesy of NASA Turbulence Modelling Resource [1].

The interest of this geometry originated from the desire to produce a highly separated flow while avoiding interactions stemming from side-wall boundary layers. Previous campaigns studied geometries that mimicked flow over aircraft wing-body junctions wherein a high likelihood of flow separation was found to be present. Previously, the study of various 2D geometries conducive to flow separation were studied. These included the backward-facing step [4, 5], backwards-facing ramp [6, 7] and the proprietary NASA hump model [8]. These 2D geometries were found to be influenced by sidewall-junction flow physics which contributed significantly to separation behavior, and thus following the result of several iterations of geometry design, the Boeing bump geometry was obtained. This was the motivation for designing the bump with error-function shoulders, however the effect on generation of 3D flow phenomena in this region must not be neglected. The decoupling of sidewall influences on the bump-induced pressure gradients allowed for the independent analysis of the highly turbulent post-bump flow region, without influence of corner-flow phenomena. Furthermore, since the geometry results in both a favorable and adverse pressure gradient, the result produces a region of separated flow aft of the bump apex. With the aforementioned in mind, the geometry was designed with the intention of providing repeatable flow-separation phenomena applicable to both experimental and numerical studies.

Due to the difficulty in predicting separation, as well as flow phenomena within the separated and highly turbulent boundary layer, several academic institutions have taken an interest in its research and understanding. This chapter will discuss the various experimental and numerical campaigns that have been performed on the Gaussian Boeing-bump geometry by a variety of institutions. The goal of this is to lay a foundation of what research has been done, what results may be expected, and where the PINN methodology may provide utility.

2.2. Geometric Variation and Boundary Layer Development

Due to the need recreate the results of this experiment across multiple institutions, it is necessary to observe a consistent experimental setup in regards to the effects on incoming boundary layer development and eventual flow separation. This section will briefly comment on details related to the geometry and setup as it pertains to the boundary layer development.

Relating to its effect on boundary layer growth, the length of the splitter plate preceding the Boeing Gaussian bump geometry is maintained across the various experimental campaigns (discussed in section 2.4). The placement of the bump geometry within the greater context of its splitter plate can be visualized in Figure 2.2, where configurations A and B are depicted.

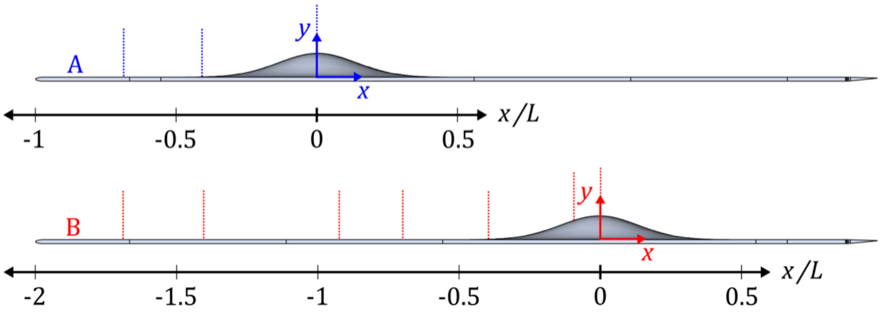


Figure 2.2: Variations of bump configuration in experimental campaigns [9], investigating the effect of boundary layer thickness on post-bump separation.

The baseline investigation of the bump geometry corresponds to configuration A, where the splitter plate precedes the bump apex by a full length unit $L = 0.914$ [m]. However, to investigate the effects of continued boundary layer growth, configuration B was also devised during the campaign of the University of Notre Dame. Continued investigation under various experimental methods allows for the comparison of incoming boundary layer growth and its effects on separation upon encountering the

[1] https://turbmodels.larc.nasa.gov/Other_exp_Data/speedbump_sep_exp.html

bump. One such comparison might investigate the importance of the ratio of boundary layer height δ to bump height h, (δ/h), on separation. For the remainder of this study, unless mentioned otherwise, splitter-plate configuration A will be assumed to be the configuration of interest, chosen for consistency with past experimental and numerical works. Furthermore, the study is aligned to two inflow conditions of $M = 0.2$ and $M = 0.1$, which aims to maintain uniformity with regards to the boundary layer development for each case.

Table 2.1: Boundary layer properties of incoming flow for both inflow conditions studied over bump configuration A [10].

M_∞	(x/L)	Re_L	C_f
0.1	-0.469	1.07$\cdot 10^6$	3.10$\cdot 10^6$
0.2	-0.469	2.11$\cdot 10^6$	2.60$\cdot 10^6$

An additional variable of relevance is the height difference from the splitter plate to the ceiling of the wind tunnel. This parameter effectively dictates the ”compression ratio” witnessed by the flow over the bump and will contribute to the strength of the FPG and APG in front and behind the bump apex, respectively. Consequently, this will affect the development, and particularly the separation and reattachment of the boundary layer.

While this variable remains more difficult to control (due to inherent differences in wind tunnel size between the University of Notre Dame, Washington and Caltech), its potential contributions to the flow behavior observed must be noted. As a result of the aforementioned factors, it can be expected to observe varying flow phenomena despite the use of a replica geometry and consistent inlet/boundary conditions. For this reason, the boundary layer properties as observed in previous works by [10] have been outlined in Table 2.1, allowing differences in eventual flow phenomena to be attributed to differences in boundary layer development.

2.3. NASA Langley Research Center

Before embarking on the individual contributions of each academic and industrial institution, the origin and support of this initiative must be discussed. In 2014, NASA published their CFD Vision 2030 Study [11], wherein they outline recommendations and strategies for developing improvements to CFD capabilities by the year 2023. Much of the report consists of the encouragement of interdisciplinary collaboration between industrial stakeholders and research partners, which at the time of writing has been largely realized through the joint efforts with parties that will be discussed in this chapter.

Additionally, the study establishes a set of 7 findings aimed at furthering the state-of-the-art in CFD and providing advancements in the fields of aeronautics and aerospace. Among their findings, the research performed on the Gaussian Boeing-bump can be linked to the following:

1. *”The use of CFD in the aerospace design process is severely limited by the inability to accurately and reliably predict turbulent flows with significant regions of separation.”*
2. *”HPC hardware is progressing rapidly and technologies that will prevail are difficult to predict.”*

Exactly a decade since the publication of their study, these two findings remain ever-present, with the latter motivating the current investigation of neural networks as candidates for flow modelling. Furthermore, these findings have encouraged and supported the research performed over the previous decade through the NASA Langley Research Center Turbulence Modelling Resource[2]. While the turbulence modelling resource and NASA’s overarching goals primarily target the advancements of CFD, they have necessitated novel campaigns in both numerical as well as experimental regimes, which will be the topic of discussion for the remainder of this chapter.

2.4. The University of Notre Dame

While the Boeing-bump geometry was originally conceptualized in 2015, tests only began at the University of Notre Dame in 2020 and culminated in 2023 after a 3-year long campaign. Performed as a collaborative effort between the University of Notre Dame’s experimental aerodynamics group and the

[2] `https://turbmodels.larc.nasa.gov/index.html`

CFD group at Boeing Research & Technology, the former coordinated the experimental campaigns to produce validation-capable benchmark data for future experimental and numerical studies. The latter, in parallel, performed validation studies using their proprietary Boeing CFD (BCFD) flow solver [12], whose methodology and results will be described in section 2.5.

The University of Notre Dame and Boeing Research & Technology's continued experimentation was performed at the Notre Dame Mach 0.6 closed circuit wind tunnel [13, 14, 10], where, flow measurement methods included the aforementioned oil-film Interferometry (OFI), as well as static pressure taps, dynamic pressure sensors and PIV (performed in both mono and stereo configuration) [9]. Through these various methods of documentation, flow separation and wall-shear stress was recorded over a range of velocities (Mach $\in [0.05, 0.2]$) and Reynolds numbers ($1.0 \cdot 10^6 \leq Re_L \leq 4.0 \cdot 10^6$). While the bulk of the data was generated in the Mach 0.2 regime for the purpose of validating compressible CFD solvers, the data obtained in this campaign will remain suitable for validation of the PINN model, described in chapter 3.

2.4.1. Fluorescent Oil Flow Visualization

Beginning with the results obtained using oil flow visualization, the authors confirm the signature "*owl face pattern*" [15, 16, 15, 17] being generated in oil flow visualization methods, shown in Figure 2.3. The flow phenomena outlined in the image obtained using oil flow visualization allowed the authors to identify regions of interest for spanwise PIV measurement, the results of which are discussed in subsection 2.4.2

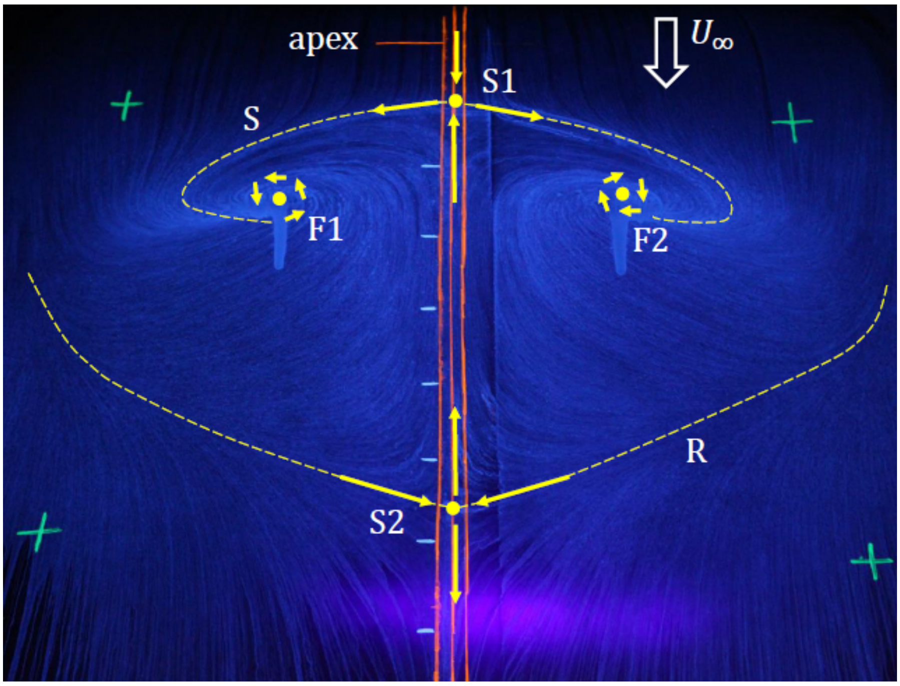


Figure 2.3: Characteristic "*owl face pattern*" obtained at flow velocity $M_\infty = 0.2$ using oil flow visualization of downstream flow. [9]

Visible in the figure, the "eyes of the owl" can be seen in locations F1 and F2, corresponding to the foci responsible for lifting vortices that ascend from the surfaces [18]. These vortices are met with the flow from S1 which indicates the bifurcation of flow (saddle point), downstream of which the flow is separated. S2 then corresponds to the area of flow reattachment further downstream. The authors attribute the results this flow behavior to the concavity of the bump, and the location of its inflection point from convex surface curvature to concave. A qualitative reconstruction of the above plot would mark a significant achievement in the PINN's capacity to model surface shear stresses over the geometry.

2.4.2. PIV Measurements

The quantitative representation, obtained through PIV, can be visualized in Figures 2.4 and 2.5, where multiple planes and angles demonstrate the mean velocity at the bump's inflection point. The streamlines can be seen in the streamwise direction, and produce the strongest level of separation at the center-line of the bump, corresponding to a z/L ratio of 0.0. Figure 2.4 confirms that the majority of these trials were focused at the center-line of the bump, providing an avenue for novel experimental campaigns to be performed at Caltech and to contribute to the turbulence resource for future validation.

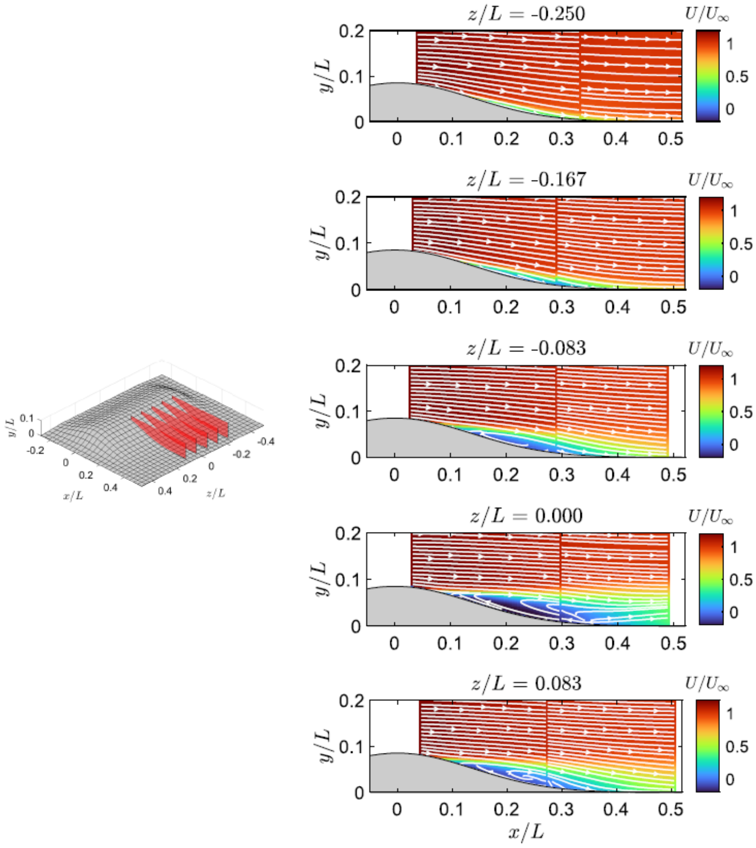


Figure 2.4: PIV performed at $M_\infty = 0.2$ showing mean velocity and streamlines displaying separation bubble growth over multiple spanwise planes.

As can be seen, flow reversal weakens towards the shoulders (max-span) of the bump. Furthermore, the z-plane velocity vector, W, shows the acceleration of flow from the edges towards the center in the cross-plane. This represents the out-of-plane velocity that is expected nearing the 3D contours (shoulders) of the bump. This will be a region of significant interest where numerical methods, and the neural network in particular are likely to struggle. This will also mark input training data for the 3D PINN network.

The next section will discuss the numerical counterpart of this investigation, carried out by Boeing Research & Technology. Noteworthy takeaways from this comparison shall address the ability to model separation, and reattachment as well as areas of peak turbulence following the bump apex.

2.5. Boeing Research & Technology

Of equal contribution to this turbulence resource is the numerical modelling aspect, much of which have been performed in parallel to the experimental campaigns at the University of Notre Dame. Furthermore, while the focus of this study is not to assess contemporary numerical modelling capabilities of turbulent flow phenomena, it is necessary to discuss the current state of the art to ultimately determine where contemporary solvers struggle, and identify where this study can contribute the most.

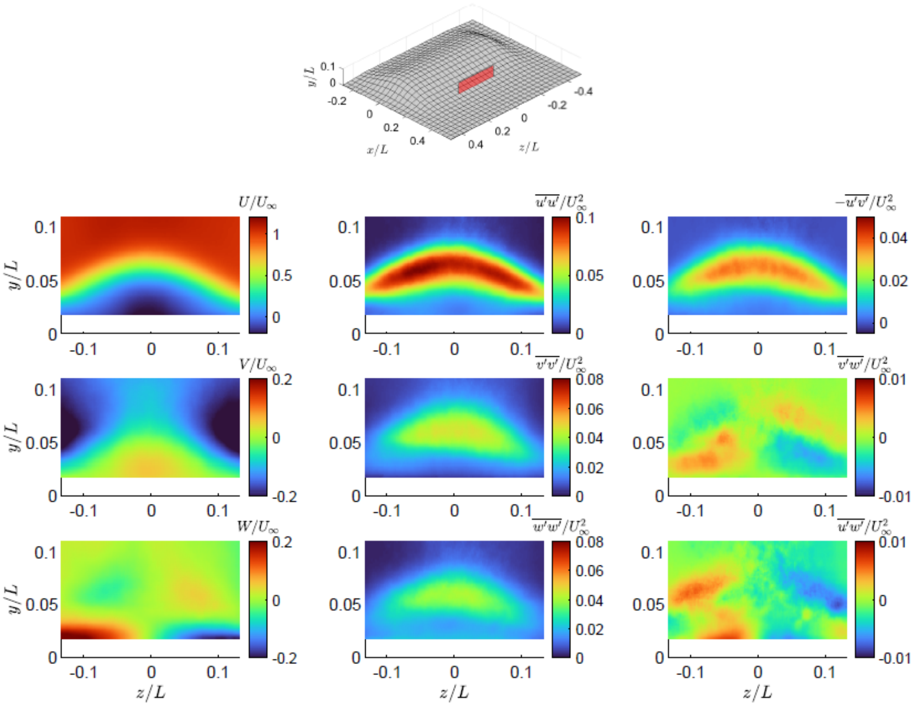


Figure 2.5: Mean velocity and Reynolds stresses in cross-plane recorded downstream of bump using Stereo PIV (SPIV).

The results and takeaways of the computational investigations by the Boeing CFD research group are compiled in the study by Lakebrink [19], wherein numerical comparisons against various experimental techniques are delivered. Furthermore, on the numerical aspect, multiple methods of varying computational complexity are employed to better discern the limitations of modelling techniques (including PIV, OFI, Hot-Wire Anemometry, Surface-Pressure and LDV). While these various data collection methods are important to the validation of CFD, this current project limits its interest to the study and improvement of PIV data. As such, this shall be the primary measurement technique discussed for use over the Boeing bump.

2.5.1. Numerical Comparison with PIV Measurement

Beginning with PIV, comparisons were performed against CFD using RANS (Reynolds Averaged Navier Stokes) as well as DDES (Delayed Detached Eddy Simulation) simulations. The results are displayed below comparing both u and v velocity vectors in Figures 2.6 and 2.7, respectively.

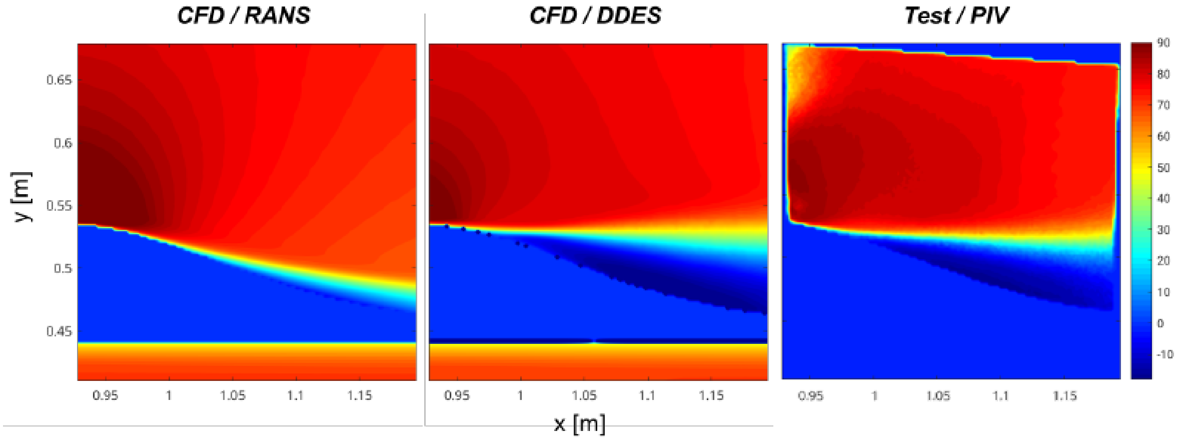


Figure 2.6: Comparison of u-velocity at center-line, aft of bump-apex using RANS CFD, DDES CFD and PIV [19].

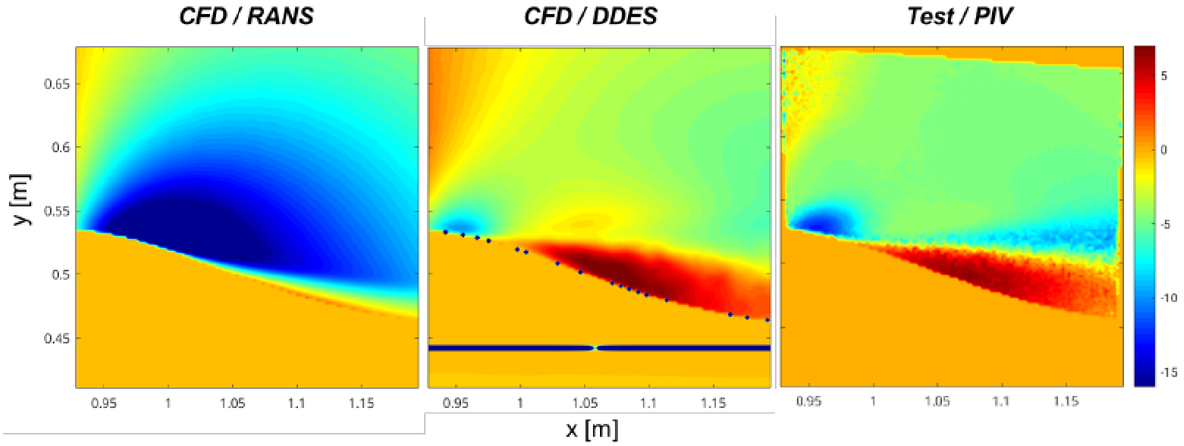


Figure 2.7: Comparison of v-velocity at center-line, aft of bump-apex using RANS, DDES and PIV [19].

As one can see, the discrepancies found using RANS are larger than that obtained using DDES, showing an inability to accurately describe the advent of separation after the bump. Conversely, DDES performs well, though the authors comment a slight over-prediction of the shear layer and an under-prediction of downwash, which they conclude to mark a pre-mature prediction of separation.

The authors of these works have critically shown where contemporary turbulence models fail when compared to experimental results, citing the need for further research to be performed on this topic. This continues to motivate an encouraging gap in the research, where a middle ground between numerical and experimental methods could apply through machine learning. This marks the purpose of the work performed at the California Institute of Technology, aiming at a solution which correctly predicts the region of separation, without the computational effort of an LES scheme.

2.6. Additional Contributors

Before this chapter is concluded, some attention is directed to additional contributions in the field of Boeing-bump research. While the majority of research has been conducted by the aforementioned institutions in direct collaboration with Boeing, some independent works have been produced. They provide a more recent overview of the results in Boeing-bump research and outline current gaps of knowledge in the space. Consequently, their conclusions will assist to inform of potential research directions that this literature review sets the baseline for. This study investigates the behavior of turbulent boundary layer flow over a Gaussian-shaped bump using Direct Numerical Simulation (DNS). By resolving all relevant turbulence scales, the authors provide a detailed dataset for studying pressure-gradient-induced turbulence and for validating computational models of turbulent flows. Most relevant to this investigation are the results on the friction coefficient over the bump surface, displayed in Figure 2.8.

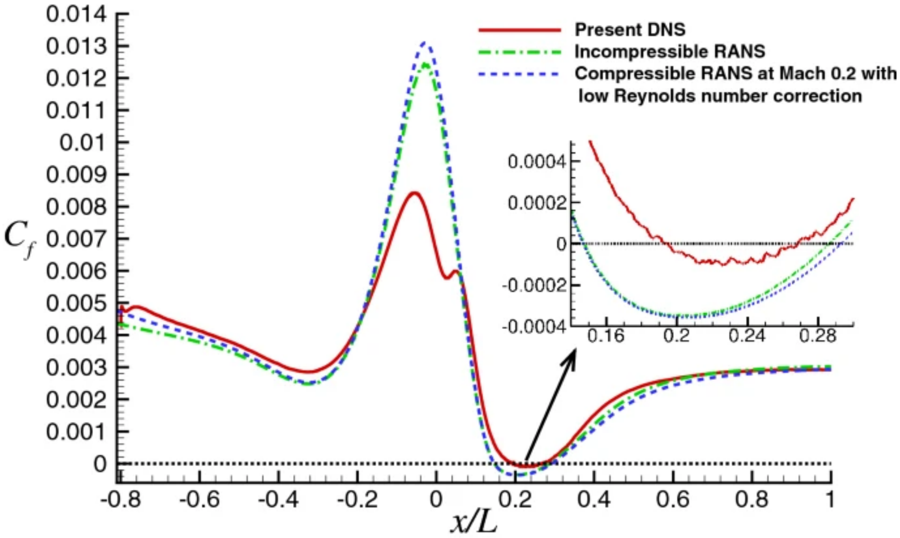


Figure 2.8: Skin friction coefficient observed over Boeing bump surface as modeled with DNS and RANS methodologies [20].

As shown, the friction coefficient, C_f, exhibits significant variation due to the changing pressure gradients along the surface. Upstream of the bump, within the region subjected to favorable pressure gradients, C_f increases as the flow accelerates. This reflects the thinning of the boundary layer and the enhancement of wall shear stress due to turbulence suppression in the presence of an FPG. As the flow encounters the adverse pressure gradient downstream, C_f decreases sharply. This reduction is indicative of boundary layer thickening and the onset of flow separation. In the separation region, wall shear stress diminishes to near-zero values, leading to minimal or negative C_f. Beyond the separation point, in the reattachment zone, C_f experiences a rapid increase. This rise corresponds to the recovery of wall shear stress as the flow reattaches and the boundary layer begins to redevelop. The steep gradient of C_f in this region highlights the transition from separated to attached flow.

Overall, the distribution of C_f is highly asymmetric due to the non-equilibrium pressure gradients. The adverse pressure gradient has a particularly strong influence, extending far downstream and significantly altering the wall shear stress distribution. These results demonstrate the importance of C_f as a diagnostic tool for identifying flow separation and reattachment, providing valuable insights for aerodynamic surface optimization. In pursuit of modeling surface shear stresses over the Boeing bump geometry, the PINN will be assessed for its ability in recreating comparable friction trends as shown by Ali et al.

In conclusion, this chapter has marked several recent development surrounding the Boeing Bump Geometry that provide strong indication for the inability of RANS-based numerical solvers to adequately model the flow features in the aft-bump region. The existence of substantial experimental data surrounding the geometry points to an opportunity to include machine learning based methodologies to provide an improved reconstruction of the flow field. The subsequent chapters will outline recent works where PINNs have been used in conjunction with experimental data, building upon its use-case as a contemporary flow enhancement method.

3
Current State of Physics Informed Neural Networks

Advancements in machine learning techniques have prompted novel approaches to numerical solutions of partial differential equations (PDEs). In specific, Physics-Informed Neural Networks have shown significant applications in a variety of fields including fluid mechanics for their ability to incorporate experimental data with physical constraints. Figure 3.1 [21] displays the exponential growth experienced in scholarly articles citing Physics-Informed Neural Networks following their original introduction by Raissi et al. in 2019 [22].

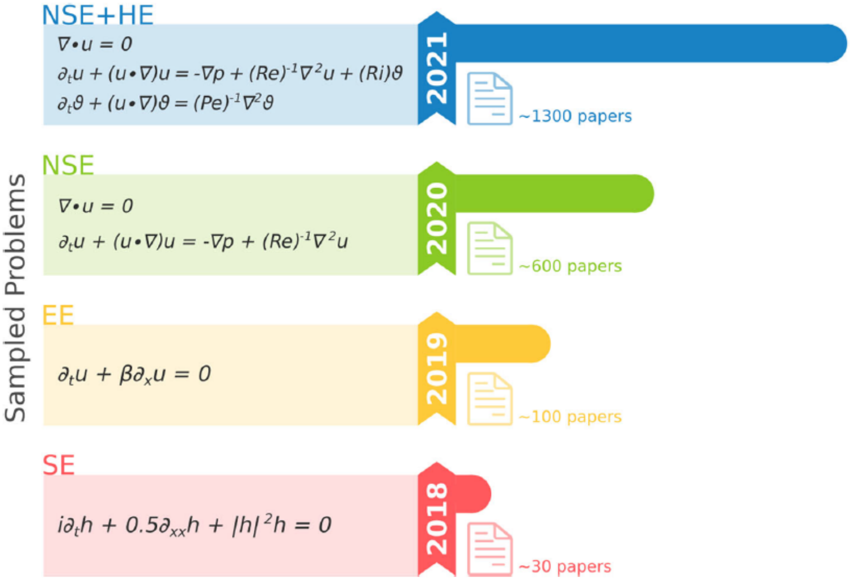


Figure 3.1: Exponential growth of literature citing PINNs applied to solution of: Schrödinger Equation (SE), Euler Equation (EE), Navier-Stokes Equations (NSE), and Heat Equation (HE) [21].

Recent examples include reconstruction of flow over an Espresso cup using Schlieren imaging [23], modelling two-phase flow and solutions to convective heat transfer problems [24], and enhancement of dense velocity and pressure field reconstruction from PIV data [25]. The latter of these is of most relevance to this project, and consists of the augmentation of the Physics-Informed Neural Networks with Reynolds Averaged Navier-Stokes (RANS) equations to resolve highly turbulent flow fields. Often, these are supplemented with a turbulence model to 'close' the RANS equations, however recent developments have also shown that PINNs can intuitively interpret the Reynolds stresses when not provided with a turbulence model. Nevertheless, much of the current development of PINNs applied to experimental fluid mechanics is still in its infancy, and limited to 2D, inviscid, laminar or low-Reynolds flow. In the context of real flows and more complex geometries, such as the Boeing bump, it is of interest to accurately model turbulent 3D flow phenomena, including separation/re-attachment. In the case of high velocity flows, shock wave perturbations are also of interest, and although out of the scope of this project, it remains to be seen whether PINNs can successfully improve the reconstruction of such intricate phenomena.

Due to the abundance of literature surrounding PINNs, the majority of this overview will retain this focus, however where relevant, the discussion of competing neural network architectures will also be included. The following section will give a summary of the PINN architecture, after which its applications to a variety of problems in fluid mechanics will be discussed.

3.1. PINN Architecture

Initially introduced by Raissi and collagues [22], the PINN consists of a Deep Neural Network (DNN) with an additional loss function based on the residuals of governing equations. This allows the enforcing of the DNN to adhere to physical equations in addition to data and boundary conditions. Apart from this contribution to the loss function, the general architecture of the PINN remains largely unchanged from the underlying Multi-Layer Perceptron (MLP), which, applied to the Reynolds Averaged Navier-Stokes equations takes the following form:

$$(\mathbf{U}, p, \overline{\nu}) = f_{NN}(\mathbf{X}; \Theta) \tag{3.1}$$

with $\mathbf{U} = (u, v, w)$ and p denoting the velocity and pressure fields respectively, $\overline{\nu}$ denoting a turbulence parameter, and f_{NN} denoting a deep neural network with $\mathbf{X} = (x, y, z)$ spatial input coordinates and Θ trainable network parameters.

As with any MLP, the network size is defined by two parameters: N_{layer} and H which denote the number of layers of the neural network, and the depth of each node, respectively. Propagation through the network from input variable x_i to output variable f_j is performed via:

$$f_j = \sigma(w_{i,j} \cdot x_i + b_j) \tag{3.2}$$

where $w_{i,j}$ and b_j denote the weights and biases respectively. The hyperparameter σ represents the activation function, which is user-defined. An activation function of $y = \tanh(x)$ is recommended for PINNs due to its continuous differentiability over the entire domain [26].

The primary difference between PINNs and a conventional MLP lies with the loss function, where in PINNs it is augmented with a physical loss which encourages the network to optimize its parameters to satisfy one or more formulae. In the case of a PINN applied to the solution of the Reynolds Averaged Navier-Stokes equations, the following loss functions would be present for the network to minimize:

$$\mathcal{L}_{data} = \sum_{i=1}^{N_{data}} |\mathbf{U}_{data}(\mathbf{X}^i) - \mathbf{U}_{pred}(\mathbf{X}^i)|^2 \tag{3.3}$$

$$\mathcal{L}_{BC} = \sum_{i=1}^{N_{BC}} |\mathbf{U}_{BC}(\mathbf{X}^i) - \mathbf{U}_{pred}(\mathbf{X}^i)|^2 \tag{3.4}$$

$$\mathcal{L}_{PDE} = \sum_{i=1}^{N_{PDE}} |\mathcal{L}_i(\mathbf{X}^i)|^2 \tag{3.5}$$

Here, $\mathbf{U}_{data}$ and $\mathbf{U}_{pred}$ are the measured and predicted velocity fields, respectively. Additionally, $\mathcal{L}_{data}$, $\mathcal{L}_{BC}$, and $\mathcal{L}_{PDE}$ represent the losses to data, boundary conditions and supplied partial differential equations, respectively. In a PINN, and contrary to a standard MLP, the PDE loss will be provided to the data and boundary condition losses as in the following equation:

$$\mathcal{L} = (1 - \alpha) \cdot (\mathcal{L}_{data} + \mathcal{L}_{BC}) + \alpha \cdot \mathcal{L}_{PDE} \tag{3.6}$$

Where α represents a weighting factor that controls the influence of the PDE loss on the remaining loss terms. The output of each of these loss parameters is determined through the mean-squared error function, $\mathcal{L}_{\text{PDE}} = \Sigma_{i=1}^{i=n}\text{MSE}(\epsilon_i)$, where ϵ_i represents the residual error for the given PDE provided to the PINN. In the context of RANS PINNs, the following residuals are of importance:

$$\begin{aligned} \nabla \cdot \mathbf{U} &= \epsilon_1 \\ \rho(\mathbf{U} \cdot \nabla \mathbf{U}) + \nabla P - \mu \nabla^2 \mathbf{U} - \nabla \cdot (-\rho \langle \overline{\mathbf{u}'\mathbf{u}'} \rangle) &= \epsilon_2 \end{aligned} \tag{3.7}$$

The first equation with residual ϵ_1 denotes the continuity equation while the second equation with residual ϵ_2 denotes the Reynolds-averaged momentum equations, where $\rho\langle\overline{\mathbf{u}'\mathbf{u}'}\rangle$ denotes the Reynolds stress tensor. Notably, the time-dependent derivatives are omitted since the analysis in this work is performed at steady-state. Differentiation across the spatial domain is performed through automatic differentiation [27] with ∇ representing the differential operator and ∇^2 representing the Laplacian.

A schematic of the PINN architecture is shown in Figure 3.2, which depicts the feed-forward network from input values to output values. After obtaining predicted output values, automatic differentiation is performed and the PDE residuals are determined. Finally, the data, boundary conditions and PDE residual losses are backpropagated and the training loop continues.

A modification to the typical MLP architecture that has been shown to benefit PINNs is the inclusion of Fourier features [26]. This extra step before feeding input parameters to the hidden layers can compensate for the inherent spectral bias that PINNs suffer from, and allow the network to more accurately represent high-frequency solutions [28, 29]. In such an implementation, the positional input data is passed through the following encoding before inclusion into the MLP network:

$$\gamma(X) = \begin{bmatrix} \text{cos}(\mathbf{BX}) \\ \text{sin}(\mathbf{BX}) \end{bmatrix} \tag{3.8}$$

With $\mathbf{B}$ representing a Gaussian sampling distribution: $\mathcal{N}(0, \sigma^2)$ with σ being a user-specified parameter: $\sigma \in [1, 10]$

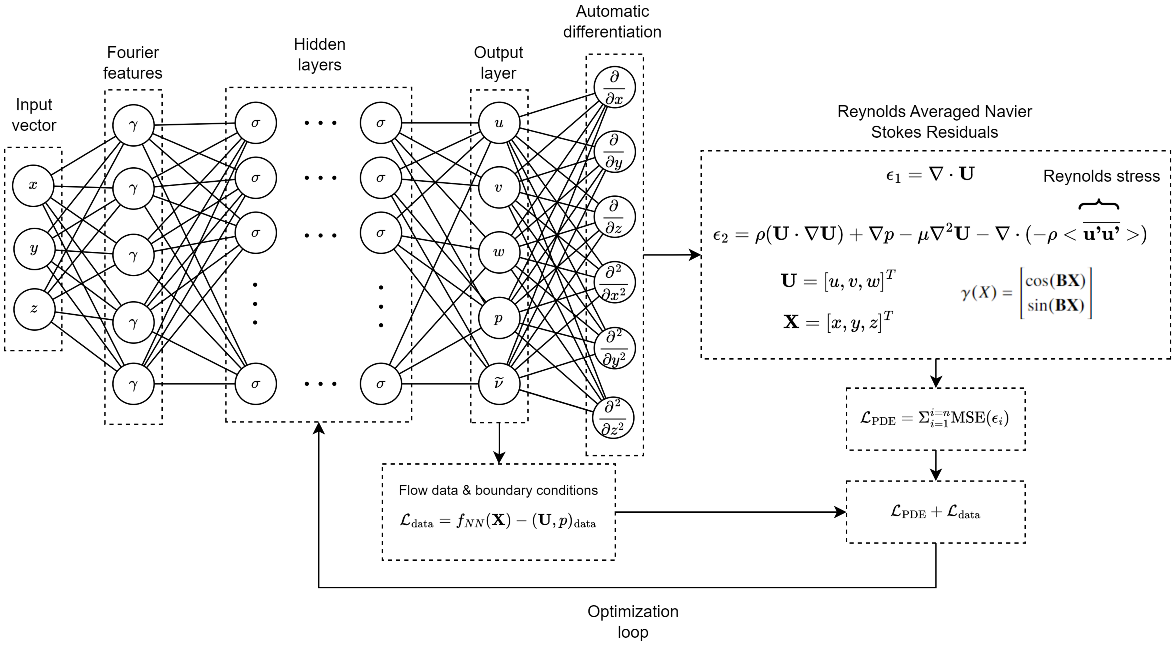

Input vector
x
y
z
Fourier features
γ
Hidden layers
σ
...
Output layer
u
v
w
p
$\tilde{\nu}$
Automatic differentiation
$\frac{\partial}{\partial x}$
$\frac{\partial}{\partial y}$
$\frac{\partial}{\partial z}$
$\frac{\partial^2}{\partial x^2}$
$\frac{\partial^2}{\partial y^2}$
$\frac{\partial^2}{\partial z^2}$
Reynolds Averaged Navier Stokes Residuals
$\epsilon_1 = \nabla \cdot \mathbf{U}$
Reynolds stress
$\epsilon_2 = \rho(\mathbf{U} \cdot \nabla\mathbf{U}) + \nabla p - \mu\nabla^2\mathbf{U} - \nabla \cdot (-\rho < \overbrace{\mathbf{u'u'}} >)$
$\mathbf{U} = [u, v, w]^T$
$\mathbf{X} = [x, y, z]^T$
$\gamma(X) = \begin{vmatrix} \cos(\mathbf{BX}) \\ \sin(\mathbf{BX}) \end{vmatrix}$
$\mathcal{L}_{\text{PDE}} = \Sigma_{i=1}^{i=n}\text{MSE}(\epsilon_i)$
Flow data & boundary conditions
$\mathcal{L}_{\text{data}} = f_{NN}(\mathbf{X}) - (\mathbf{U}, p)_{\text{data}}$
$\mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{data}}$
Optimization loop

Figure 3.2: Schematic diagram representing PINN architecture applied to 3D Reynolds Averaged Navier Stokes equations. Diagram is produced specific to the work of this thesis and displays the PINN architecture used in the 3D analysis.

Despite the simple architecture of PINNs, there have been significant efforts to further improve upon the MLP architecture to support its functions for forward and inverse solutions within fluid mechanics. This next section will discuss relevant instances where PINNs have been applied to fluid mechanics problems, and discuss difficulties in achieving convergence to a globally physically compliant solution.

3.2. Applications to Fluid Mechanics

Since their original introduction, much of the interest of PINN application has been directed to the solution of partial differential equations for fluid mechanics. Often, the successes of PINNs and their variations are measured in benchmarks that assess the solution of the Burgers equation [30], or Navier Stokes equations [22].

In an implementation with the Navier-Stokes equations, Lai et al. [31] have introduced a novel methodology, combining high-frequency pressure probe and sparse PIV-data in an attempt to produce high-temporal and spatially resolved flow fields. This is of particular interest to the current project, since both static-wall pressure and PIV measurement techniques are available at Caltech's Lucas wind tunnel. While their work was produced using synthetic data, their conclusions with regards to non-periodic flows suggest the complexity of turbulent flows may not be out of reach for PINNs to reconstruct to a sufficient degree of accuracy. The authors do conclude that due to their methodology being conducted at laminar flows ($Re = 100$), the need remains to further investigate the suitability of PINNs to reconstruct turbulent flows.

Furthermore, as the previous section has outlined, despite the implementation of PDEs as governing equations, ML methods remain data-driven, and as such their success in being applied to problems in fluid mechanics relies on the availability of data. While the source of this data can vary, e.g. computational vs experimental, Particle Image Velocimetry (PIV) results serve as excellent candidates for this for their widespread availability in literature and spatial sparsity. This has been noted in the literature, with a significant number of publications investigating the potential for this collaboration of approaches. The next subsection will be dedicated to outlining the recent results obtained from integrating PIV results into the PIML methodologies.

3.2.1. Applications to PIV Data

Despite narrowing the field of search to PIML methods applied to PIV data, the available literature remains vast and diverse. For instance, much work is dedicated to the investigation of PIV on hydrodynamics [25] and hemodynamics [32]. In many cases, the takeaways from these works are substantial, where the former investigation used TOMO-PIV to assess the flow over a hemisphere submerged in water and implemented a PINN to successfully identify vortices produced in the turbulent regime. Authors noted a successful reconstruction of DNS simulations, albeit limited to larger scales. Furthermore, in the latter study on hemodynamics, PINNs were successfully applied to the modelling of wall shear stresses in near-wall blood flow. These results were produced despite the lack of boundary condition information supplied to the PINN. The authors celebrate the potential for PINNs in improving hemodynamic modelling for the purposes of cardiovascular disease diagnosis. Despite these apparent victories, the relative viscosities of water and blood remain far higher than that of air, and these successes do not necessarily comment on the abilities for PIML methods to translate to air-based turbulent flow phenomena. In the coming studies, this will be an important metric to distinguish, as the Reynolds number remains highly conducive to the PINN's successes and failures.

Applications for Reynolds numbers below 10,000

In a study on the applicability of PINNs to reconstruct sparse PIV data [33], researchers showed promising ability at recreating flow fields as well as pressure fields. Notably, despite no information being given about pressure, the PINN remains capable of reconstructing the pressure field through the constitutive relations of the Navier-Stokes equations. They extend this by establishing that the PINN can reconstruct flow fields of obstructed frames and missing flow volumes. Their results are shown in Figure 3.3, where varying quantities of experimental dataset are removed from training.

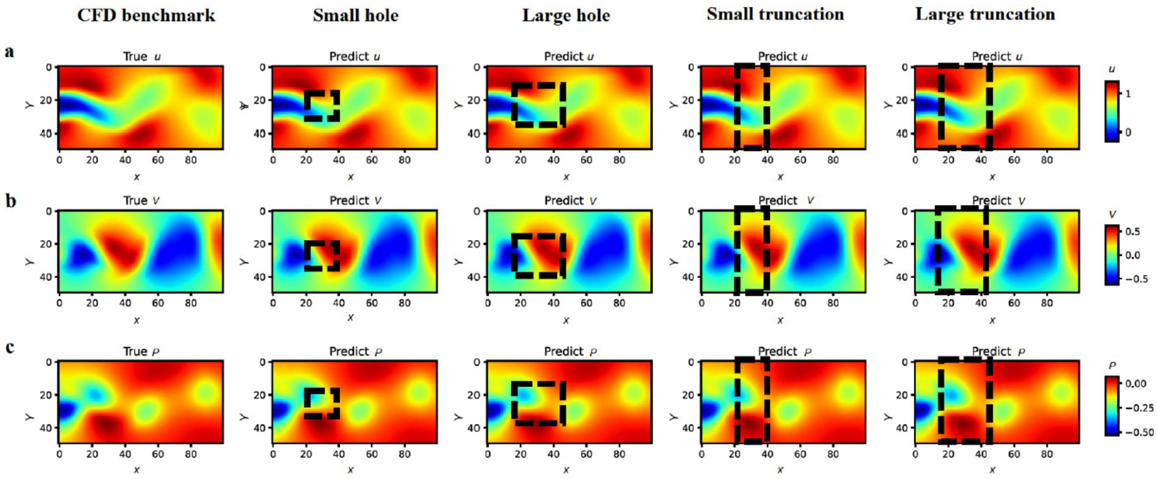


Figure 3.3: Comparison of PINN prediction performance with varying levels of truncated data removed; (a) streamwise velocity component; (b) spanwise velocity component; (c) pressure field [33].

As the figure shows, there lies significant capability in PINNs at reconstructing missing flow fields when given sufficient data. Regrettably, however, the investigation was only performed under the regime of incompressible flow at low Reynolds numbers in the order of $Re_\theta = 10^3$, and work at higher Reynolds numbers remains necessary to draw further conclusions.

In another work, researchers compared the ability of PINNs to reconstruct pressure fields compared to contemporary pressure reconstruction methods [34] over a variety of cases. Their results showed that when compared with traditional methods, PINNs exhibit the lowest overall pressure error, despite being trained on noisy data. They note that PINNs can correct noisy velocity data through the imposition of PDEs, and exploit this optimized velocity field for their pressure predictions. Another interesting takeaway concluded by the authors is the reliance of the PINN method on N_{cell}, N_{eqns}, and N_{data}. While increasing all of these parameters can positively affect the PINNs accuracy, the parameter of greatest influence remains N_{data}. This reinforced the idea that PIML methods remain data-driven in nature. This notion will be repeated in studies at higher turbulence levels in the subsequent section.

In a separate study, researchers refrained from using a PINN, and instead used an Artificial Neural Network (ANN) to investigate the recovery of incomplete PIV data of flow over a cylinder [35]. Authors credit the ANN in allowing the PIV window to be extended beyond experimentally obtained data. Additionally, while it is interesting that an alternative to PINNs was explored, the authors do not elaborate on why this choice was made. Nevertheless, this investigation gives credence to the growing trend of applying ML methods to the enhancement of fluid mechanics data. Although this simulation was performed over a flow with a Reynolds number of $Re = 8,000$, the authors encourage the continued implementation of machine learning methods, particularly for cases when only sparse data is available.

Applications for Reynolds numbers above 10,000

In a RANS-approach, the provision of a turbulence model was not incorporated in the PINN [36], and the authors outline impressive agreement in the PINN's prediction of Reynolds stresses. In this method, the PINN interpolates its own turbulence model that best matches the flow, which for one experimental case was stated to equal $Re = 0.2$ million [-]. The authors note excellent agreement between the results and the reference data, which marks an encouraging milestone for PINN solutions to turbulent data. The authors note several avenues of further research, including resolution of the near-wall boundary layer region which is seldom captured in traditional PIV systems. Reconstructing the viscous sub-layer and as a result predicting wall-shear stresses are stated by the authors to be further developments of this work.

In line with this desire to model the viscous sub-layer, another study has incorporated data surrounding the wall shear-stress over a backwards-facing ramp (diffuser) geometry into a PINN model [37]. This work is interesting, both for its incorporation of a ramp-like geometry which exhibits flow separation and reattachment, but also for its attempt to simulate the wall shear-stresses. They describe that despite the imperfections in experimental measurement of the near-wall boundary layer, the PINN is able to compensate in accuracy through its implementation of the 3D Navier Stokes equations. They comment on good agreement of the near-wall velocity profiles when operating solely on the Navier-Stokes equations, with additional loss terms (pressure, shear-stress) contributing little or negatively to the final predictions. They summarize by saying that when devoid of near-wall velocity data, the PINN struggles in computing the magnitude of the velocity gradient in the viscous sub-layer.

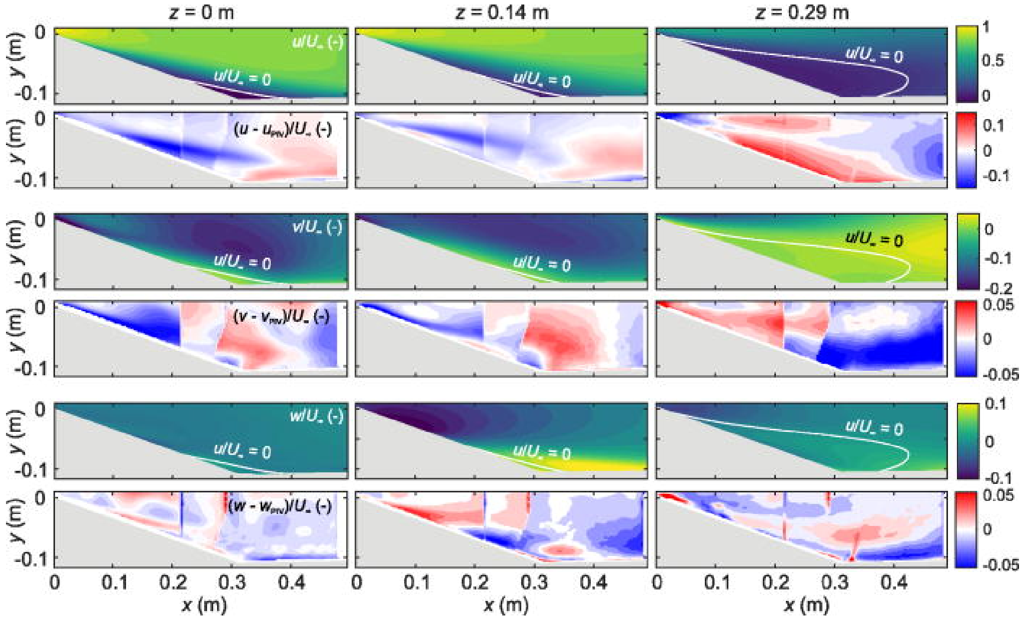


Figure 3.4: Comparison of predicted and measured velocity fields for each velocity component. [37]

Figure 3.4 displays the results from this work, where the predicted velocity fields by the PINN are shown for each velocity component. The predictions are compared to the PIV data, and shows impressive agreement at outlining the region of separation of the flow. Due to this work's relevance to that of the Boeing-bump, it will remain an interesting reference point over the course of this project.

Finally, a work assessing the ability of PINNs to reconstruct experimental flow data over periodic hills for a range of turbulent Reynolds numbers (up to $Re = 37,000$) corroborates reasonable predictions of an APG boundary layer [38]. The authors limited the PINN's exposure to 6 data points of the inner boundary layer and 13 points within the outer boundary layer. Reattachment locations were predicted with good agreement to experimental results, though friction coefficient C_f magnitudes remained areas where the PINN struggled. They go on to assess the recirculation zone, whose volume was well predicted, but internal velocity field was unsatisfactorily modelled.

An interesting conclusion that the authors arrive at is that available data remains the most crucial variable in predicting PINN success. With regards to their setup, they recommend that experimental data must be obtained from the region of interest (e.g. recirculation zone) to obtain the desired improvements in prediction accuracy. This brings important relevance for the project for this literature review, since they assert that PINNs cannot improve understanding of regions of flow that it is not provided data on.

To conclude on the broader trends of PINN work applied to PIV data, it is evident that PINNs remain data-driven in their methodology, and their success is directly dependent on the quantity of data they are fed. For this reason, many of the mentioned use cases limit the PINN method to works of interpolation and do not study data-deficient problems where PINNs may propose strong use-cases. This work will aim to break new ground on the topic, by assessing the prediction accuracy by training on experimental data at Reynolds numbers not examined previously in PINN works.

Part II

Methodology & Results

4 Methodology

This chapter will serve to describe the overall methodology used in obtaining experimental data, as well as developing the 2D and 3D PINN models used for testing. For an overview of how the experimental campaigns and machine learning methodology coincide, Appendix A provides a flowchart as a visual aid. Section 4.1 described the methods used to obtain experimental data at Caltech's Lucas Wind Tunnel, while Section 4.2 describes the work done to produce the 2D and 3D PINN models.

4.1. Experimental Data Collection

This section will document the efforts made in Caltech's Lucas Wind Tunnel to obtain experimental velocity and pressure data over the Boeing bump geometry. The focus of this campaign was to produce PIV data to contribute to the open repository of experimental and numerical results hosted on the Nasa Turbulence page[1].

Despite the efforts directed at the experimental campaign producing internal results over the Boeing bump, it must be acknowledged that these were not used in testing and validating the Neural Network (discussed in section 4.2). The reason for this arose from the need to begin NN training while preparing the wind tunnel environment. These were pursued in parallel, and as such delays in experimentation did not adversely affect neural network development. Additionally, the wind tunnel test sections (e.g. cross-section dimensions) vary substantially between the academic institutions of Caltech, University of Washington and University of Notre-Dame. For this reason, inevitable differences in experimental results have been and are likely to arise, and a consistent selection was needed throughout the machine learning training process.

Due to the contribution of these experimental results to the experimental Boeing bump repository hosted by NASA[2], as well as contribution to further works by colleagues at Caltech and Boeing, it felt pertinent to include a discussion of the methodology. Their availability can thus encourage continued work integrating experimental data into the PINN methodology to further improve on the results shown in section 5.2.

4.1.1. Experimental Setup

This section will describe the aspect of experimental measurements that aimed at collecting PIV and static pressure data of the flow over the Boeing bump. Due to the goal of obtaining a pipeline that inputs and outputs velocity fields, it was logical to investigate PIV as an input for the PINN. Furthermore, the established literature supports the ability of PINNs to successfully integrate PIV and pressure data into their training (see chapter 3).

An image of the experimental setup as used in the Lucas Wind Tunnel facility at Caltech is shown in Figure 4.1. The setup is designed for compatibility with modern flow measurement techniques, including

[1] https://turbmodels.larc.nasa.gov/index.html

[2] https://turbmodels.larc.nasa.gov/Other_exp_Data/speedbump_sep_exp.html

an acrylic (left) wall to allow for optical access for PIV. The back wall is covered in non-reflective tape to allow for better imaging contrast and minimize reflectivity, particularly laser reflectivity as encountered in PIV. A similarly non-reflective (and heat-resistant) tape is applied along the bump, to indicate where PIV imaging should take place. Figure 4.1 displays only a sample image, and the true measurement location was chosen with previous experimental campaigns in mind. Given the wealth of previous data available, a novel, but comparable, setup was sought to further contribute to the availability of data surrounding the geometry. Much of this concerns the 3D region at the shoulders of the bump, due to the propensity for out-of-plane flow phenomena and the associated difficulty of accurately capturing such flow behavior.

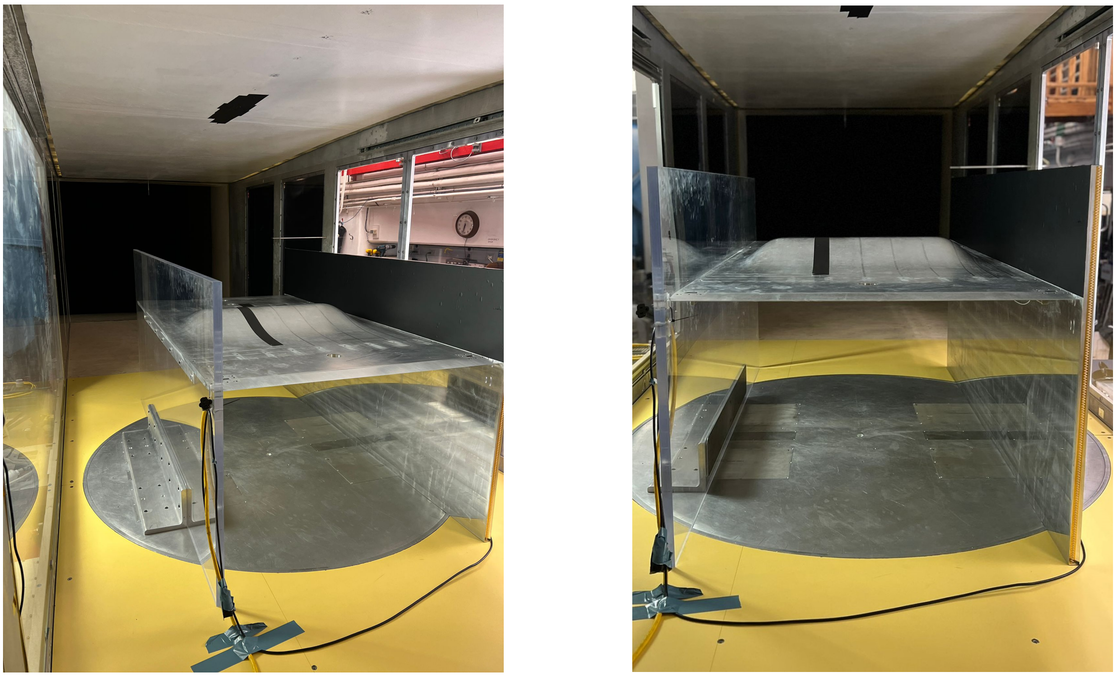


Figure 4.1: Boeing Gaussian Speed Bump model (L = 0.914 m) installed in the Lucas Wind Tunnel.

Also shown in the images is the turntable setup, allowing for rotation of the entire bump setup. Despite this, to mirror previous experimental and numerical campaigns, the setup is inserted with the walls parallel to the flow. The setup also accommodates circular windows along the center line to provide additional access for lasers that a PIV campaign may require. The direction of the camera faces the upstream direction, pointing into the wind-tunnel contractionary zone. The bed of the setup extends symmetrically to a splitter plate with a fixed leading edge and adjustable trailing-edge flap (fixed at 0° deflection.

Finally, the images also display the curved ceiling of the wind tunnel, where the adaptive ceiling allows for modification of the wind tunnel height at various streamwise locations (adapting the ceiling geometry) to accommodate a consistent streamwise static pressure distribution throughout the test section. This is a notable feature of the Lucas wind tunnel, but evidently not conserved across previous experimental campaigns. For this reason, the possibility of mounting the bump and splitter-plate upside down will be investigated in ongoing experimental campaigns. This will also allow the channel to be closed on the sides of the bump, as opposed to its current configuration where the walls do not extend to the ceiling and make for an unconstrained test section.

4.1.2. Particle Image Velocimetry

The PIV experiments were conducted over the Boeing bump using a conventional setup depicted in Figure 4.2. A single imaging device (Imperx B3420M) was used, which could contribute a maximum frame rate of 42 FPS, which, by appropriate timing of laser pulses, was able to capture frames at flow velocities up to 20 [ms^{-1}] [39]. The limitations imposed by the Lucas wind tunnel's maximum flow velocity, as well as the available frame rate of the imagine devices prevented measurements above a flow velocity of 20 [ms^{-1}], further contributing to the reliance on pre-existing data by courtesy of the previous collaborations with Boeing's Research & Technology group. The light sheet was provided by a Quantel Evergreen 532 [nm] laser which was expanded using a series of converging-diverging lenses from ThorLabs.

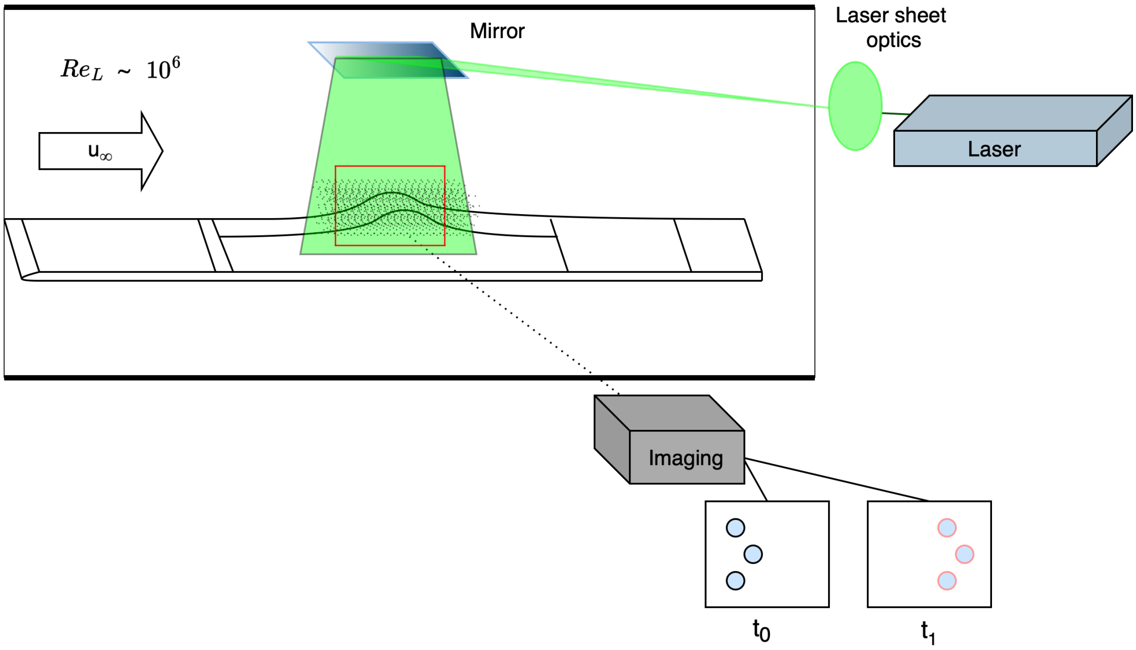


Figure 4.2: Schematic of PIV setup as conducted in Lucas Wind Tunnel.

Post-processing of the raw image data was performed using PIVLab [40], capable of efficiently analyzing and averaging multiple frames in parallel. A total of 200 frames were analyzed to overcome the influence of any statistical outliers captured in individual frames.

4.1.3. Pressure Tap Data Collection

Since the PINN can accommodate a variety of data types, static pressure data collection was also prioritized within the Lucas Wind Tunnel.

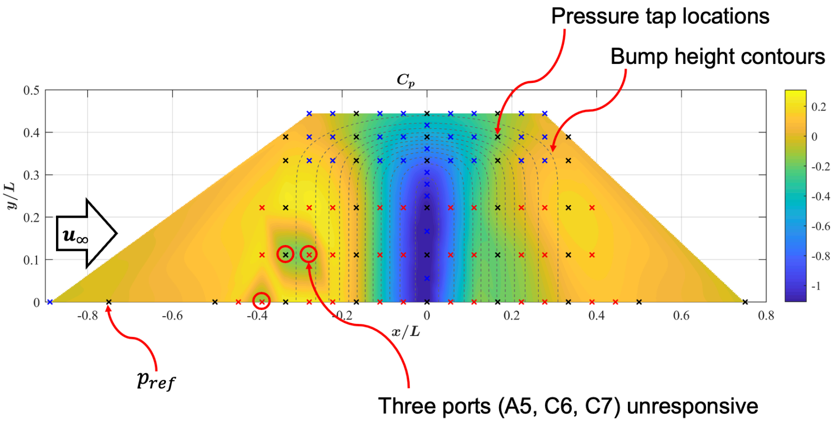


Figure 4.3: Location of static pressure taps where measurements were taken in Lucas Wind Tunnel [41].

Figure 4.3 displays the locations over the bump surface where static pressure taps are located. Oshima and colleagues [41] performed the initial pressure measurements displayed above, while additional pressure data was taken to conform to the PINN study. The locations are color coded red, blue and black, corresponding to the Scanivalve tubing outputs, where a separate key denotes the exact locations in 3D space. The pressure values are denoted in terms of pressure coefficient, computed using the relation in Equation 4.1.

$$C_p = \frac{p - p_{ref}}{\frac{1}{2}\rho_\infty u_\infty^2} \tag{4.1}$$

Since the machine learning methodology was performed in parallel to experimental measurement in the Lucas Wind Tunnel, the PINN model begun training on previously obtained data at the Universities of Notre-Dame and Washington, publicized by the NASA Turbulence Page [3]. This is an important distinction to make, since the measurement techniques implemented at Caltech serve to support future investigations in machine-learning enhanced studies. For this reason, it is of benefit to be able to iterate the PIV planes, and obtain a more robust distribution of static pressure ports than what was previously made available. Furthermore, in the interest of consistent training data, only data from unique campaigns were used for training to prevent conflicts between the boundary layer development across different experimental campaigns. The next chapter will discuss the machine learning methodology which was undertaken in parallel to the experimental data collection.

4.2. Machine Learning Methodology

Initially, a larger variety of machine learning methods were considered for this investigation, but due to the simplicity and emerging literature on PINNs, they became the logical candidate.

Employing a ground-up approach, the PINN model was initially trained in a 2D domain, with the goal of an eventual expansion to 3D. Many of the inherent difficulties in training PINNs appear in both 2D and 3D domains, so a 2D domain was a logical choice to begin with to gain familiarity with the network and easily debug issues. This chapter will document the PINN development across both 2D and 3D domains, providing motivation on certain design choices and training challenges. Finally, architectural considerations for the neural network design will also be discussed.

4.2.1. 2D PINN Model

The choice for a 2D model was motivated both by the desire to begin with a simple model, as well as the need to verify the implementation, particularly for the turbulence modeling. For this reason, it could most easily be compared to open source CFD simulations[4] that had been run over the Boeing bump geometry, which provided the initial/boundary conditions for the problem. The following subsection will discuss the boundary conditions, while subsequent sections will discuss the loss function and turbulence model that comprise the rest of the model. For the PINN architecture, chapter 3 can be referenced, and Figure A.1 can be referenced for an overarching description of the processes and their flow.

Boundary Conditions

To simulate the 2D flow in a channel over the Boeing bump, the domain in Figure 4.4 was established, with an inlet on the left, and outlet on the right. The inlet boundary specifies the inlet velocity ($M = 0.2$), and turbulence variable (ν), whose value was fixed corresponding to the inlet Reynolds number of $Re = 3 \cdot 10^6$, determined from the relation $Re_L = \frac{vL}{\nu}$. Since velocity was fixed at the inlet, static pressure was fixed only at the outlet to the atmospheric value $P_s = 101325$ [Pa].

The top and bottom boundary conditions represent the floor and ceiling through a no-slip condition, respectively, where horizontal velocity, u, and vertical velocity, v, are fixed to 0.

[3] https://turbmodels.larc.nasa.gov/Other_exp_Data/speedbump_sep_exp.html

[4] https://turbmodels.larc.nasa.gov/index.html

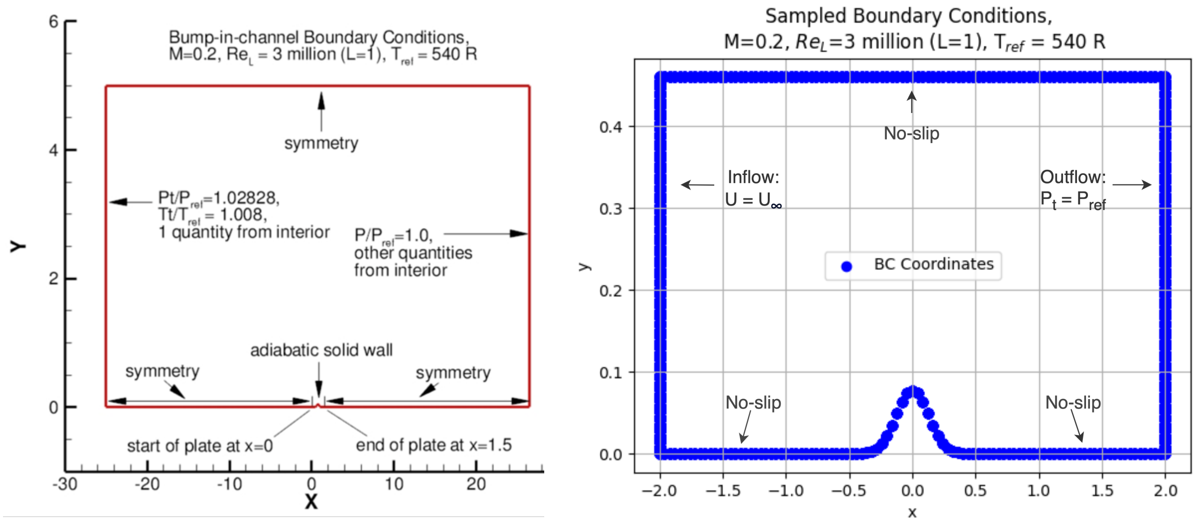


Figure 4.4: Boundary conditions as used in the OpenFoam simulations[5](left) and replicated domain for the PINN (right)

Notably, there are certain differences between the boundary conditions imposed in Figure 4.4, specifically regarding the size of the domain. Due to the relative expensiveness of training a PINN compared to a RANS simulation, it was necessary to conserve the computational effort by limiting the domain size. Furthermore, the domain enforced in the current study has been produced to match the geometry of the wind tunnel experiments (e.g. cross-section dimensions), owing to the need for experimental data integration. This extends also to the exact curvature function describing the surface of the bump, which differs slightly between numerical and experimental campaigns. For this reason, some inherent differences can be expected between 2D results produced by the PINN and by the CFD solver, further discussed in section 5.1.

Physical Data Integration

The primary selling point of PINNs (as well as Physics-Informed Machine Learning methods in general), remain their aptitude for physical data acquisition. While possible in conventional CFD solvers, it cannot be disputed that the PINN method involves a drastically simpler approach to integrating experimental data.

In similar fashion to the integration of boundary conditions (through generation of collocation points at domain locations where conditions are known), the data can too be integrated into the PINN domain. This is shown in Figure 4.5, where multiple planes of streamwise (U-component) velocity are integrated over the bump boundary, in conjunction with boundary conditions (blue).

[5] `https://turbmodels.larc.nasa.gov/index.html`

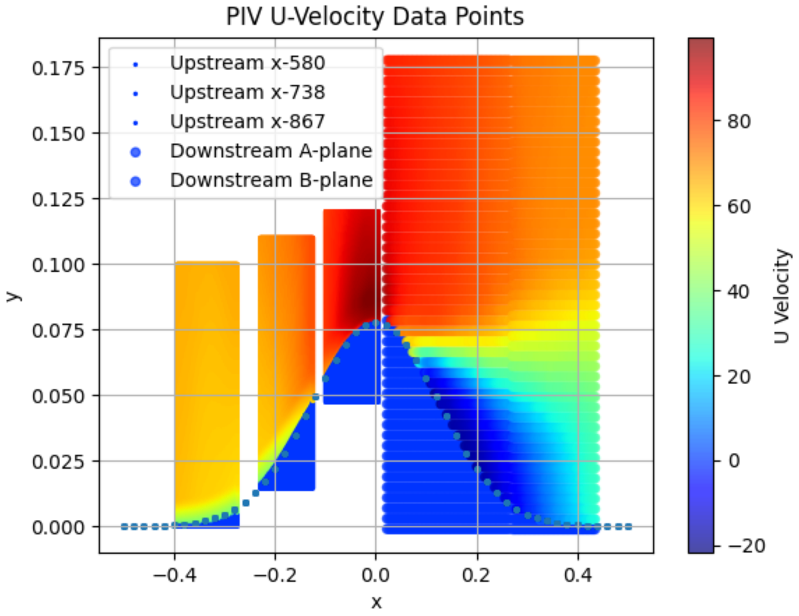


Figure 4.5: 2D, 2-component PIV streamwise velocity data over bump centerline, capturing incoming flow (from left to right: x-580, x-738, x-867), and downstream flow data (A & B plane). Nomenclature is consistent with source of data in [10, 13, 14]

As shown in the figure, data acquisition remains as straightforward as the prescription of boundary conditions, since, their introduction to, and evaluation within the neural network remain the same. Use of a uniform coordinate system centered at the bump apex ensures alignment of data with no-slip boundary condition imposed over bump surface. The nomenclature of the various PIV planes is consistent with that of the original experiments with which they are produced [10, 13, 14]; with numbers for upstream data indicating distance in [mm] from front splitter plate, located at $x/L = -1$, where $L = 0.9144$ [m]. Planes A and B begin at $x = 0.022$ and $x = 0.272$ [m], respectively.

Physics-Informed Loss Formulation

The third and final component, which distinguishes the PINN method from physics-unaware methods, is the introduction of collocation points upon which PDE residuals are to be evaluated. As supported by recent works [42], these can be best distributed in a pseudo-random manner, though the spacings between these points can be fit to ”mesh” a certain geometry more appropriately. Figure 4.6 displays an example of 2601 collocation points pseudo-randomly distributed over a 4 x 0.5 [m] domain, mirroring that of the experimental test section. In reality, 10201 collocation points were used for the 2D simulation, however this image serves to display the relative easy of generating a grid of points when compared to the meshing requirements of CFD methods. To provide sufficient collocation point density over the bump and lower surface, an exponential spacing is placed in the y-dimension. The density of these collocation points can best be compared to the mesh that a typical CFD solver would require, and similarly a sufficiently refined representation is required for optimal results. Particularly due to the efficient computation of derivatives using the autograd method, excessively sparse points can further hinder the accurate computation of partial derivatives, introducing errors into the optimization.

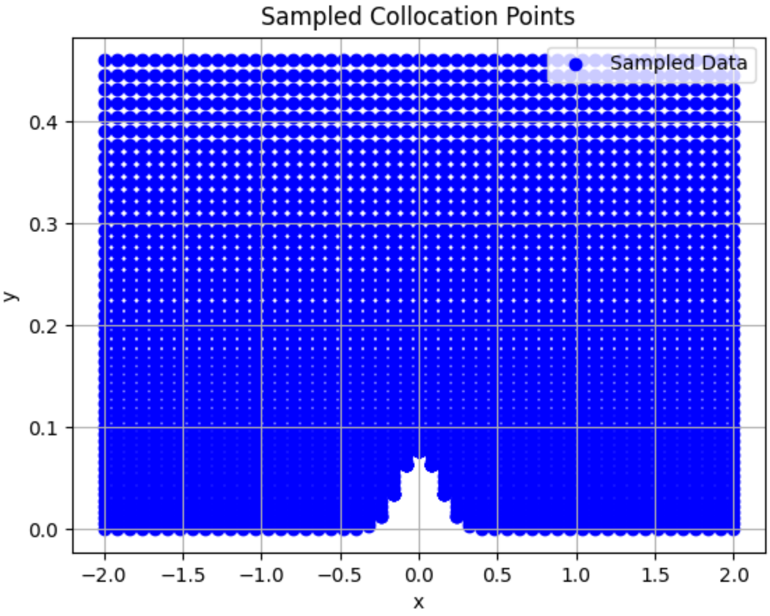


Figure 4.6: Collocation points distributed across domain, with exponential spacing in y-dimension indicating coordinates where PDE residuals are evaluated and returned as loss to PINN.

The governing equations of relevance to this investigation are the 2D Reynolds-Averaged Navier Stokes equations, depicted Equation 4.2. In this sense, the governing equations allow the PINN to intelligently interpolate over the domain at points where boundary conditions or data are not available to provide information.

$$\begin{aligned} \nabla \cdot \mathbf{U} &= \epsilon_1 \\ \rho(\mathbf{U} \cdot \nabla \mathbf{U}) + \nabla p - \mu \nabla^2 \mathbf{U} - \nabla \cdot (-\rho \langle \overline{\mathbf{u'u'}} \rangle) &= \epsilon_2 \end{aligned} \tag{4.2}$$

While the object of this investigation is not to replace contemporary CFD methodologies, it is most closely comparable to a RANS methodology, which was the justification for this choice of governing equations. Given the unclosed terms within the RANS equations, it was therefore necessary to include a turbulence model, for which the Spalart-Allmaras model was chosen.

Turbulence Modeling & Verification

The Spalart-Allmaras (one-equation) model is governed by the transport equation for the eddy viscosity-like quantity, for which it was chosen, $\tilde{\nu}$:

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right] \left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left\{(\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right\} + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right] \tag{4.3}$$

where:

$\tilde{\nu}$ is the modeled eddy viscosity-like quantity, t is time, x_j and u_j represent spatial coordinates and velocity components, respectively, d is the distance to the nearest wall, ν is the kinematic viscosity of the fluid, C_{b1}, C_{b2}, C_{w1}, σ are model constants, and $\tilde{S}$ is the modified strain rate, given by:

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \tag{4.4}$$

for which S is the strain rate magnitude, given by:

$$S = \sqrt{2 S_{ij} S_{ij}}, \tag{4.5}$$

and S_{ij} is the strain rate tensor, given by:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{4.6}$$

The damping functions f_w and f_{v2} are defined as:

$$f_w = g\left[\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right]^{1/6}, \quad g = r + c_{w2}(r^6 - r), \quad r = \frac{\tilde{\nu}}{S\kappa^2 d^2}, \tag{4.7}$$

$$f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \quad \chi = \frac{\tilde{\nu}}{\nu}. \tag{4.8}$$

The model constants are:

$$\begin{aligned} &C_b1 = 0.1355, && C_b2 = 0.622, && \kappa = 0.41, \\ &C_w1 = \frac{C_b1}{\kappa^2} + (1 + C_b2), && \sigma = 2/3, && c_{w2} = 0.3, \\ &c_{w3} = 2, && f_{v1} = 1.4. && \end{aligned}$$

While various implementations and variations of the Spalart-Allmaras model exist, the NASA turbulence page was used as a baseline and source of information. The purpose of the 2D model, remained only to verify the turbulence model selected, and therefore consistency with the turbulence model used in the OpenFoam simulations was the primary criteria.

Regarding its inclusion in the Physics-Informed Neural Network, its contribution came in the form of another loss variable:

$$u_j\frac{\partial\tilde{\nu}}{\partial x_j} = c_{b1}(1-f_{t2})\tilde{S}\tilde{\nu} - \left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left\{(\nu+\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\right\} + c_{b2}\frac{\partial\tilde{\nu}}{\partial x_i}\frac{\partial\tilde{\nu}}{\partial x_i}\right] = \epsilon_3 \tag{4.9}$$

The outcome of the 2D verification campaign is discussed in section 5.1. The creation and evaluation of a successful 2D model remained crucial to the eventual expansion to a 3D model, as will be seen in the subsequent chapters. To recall the way in which these three components are summed into a unified loss for the PINN to train upon, chapter 3 provides the architectural diagram of the PINN designed for this exercise and the relevant equations that comprise the loss function. The 2D PINN architecture differs to the 3D architecture only in the absence of the 3rd dimension, z, and its velocity variable w. As such, the 2D PINN case accepts two (x, y) coordinates as input and only outputs $(u, v, p, \tilde{\nu})$.

4.2.2. 3D PINN Model

Building upon the 2D PINN model, only a minimal number of changes were required in achieving a 3D PINN model, which are outlined in this section. The simplest modification came in the PINN architecture, which required only an additional input variable (z-axis coordinate), and an additional output variable (w-component velocity. The most involved modifications came in the form of boundary condition and data integration, as well as re-fitting various hyperparameters. These will be discussed in appropriate detail in the following sections.

Modifications to Domain & Boundary Conditions

To visualize the 3D domain, the ParaView software was used and an STL file of the Boeing bump geometry was imported to provide context to the flow. The geometry overlaid with collocation points to be evaluated by the PINN is shown in Figure 4.7. Similar to the previous 2D case, the collocation points follow an exponential distribution giving increased density at lower y-coordinates. This is done to provide sufficient representation in the boundary layer of interest over the bump surface, owing to the need for accurate evaluation of partial derivatives. As such, a compromise was required, allowing a sufficiently dense distribution to be spread over the lower surface while not sacrificing the required density to accurately represent the upper boundary layer. The distribution of collocation points is also increased over, and slightly after the bump apex, to adequately represent the regions of interest.

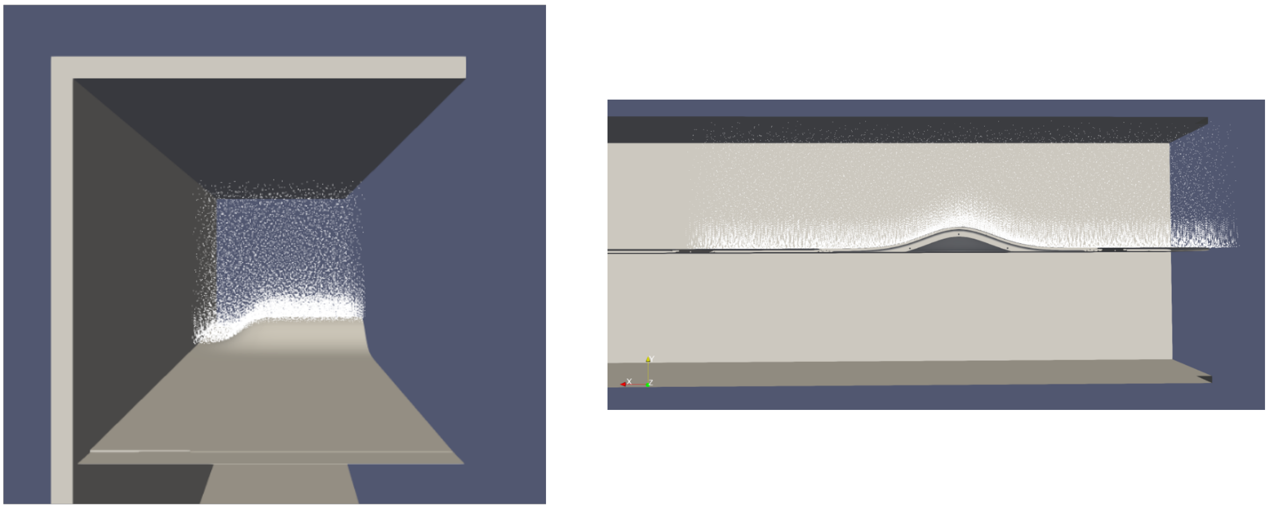


Figure 4.7: Front, downstream view of collocation points through bump channel (left) and side view (right), with exponential sparsity distribution.

On the Boundary Layer Resolution (y+ criterion) and Convergence Study

In a CFD study, it would be necessary to perform a convergence study, varying the mesh count over simulations until the error no longer diminishes, verifying convergence of the method. A similar method must be implemented in training PINNs, where the collocation point count is increased until no further benefit is witnessed. While this study was limited in compute power, the need to investigate convergence was respected, and the results are described in section 5.2. Similarly, owing to the original interest to model the viscous sublayer of the boundary layer, the y+ criterion was determined such that it could be fulfilled. While this study did not contain the required computational power to effectively resolve the boundary layer (discussed also in section 5.2 and 6), the consequences are noted.

Due to the symmetry of the domain and the need for efficient computation, the domain was cut in half across the axis of symmetry. To correctly enforce the symmetry at the center-line, a symmetry boundary condition was imposed, as shown in Figure 4.8. The same figure also displays the no-slip boundary condition ($u, v, w = 0$) which was implemented over the surfaces of the bump, sidewalls and ceiling.

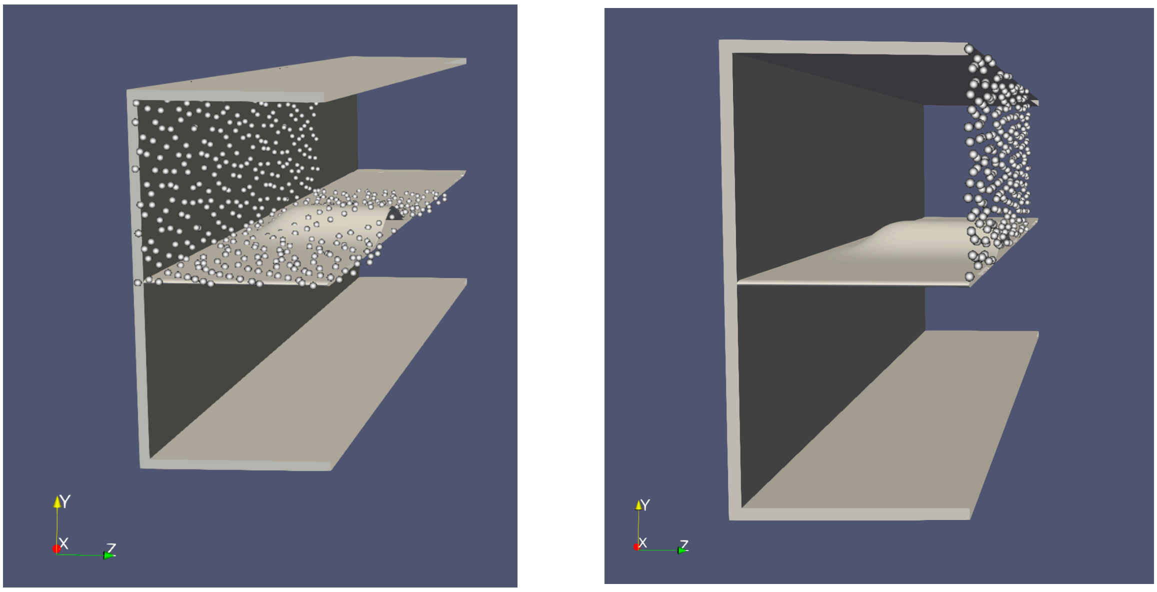


Figure 4.8: No-slip boundary condition enforcing zero velocity ($u, v, w = 0$) at surfaces (left) and symmetry boundary condition enforcing 2D flow ($w = 0$) at $z = 0$ (right).

Not shown but of equal importance to the simulation of a correct solution are the inlet and outlet boundary conditions. These remain identical in magnitude to the 2D case, with the inlet boundary condition prescribing the inlet velocity (parallel flow, $u = u_\infty; v, w = 0$), while the outlet fixes the static pressure ($p = p_0$).

Correct implementation of the boundary conditions is crucial to a successful simulation, not unlike CFD. Having verified the proper implementation in 2D, the expansion to a 3D domain only consisted of the inclusion of one additional dimension, and could be easily implemented.

Modifications to Data Integration

Similar to the inclusion of 3D boundary conditions, additional data could be included for a 3D domain. Figure 4.9 displays the inclusion of all PIV planes for which streamwise velocity measurements were taken. In the downstream segment of the bump, a total of 4 planes of 2D2C PIV measurements were integrated. Upstream, 3 additional planes of 2D2C SPIV measurements were integrated, however along the same (center-line) axis.

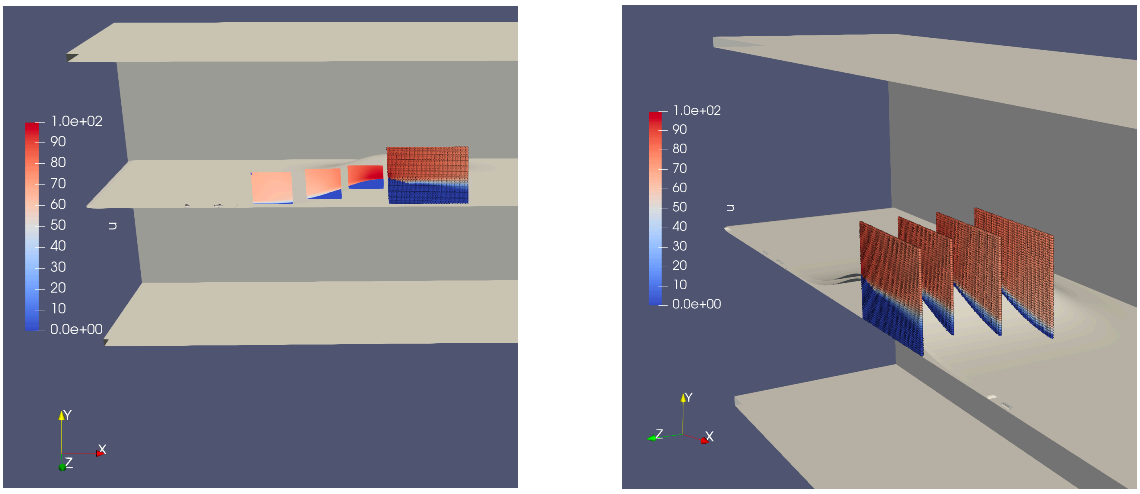


Figure 4.9: Streamwise center-line velocity data consisting of 2D2C PIV downstream of bump and 2D2C SPIV upstream data of bump (left), shown alongside multiple spanwise planes of downstream 2D2C PIV.

In similar fashion, the vertical component of the velocity (v) can also be shown, with which the PINN was also trained, shown in Figure 4.10. Important to note is that the PINN receives no information about the spanwise velocity (out-of-plane component), which it is to interpret from 2D data exclusively.

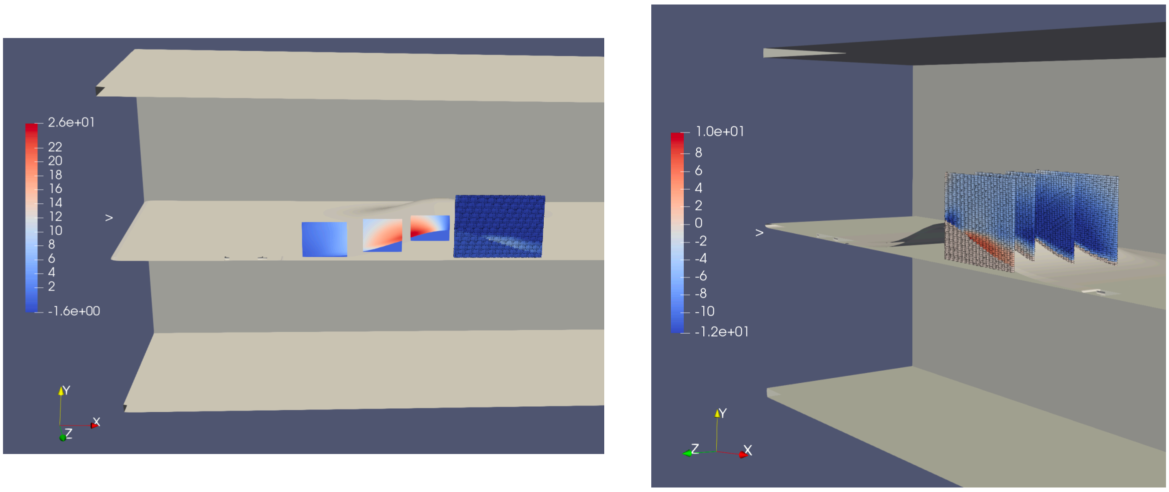


Figure 4.10: Vertical center-line velocity data consisting of 2D2C PIV downstream of bump and 2D2C SPIV upstream data of bump (left), shown alongside multiple spanwise planes of downstream 2D2C PIV.

Finally, the third data component with which the PINN can be trained is the static pressure collected over the surface of the bump, shown in Figure 4.11, where a clear pressure minimum can be seen over the apex of the bump. Similarly, in the upstream region of the bump, an increase in pressure can be observed, owing to the incoming velocity entering the throat of the channel.

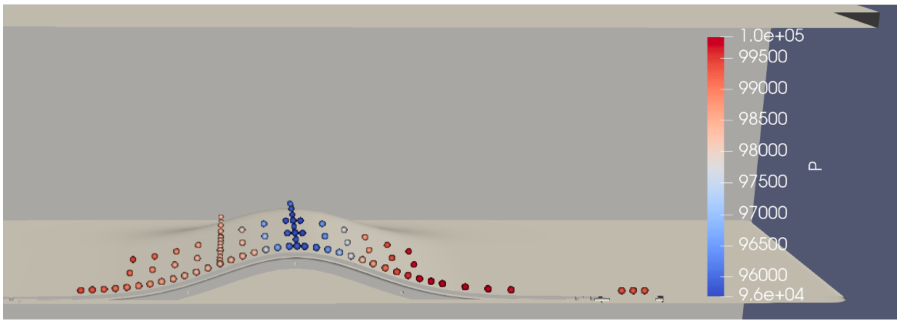


Figure 4.11: Static pressure measured using taps over the Boeing bump surface recreated from [14].

Indeed, the approach to data acquisition did not differ much from the 2D case, and exemplifies the PINN use case for its straightforward ability to assimilate a variety of data types. Ultimately, it will be of interest to discuss with how little information the PINN can operate, and which data types are the most crucial to a successful simulation (see section 5.3).

Modifications to Governing Physical Equations

The expansion from 2D Reynolds Averaged Navier Stokes equations to a 3D PDE consists only of adding the third component equation. As such, this was the simplest aspect of the PINN to expand to a 3D solution. Due to the existence of a third dimension, it was then possible to extend this investigation and predict the direction of the surface shear stresses of the flow. The surface shear (traction vectors) are related to the first spatial derivatives of the flow velocity in every axis, namely through the following definition:

$$T_i = -p \cdot n_i + \mu \Sigma_j \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_j \tag{4.10}$$

The velocity gradients are already available from the PINN method, due to their calculation for the RANS equations used in Equation 4.2. In expanded form, these stress tensor components take the following notation:

$$T_x = (-p + 2\mu\frac{\partial u}{\partial x})n_x + \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})n_y + \mu(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x})n_z \tag{4.11}$$

$$T_y = (-p + 2\mu\frac{\partial v}{\partial y})n_y + \mu(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})n_x + \mu(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y})n_z \tag{4.12}$$

$$T_z = (-p + 2\mu\frac{\partial w}{\partial z})n_z + \mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z})n_x + \mu(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y})n_y \tag{4.13}$$

In fulfillment of the ultimate objective in this investigation, these equations allow for the computation of the shear topology over the surface of the bump, only available in a 3D campaign.

Since they provide only a qualitative insight into the flow behavior, their magnitudes are normalized, and combined into a normalized vector to provide a general flow direction with no extrapolation on the magnitude of the surface shear stress. The need for normal vectors (n) at every coordinate was fulfilled from previous data obtained during the collaboration between Caltech and Boeing [41].

4.2.3. Preliminary Analysis and Architectural Considerations

This section will discuss the architectural considerations that proved to be important in the search for a global solution. While the base PINN architecture was proposed to be simple, the desire to apply such networks to more advanced problems accompanies the need for more advanced network architectures. Stemming in equal parts from a computational cost and accuracy perspective, certain alterations to the architecture were explored in attempts to reduce the RAM limitations, improve training time and improve overall convergence. As will be discussed in the remainder of this study, the limitations imposed by the RAM overhead remain a significant hurdle in the PINN use-case, ultimately limiting the accuracy of results and convergence, as well as hindering the training time.

Model Parallelization over Multiple GPUs

The most obvious solution to maximize the computational resources available for this investigation was to use 2 GPUs instead of a single GPU. Available for this study were two Nvidia RTX A5000 GPUs, upon which the 3D model was run when Google Colab's A100 GPUs provided insufficient RAM for training. A single Nvidia RTX A5000 GPU offers 24GB of RAM, with two GPUs offering RAM just short of 50GB.

For this reason, model parallelization was prompty investigated, and implemented. When discussing parallelization, it is important to differentiate between model parallelism and data parallelism:

- Model parallelism involves modifying the neural network architecture to manually (or in an automated fashion) distribute the layers and nodes over two or more GPUs.
- Data parallelism involves distributing the training data, allowing the network to process multiple chunks of information simultaneously.

Model parallelism remains more straightforward and applicable for PINN problems due to the need to represent the full domain over every epoch. Splitting the data, as with data parallelism, would require enforcing boundary conditions for each tranche of data which has been distributed. Model parallelism, on the other hand, continues to process the entire dataset, with different devices responsible for hosting different parts of the neural network. This remains more straightforward to implement, since the training data can remain on a single device, provided the input and output layers of the neural network reside on the same device. This was the approach that was implemented to access the RAM of both GPUs, where the training data was introduced to the first layer of the PINN, which resided on GPU 0, and then passed the output of the forward pass to the subsequent layers, which resided on GPU 1. The entire ”body” of the PINN was then trained on GPU 1, until its final output was passed to the final two layers, which again resided on GPU 0. Figure 4.12 displays a diagram of the distributed network over 2 GPUs.

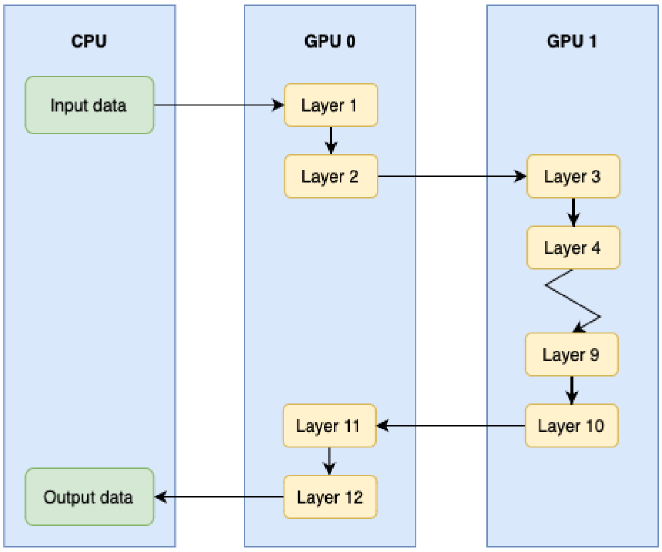


Figure 4.12: Sample diagram for network PINN with 12 layer architecture distributed over 2 Nvidia RTX A5000 GPUs.

Noteworthy is that while this significantly increases the available RAM, training time is hindered due to the need for both GPUs to communicate the information between each other. While the increased time as a result of this implementation over 2 GPUs was negligible, it remains important to note if considering a larger GPU cluster. Accordingly, this parallelization offered double the RAM allowance compared to a single GPU, approximately allowing for the choice of increasing the network size by factor 2, or increasing the number of collocation (evaluation) points by factor 2. Following subsections will discuss the trade-off between these two avenues, and problem-specific consequences on training time, convergence and accuracy.

Trade-off on Collocation Points and PINN Size.

Following the limitations imposed by the RAM availability of the two RTX A5000 GPUs that were used for training, a trade-off arose between the number of collocation points and the size of the PINN. These parameters come at odds with another, with the former dictating the accuracy of the physical loss term (whose terms depend strongly on the spatial derivatives of the velocity and pressure variables in question). Therefore, an insufficient coverage over the entire domain will limit the model's accuracy in data-deficient locations. Conversely, the size of the neural network directly affects the model's ability to attain sufficient expressivity as a universal approximator; an insufficiently large model will be incapable of expressing the complexities of a turbulent 3D flow. Finally, on either end, significant computational cost is to be expected, with additional neurons/layers as well as collocation points incurring significantly more resources and training time.

For this reason, it was imperative to track the effects of these parameters on the overall error of the solution. This encourages the use of the most optimal parameters, which in PINN studies remains highly problem-dependent.

The first parameter varied is that of the node count (the width of the Multi-Layer Perceptron network). Due to the industry practice of increasing this parameter in powers of 2 [43], it came at significant cost of compute power. For this reason, an increase in node count by factor 2 accompanied a decrease in collocation point count by the same factor. The effects are summarized in Figure 4.13.

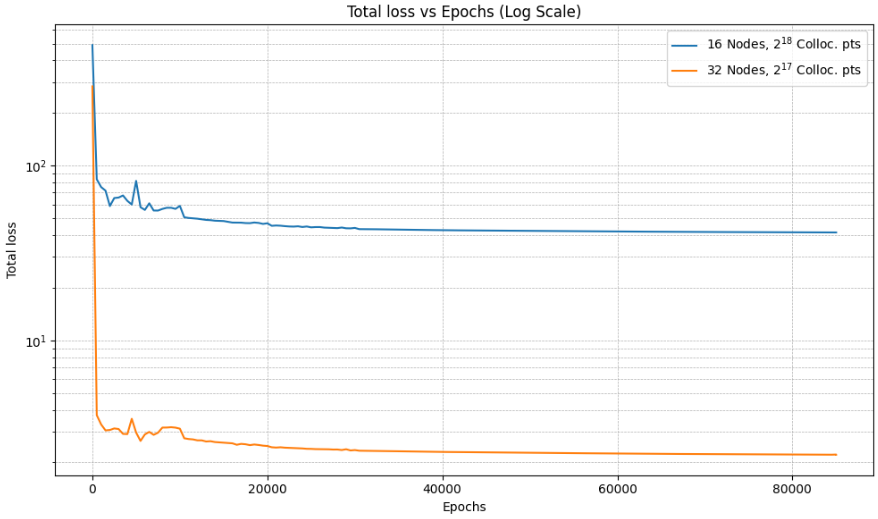


Figure 4.13: Comparison of training performance as PINN hidden layer count is varied.

The figure above seems to unambiguously reinforce the relative importance of architecture node count over collocation points, and the benefits of a larger network cannot be denied. Nevertheless, the collocation points density directly affects the accuracy of velocity/pressure gradient computation. Furthermore, it remains possible that the lower boundary of expressiveness was limited through the architecture, beyond which further improvements could only be yielded through increased collocation points. Nevertheless, at this stage of computational effort, it remains undeniable that increasing the PINN size unlocked significant gains in decreased loss.

Similarly, one can assess the effects of increasing the number of NN layers, while maintaining a constant node and collocation count. The results of such a study are shown in Figure 4.14, where the training losses are shown for various layer counts for the best performing combination of node and collocation count outlined previously. Notable is that increasing the layer count does not affect computational cost as adversely, since improved expressivity are attainable through integer increments of hidden layers. As such, it was only necessary to increase by one to two layers at a time, and diminishing returns were observed.

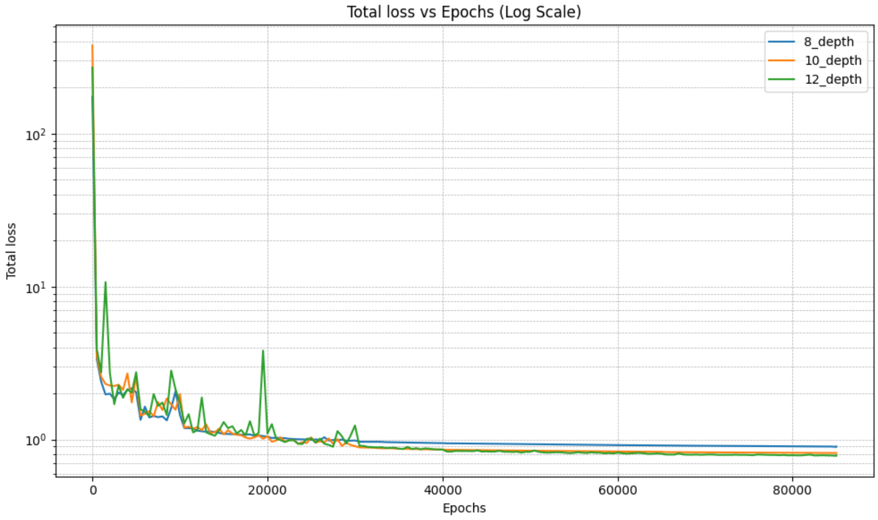


Figure 4.14: Comparison of training performance as PINN hidden layer count is varied.

Important to note is that despite the apparent improvement in total error reduction by virtue of the increased number of hidden layers, this is not indicative of improved performance when compared to the validation data. When plotting the data produced by the PINN with 12 layers, it was discovered that it trained towards a trivial solution, where the z-axis velocity, w, was predicted to be 0 throughout the entire domain. While this is likely to be a manifestation of the random initialization of collocation points, it remains important to note that preliminary analysis only provides a limited view to the PINN's final performance.

Table 4.1: Overview of training times per architectural configuration.

	NN Depth	NN Nodes	Fourier Frequency	Batch Count	Collocation Points
Value	9	32	2	25	2^{17}
Range	8-12	32-256	1-10	1-50	$2^{15} - 2^{18}$

Ultimately, a PINN architecture with 8 hidden layers and 32 nodes was decided due to the diminished returns observed when increasing the architecture depth. Instead, offering the computational resources towards an increased collocation point count offered far more return on investment, and was selected for the remainder of this investigation. Nevertheless, the possibility remains that due to the need to trade-off collocation points for network size, that an insufficient number of collocation points were provided to fully exploit the expressiveness of the larger networks. For this reason, the subsequent section discusses methods employed to increase the domain coverage with use of fewer collocation points.

Mini-Batching and Batch Sizes

A common method in machine learning is to deploy the training data in smaller batches to reduce the overhead during a single training epoch/cycle. Since PINN applications typically require a full domain to be introduced to the network, ”mini-batching” limits the minimum collocation points that can be supplied for successful training. Nevertheless, to provide a larger coverage of the domain than RAM-limits allow, conventional methods cycle the coordinates of these collocation points in every epoch to allow the PINN to train over more spatial locations. In situations where the RAM-limits allow for more collocation points than necessary, this methodology can also be used to compromise the number of collocation points used in a single training epoch, while still maintaining domain coverage.

An additional consequence of the shuffling of these collocation points per epoch (or several epochs), is the natural introduction of noise during the training process. This (as with many hyperparameters) provides another optimization problem, where excess noise can prohibit training, but an appropriate degree can improve training and encourage the network to explore different paths of optimization (and reduce the risk of over-fitting).

Due to the increasing number of hyperparameters in question, a quick study was performed on the batch size to explore its optimal upper and lower limits.

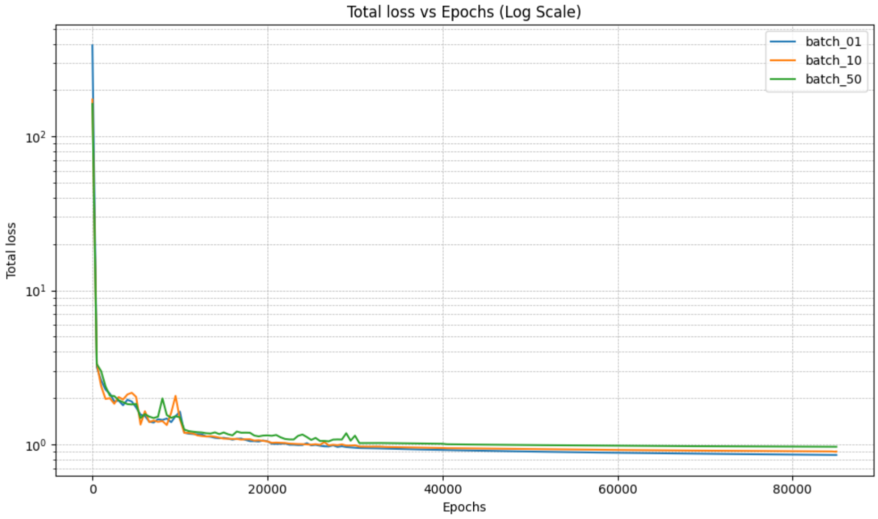


Figure 4.15: Comparison of training performance as quantity of sub-batches (collocation points) are varied.

Optimally, a robust multi-variable analysis would be performed due to the co-dependence between variables such as batch size, network size, collocation points, etc. In the interest of time and computational resources, this study will only provide a brief study on these variables insofar as it related to a better validation error with regards to the test data. chapter 6 will suggest various recommendations to improve the search for an optimal network size and its related parameters.

Choice of Loss Function

The structure of PINNs with their evaluation of the PDE residuals allows for active loss tracking throughout the entire training process. In other words, PINN's "learning" can be observed through time, providing insight into which areas of the domain remain difficult to resolve accurately, and whether a local or global minimum is being targeted. At each epoch, the neural network estimates the loss landscape through backpropagation of the linear sum of all loss terms. As a result, the exact method of computation of these loss parameters are highly influential on the loss landscape, and conducive to whether the PINN may approach a local or global minimum. While the literature recommends the use of a Mean Squared Error (MSE) function to evaluate the loss terms, this investigation discovered that a Root Mean Squared Error (RMSE) function is more conducive to a balance landscape, and the approach to a global solution.

To illustrate the importance of this, Figure 4.16 displays two identical plots of the z-momentum equation loss over all collocation points during late-stage training. The left figure, displays what appears to be a nearly uniform loss landscape, despite the scale showing a large error (existing in the farfield of the domain). When clipping the error and removing the outlier loss, the right plot is obtained displaying the unresolved z-momentum equation. The strongly defined plane on the down-sloping region of the bump shows the error over one of the validation planes.

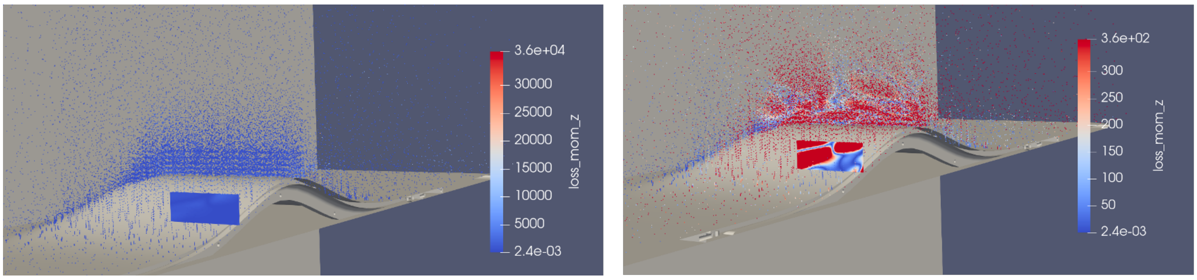


Figure 4.16: Comparison of identical loss landscape with (left) outlier losses included and (right) outlier losses removed.

In relation to the loss functions described previously, it was discovered that applying a RMSE function as opposed to a MSE function, returns a more homogenous loss landscape to the optimizer, and avoids the targeting of these outlier losses. This realization was crucial in the approach of a global solution, due to the relative unimportance of errors in the farfield, compared to the post-bump flow region. Other candidates for the loss function often implemented are the Huber loss function or the log function; empirically, the RMSE function worked best in this case, though further investigation is necessary to determine whether this is a problem-specific observation.

Choice of Optimizer

The choice of optimizer is equally crucial in the search for a global solution to PINN problems. Common in modern PINN literature is to begin the training with an Adam optimizer, and then pivot to L-BFGS as an optimizer after several thousand epochs. This is well described by authors in [44], where the relative performance of each optimizer, as well as their combination is assessed. Ultimately, it remains important to note that the performance of these optimizers is (as with many PINN problems), case-dependent. In this investigation, the pivot to an L-BFGS optimizer showed little promise in further reducing the loss as literature might suggest.

The theory-based justification for this change of optimizer argues that L-BFGS being a quasi-Newtonian optimizer targets local minima. This, coupled with its computational expensiveness as a result of its repeated approximation of the Hessian at every iteration, make it a better candidate for late-stage training when the global solution is approached, and fewer iterations are necessary. Adam, as a gradient-descent optimizer instead avoids such saddle points that quasi-Newtonian optimizers struggle with [44].

Ultimately, as it relates to this investigation, the Adam optimizer was used in the entirety of the training. Coupled with the fact that no noticeable benefit was observed (at the cost of significantly greater computational effort), the PyTorch implementation of L-BFGS requires the use of a single GPU, which was no longer feasible when the model was split over 2 GPUs. The lack of benefits observed from the L-BFGS optimizer in this investigation did not warrant the solutions to either, write an independent quasi-Newtonian optimizer, or employ a different machine learning library. Nevertheless, one must conclude that the lack of success of the L-BFGS optimizer may stem from an ill-formalized optimization where the Adam optimizer never escaped a saddle point, explaining the lack of benefits of the quasi-Newtonian algorithm.

Random Fourier Features

Implementation of a Fourier transform prior to the first hidden layer of the PINN has become common practice and crucial in this study. The simple justification for this modification of the standard MLP architecture of the PINN is an ability for the PINN to approximate non-linear solutions that are linearized in the Fourier space. A reverse Fourier transform is not necessary, since the MLP learns to implicitly perform the reverse Fourier transform, though adaptations to this implementation exist where a manual reverse transform is performed, or alternatively, consisting of multiple transformations.

The effects of a well-tuned Fourier transform in the PINN is best visually observed in the 2D model, through which is was initially calibrated. section 5.1 discusses the differing results based on this parameter and its importance for a convergent solution.

Learning Rate and Decay Rate

The benefits of an exponentially decaying learning rate scheme are well documented in PINN literature [44], and much effort was therefore not dedicated to the optimization of this parameter. A slightly larger than usual initial learning rate of $lr = 5 \cdot 10^{-3}$ was employed to accelerate initial training, since lower learning rates empirically showed the same outcome despite a higher time burden. Additionally, the use of a higher learning rate is conducive to the optimizer's ability to approach the global minimum of the solution as opposed to trending towards a local minimum. The specific decay algorithm(s) used in this investigation are detailed in Appendix B.

Overview of Network Parameters

In summation of the architectural decision outlined in this chapter, it is relevant to provide an overview of those most relevant to achieving convergence towards the global minimum solution. The findings are summarized in Table 4.2, where the hyperparameters most relevant to the proper convergence are displayed.

Table 4.2: Overview of relevant hyperparameters relevant for tuning

Parameter	NN Depth	NN Nodes	Fourier Frequency	Batch Count	Collocation Points
Value	9	32	2	25	2^{17}
Range	8-12	32-256	1-10	1-50	$2^{15} - 2^{18}$

The parameters above can be likened to tuning knobs in a control system, where the PINN prediction is highly sensitive to its arguments. For each parameter, the optimal value for this investigation is displayed, as well as the appropriate range supported by findings in literature. It is important to note here that the size of the neural network (nodes, depth), as well as the number of collocation points are best maximized when possible. Only the computational cost limits this variable from being maximized, and no hindrance to performance is exhibited at excessively large values. Conversely, excessively large values for the Fourier Frequency and batch count can severely affect the network's ability to convergence to the correct minimum, or converge at all.

5 Results & Analysis

This chapter will document the PINN prediction results after full training cycles and optimized hyperparameter selection. The goal of this investigation remains to predict a fully developed 3D velocity and pressure field from 2D experimental data. It is necessary to note that the results presented in this chapter are obtained by training on publicly available data obtained from the NASA turbulence modeling resource[1]. This was decided from the need to maintain consistency between training and testing (validation) data, which for the $M = 0.2$ case (demonstrating greatest separation of flow), was not able to be produced in the Lucas Wind Tunnel at Caltech. The results will be documented both for the 2D and 3D cases studied, and an ablation study will be included for the latter, allowing conclusions to be drawn regarding the lower limit of data required for a successful reconstruction. Finally, a subsection will be dedicated to the prediction of surface shear stresses and the coefficient of friction over the geometry.

5.1. 2D Vector Field Reconstruction

To ensure a functional data acquisition and training pipeline, the model was first examined over a 2D domain, as discussed in section 4.2. The prediction of an accurately developed boundary layer, and correct representation of flow reversal remained difficult, despite provision of PIV-data. This allowed exposure to the behavior of the PINN in an environment where errors were easy to diagnose. Furthermore, errors in PINN prediction recognizable by the eye test, would go on to present themselves in similar manners in 3D results. An example of this, is shown in Figure 5.1, where two simulation predictions of streamwise velocity are shown, differing only in 1 hyperparameter setting. Notably, this setting was the frequency of the Fourier transform applied to the PINN, shown as the first layer of the MLP architecture in Figure 3.2. The intuitive explanation of the Fourier transform comprising the first layer of the PINN architecture, is the capacity for the PINN to express increasingly non-linear solutions (controlled by the weights of the Fourier transform). In essence, the PINN is able to more easily approximate non-linear basis functions which are linearized in the frequency domain, allowing larger gradients in the spatial domain to be represented. As such, the frequency of the transform, as a proxy for the non-linearity of the solution can allow higher frequency solutions to be modeled, as required for the correct representation of boundary layer thickness, for instance.

[1] https://turbmodels.larc.nasa.gov/Other_exp_Data/speedbump_sep_exp.html

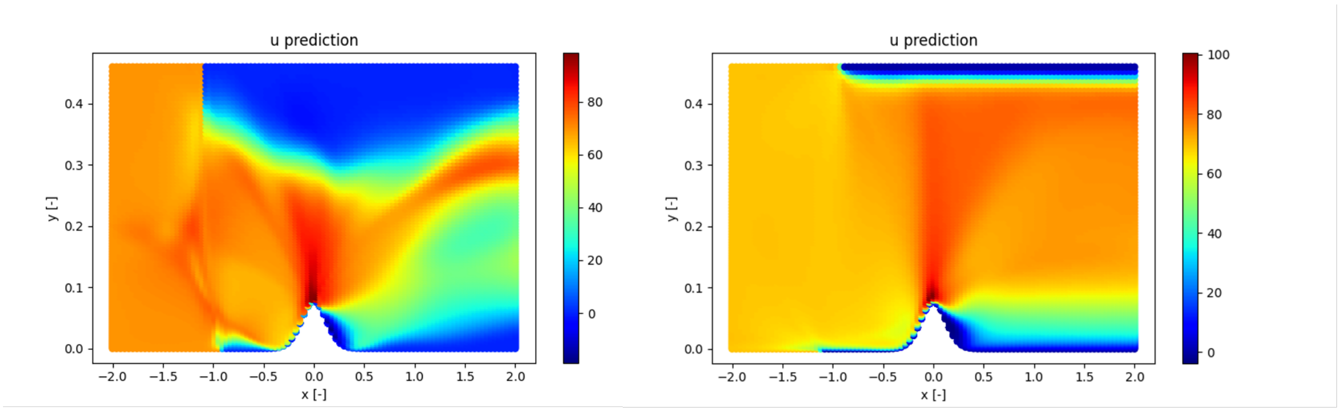


Figure 5.1: Comparison of 2D PINN streamwise velocity prediction without (left) and with (right) effects of Fourier transform.

The larger boundary layer on the upper ceiling can be attributed to the overlapping datapoints, as well as increased sparsity of collocation points with increasing y-axis.

The greatest conclusions obtained at the 2D stage of training was the following relation which strongly dictated the accuracy of the prediction:

$$\epsilon \propto f(\frac{\mathcal{L}_{data}}{\mathcal{L}_{physics}}, \mathcal{F}(\omega)) \tag{5.1}$$

where, $\mathcal{L}_{data}$ and $\mathcal{L}_{physics}$ dictate the weights of the data and physics losses, respectively, while $\mathcal{F}$ represents the Fourier transform, with input frequency, ω. This echoes a similar conclusion discussed previously, following the works of [34], where authors conclude that the natural limitations of data-driven methods such as PINNs remain the availability of training data itself. As such, of the relevant parameters N_{cell}, N_{eqns}, and N_{data}, the latter remains the most influential in the optimization towards a globally low error solution.

Extending from the previous plots, the normalized eddy viscosity over the apex of the bump is shown in Figure 5.2. Important at this stage of verification, is that the trend was correctly matched, as well as order of magnitude. Note the difference in apex height (0.075 vs 0.05), stemming from different bump geometries used in the OpenFOAM campaign vs PINN study. The PINN study could not modify its geometry to that of the OpenFOAM study due to the need to match the training data (obtained using experimental geometry). For this reason (and others elaborated on subsequently) an exact recreation of the turbulence variable in both form and magnitude was not deemed necessary before advancing to a 3D model.

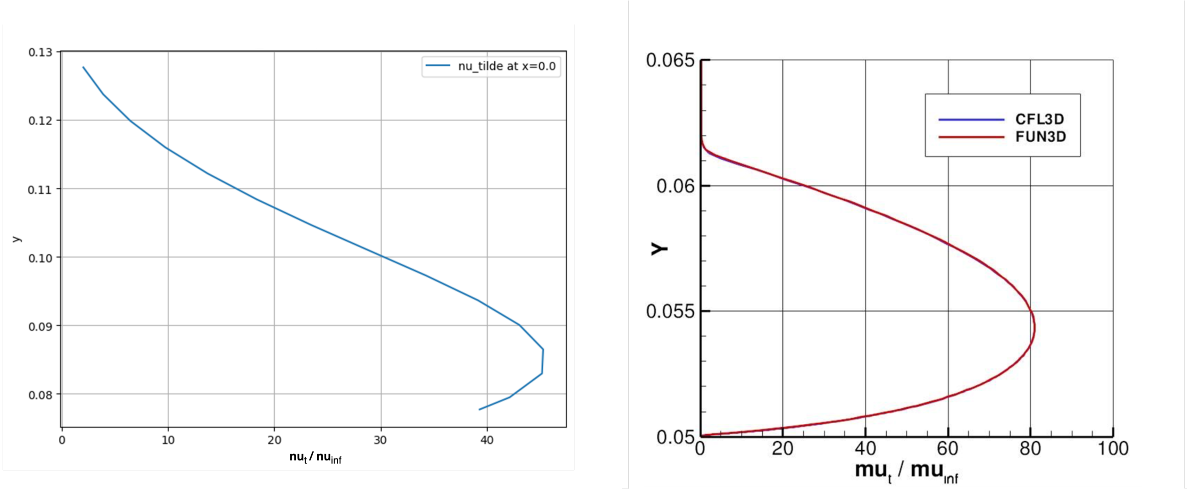


Figure 5.2: Comparison of normalized eddy viscosity over bump apex predicted by 2D PINN (left) and 2D OpenFOAM simulation (right). Differences in y-axis are due to slight differences in Boeing bump geometry used in experimental and CFD campaigns.

Ultimately, due to the difficulty in correctly replicating the results of the OpenFOAM simulations, following from the use of an independent turbulence model, and inherent differences between the experimental and numerical model/domain, a correct trend was sought instead of a exact reproduction. Furthermore, due to the reliance of the turbulence variable on the velocity gradients (through the Boussinesq assumption only), it was unclear whether a 1:1 recreation was achievable. This stems from the efficient approximation of partial derivatives using the autograd function, and the inherent numerical accuracies from such a method. This raised the idea that experimental data may be required for the correct modeling of the turbulence variable (discussed further in section 5.2).

The purpose of the 2D investigation remained largely to calibrate the model, and gain familiarity with its behavior before advancing to a 3D model. For this reason, and due to the lack of validation data for a 2D prediction, the simulated velocity field shown in Figure 5.1 was deemed to pass the eye test, and sufficient at the 2D stage of development.

To confirm that much of the accuracy had been realized, a brief ablation study was performed on the size of the neural network (nodes vs. layers), presented in Figure 5.3.

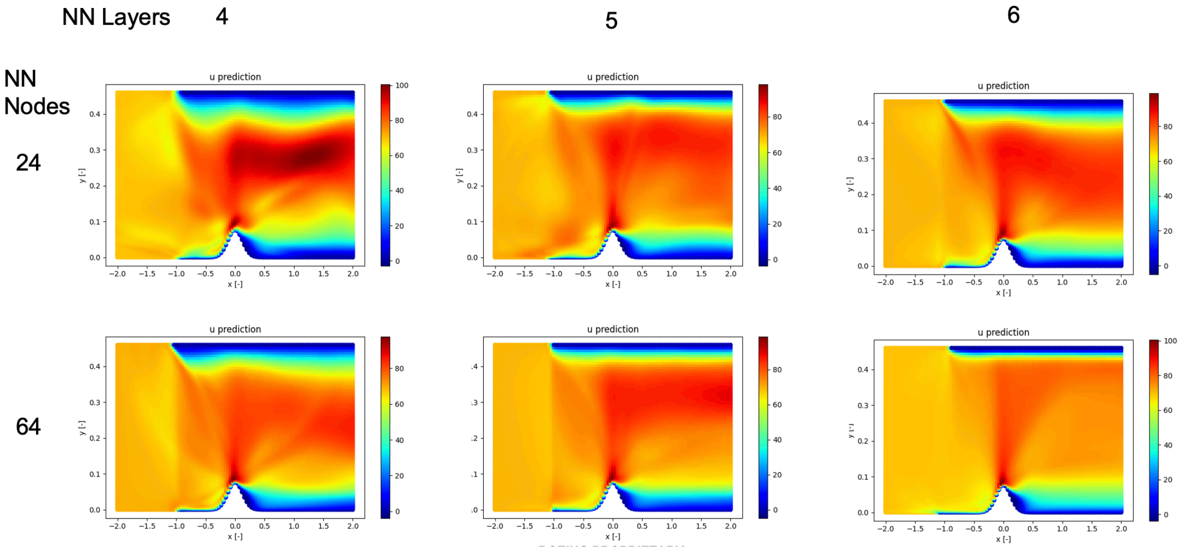


Figure 5.3: Comparative study of architectural size (layers, nodes) and the corresponding effects on streamwise velocity reconstruction accuracy.

As can be seen, this confirms the relation presented in Equation 5.1, showing that NN size improvements only yield limited improvements, after which hyperparameter optimization is required for lower error. Also important to note is the tendency towards ”higher-frequency” solutions observed in the results of the smaller architecture. This highlights the co-dependence between hyperparameters such as the Fourier layer and the network architecture, and thus the need to consistently re-optimize the network parameters.

Not shown in Figure 5.3 are larger combinations of NN nodes (128, 256) and layers (8, 10, 12) for which no noticeable difference was observed to the prediction made with 64 nodes and 6 layers. Similarly, and as done in CFD, a convergence study was similarly performed over collocation point quantity, where no noticeable improvements were yielded after exceeding 10,000 points for this domain size. For simulations of this size, a training time of 3-6 hours was expected, depending on network size, with collocation count remaining constant and a training cycle of 12,000 epochs. These findings are summarized in Table 5.1, where one can see that the largest impact on training time comes from the large increase in node count:

Table 5.1: Training times for 2D PINN model, trained on Nvidia A100 using 10,000 collocation points for 12,000 epochs.

	4 Layers	5 Layers	6 Layers	Units
24 Nodes	214	269	315	[min]
64 Nodes	596	637	724	[min]

The takeaway from this table is the observation that layer count has diminished effects on training time when compared to node count. Furthermore, due to the quick diminishing effects of layer count increases, these are most easily maximized for a set cost, though it must be noted that vanishing gradients owing to excessive layers can have detrimental effects on training as well. Under idealized circumstances, both architectural characteristics are optimized.

5.2. 3D Vector Field Reconstruction

For the depiction of 3D PINN predictions, the Paraview software is used again, where the entire flow field can be easily viewed in vector form. Figure 5.4 shows the prediction over the entire bump-in-channel geometry, with a fully developed boundary layer aft of the bump. The vectors represent the resultant velocity, colored by the streamwise velocity magnitude to display the strong sense of flow reversal commensurate with experimental findings.

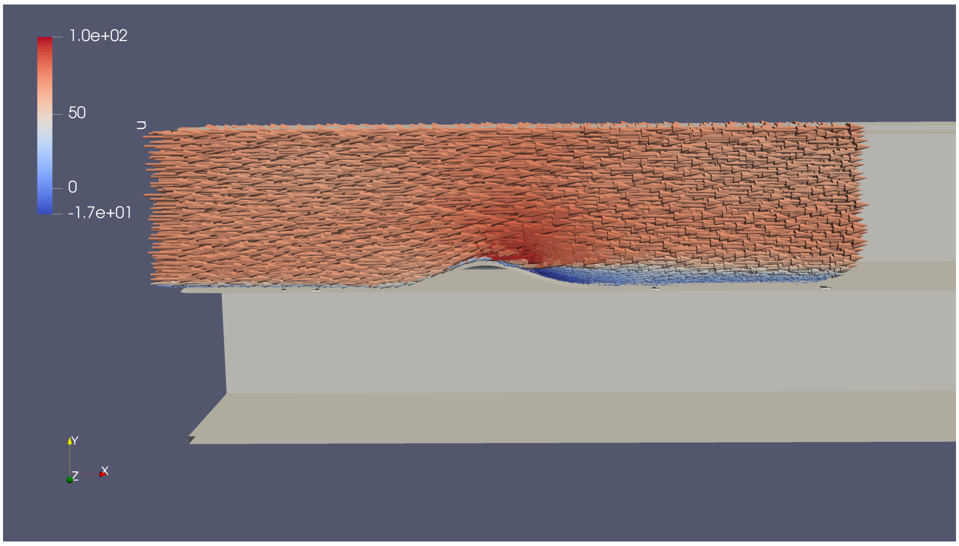


Figure 5.4: Sideview of 3D PINN predicted resultant velocity flow field, colored by streamwise velocity, u, to highlight downstream flow reversal.

The flow can also be colored by different magnitudes, such as the vertical/spanwise velocity, pressure and turbulence viscosity. These are shown in Figure 5.5, and empirically also show consistency with the experimental data.

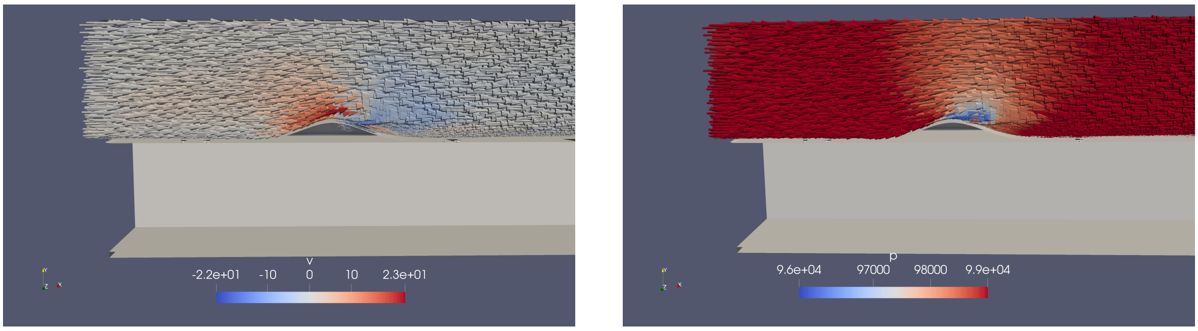


Figure 5.5: Sideview of 3D PINN predicted resultant velocity flow field, colored by vertical velocity, v, (left), and pressure, p, (right)

To provide insight into the training process of the PINN that facilitated these results, Figure 5.6 displays the evolution of the training losses over time. A total of 80,000 epochs are simulated, over which all loss components can be seen to decrease, eventually plateauing long before the 80,000 epoch mark. This excessive training time was used both to resolve the smaller scales which are crucial to the reconstruction accuracy of the flow, as well as to confirm convergence of the simulation.

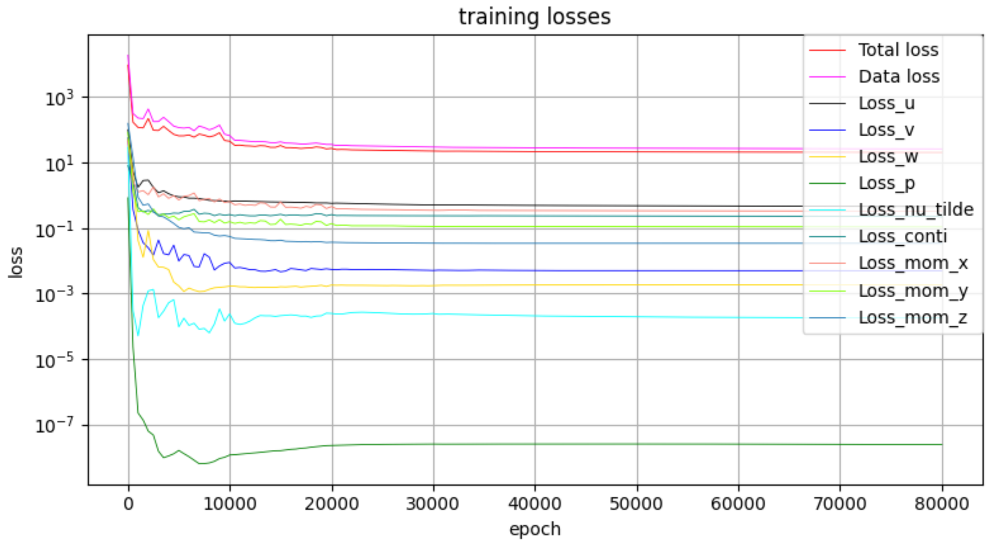


Figure 5.6: PINN training loss over 80,000 epochs to demonstrate converged behavior.

The plot of training loss is among the most crucial to observe during training to ensure the convergence towards a global minimum. An ill-posited problem may convergence towards a minimum that does not satisfy all loss criteria, and this would be observed in the above plot if all losses would not decrease over epochs. The training time for a simulation representative of these results totaled approximately 5,000 minutes.

5.2.1. 3D Flow Field Validation

Following from the successful empirical recreation of experimental data, the next step was to validate the results compared to experimental data upon which the network was not trained. In other words, the PINN's performance was assessed on data it had not previously seen, truly assessing whether the recreated flow field was accurate. Figure 5.7 displays the locations of the PIV planes which were used for downstream training, and those used for testing.

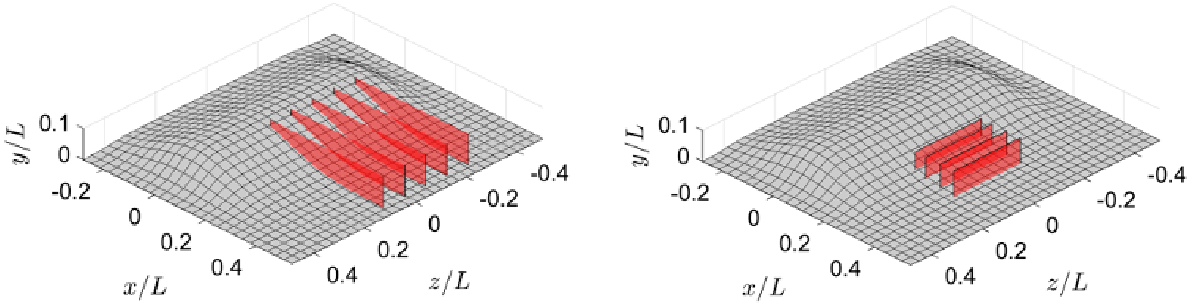


Figure 5.7: Streamwise (left) and spanwise (right) locations of PIV planes used for training and testing, respectively [9].

While the training data contains only 2D2C velocity data upon which the PINN was trained, the test data contains 2D3C stereo PIV data, allowing a quantitative assessment to be made of all 3 velocity components predicted by the model. The spanwise planes used for validation were plotted alongside the PINN prediction at those locations. Figures 5.8, 5.10, and 5.12 display these for all velocity components, namely streamwise, u, vertical, v, and spanwise, w. Each plot contains red dashed, vertical lines indicating where the streamwise training data was located, suggesting regions where the prediction is expected to be consistent with the experimental results.

In the plots that follow, one can see the empirical assessment of the flow field reconstruction in all dimensions, in an upstream view. For all velocities shown, the plane corresponds to the third plane downstream of the bump as depicted in Figure 5.7. In distances, this corresponds to a location at $x/L = 0.306$ post-bump. It should be noted that the final plane was removed from the study due to an error in the experimental data obtained leading to inconsistency between the spanwise and streamwise data. The prediction, and corresponding error plots, for the remaining planes can be found in the appendix (Appendix C).

Due to the need to numerically assess the numerical accuracy, the error between the prediction and experimental results were computed over each spanwise plane. To deliver a statistical representation of the overall plane, box and whisker plots were created to assess the performance at each location. The results of these are displayed in Figures 5.9, 5.11, and 5.13 for each velocity component.

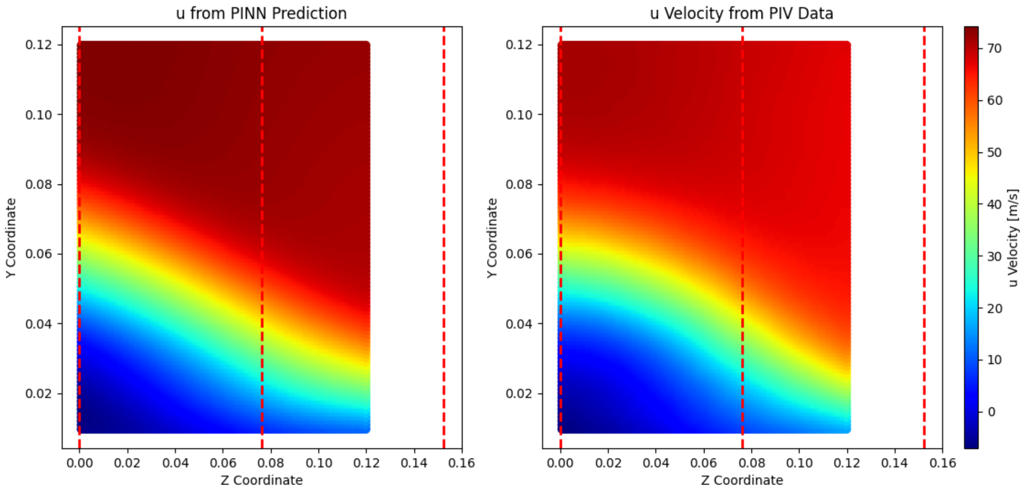


Figure 5.8: Side-by-side comparison of predicted (left) and experimental (right) streamwise velocity flow field at distance $x/L = 0.306$ post-bump apex.

Figure 5.8 displays the PINN prediction alongside the experimental data obtained from the spanwise PIV campaign. In this example, one can see the qualitative success of the reconstruction. Red dashed lines indicate the locations where 2D2C PIV data were fed to the network for training; zones between these dashed lines are interpolated solely from the governing equations.

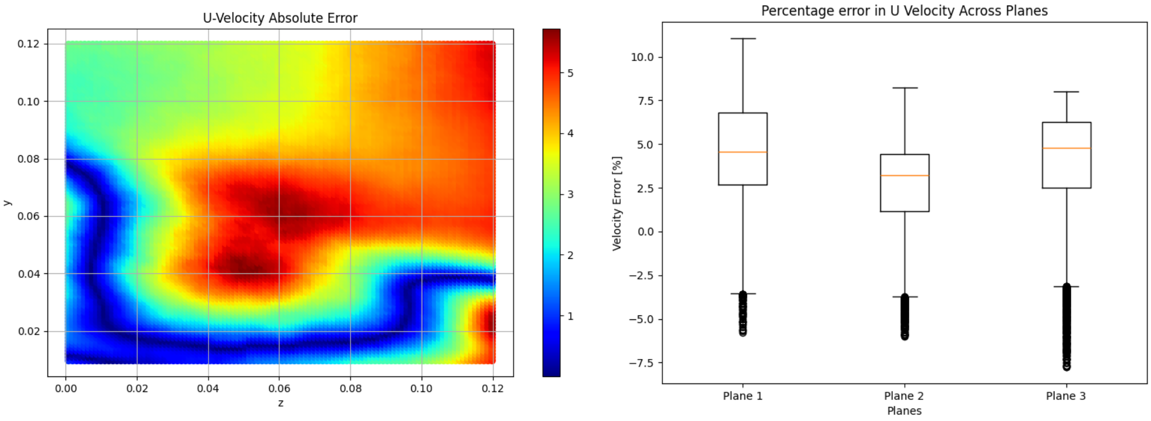


Figure 5.9: Absolute velocity error for streamwise velocity flow field at distance $x/L = 0.306$ post-bump apex (left), and box-and-whisker plot of percent error across all planes (right).

Figure 5.9 displays both the absolute error of the reconstruction, as well as box-and-whisker plots for the reconstruction error over all planes used in the validation study. The boxplot takes the conventional format where the quartiles and maximum/minima are also displayed for each plane. The whiskers here are drawn at the 1.5 IQR value, and values exceeding these limits are denoted as outliers. For all following velocity error plots, the absolute error has been used $(u_{true} - u_{pred})$, maintaining the error in its base units of [ms^{-1}]. For the box and whisker plots, the mean percentage error (MPE) is used instead to maintain information regarding the positive or negative direction of the error:

$$\text{MPE} = \frac{1}{N}\Sigma_{i=1}^{N}\frac{y_{pred_i} - y_{true_i}}{y_{true_i}} \tag{5.2}$$

One can notice the overall strong performance at the left (centerline) and bottom boundaries, where data as well as boundary conditions are provided. Departing from these, the center and right-hand side of the plot show largest regions of error, where the PINN relies almost entirely on the accuracy of the governing equations. The boxplot shows that as a percentage, these errors seldom exceed 10%.

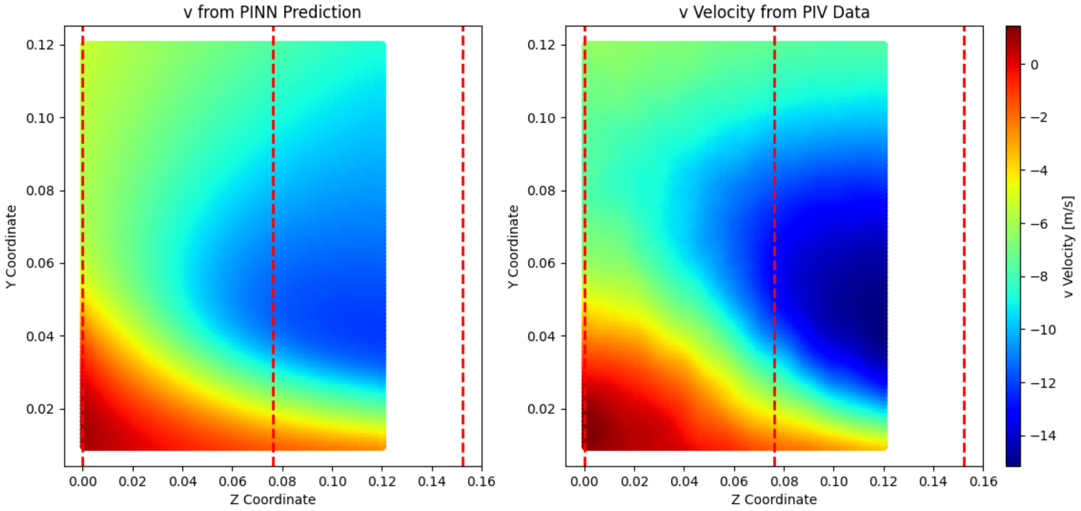


Figure 5.10: Side-by-side comparison of predicted (left) and experimental (right) vertical velocity flow field at distance $x/L = 0.306$ post-bump apex.

Figure 5.10 displays the same side-by-side reconstruction for the vertical velocity, v. Again, a qualitatively strong reconstruction can be observed, despite an underestimation of the magnitude at its maxima and minima.

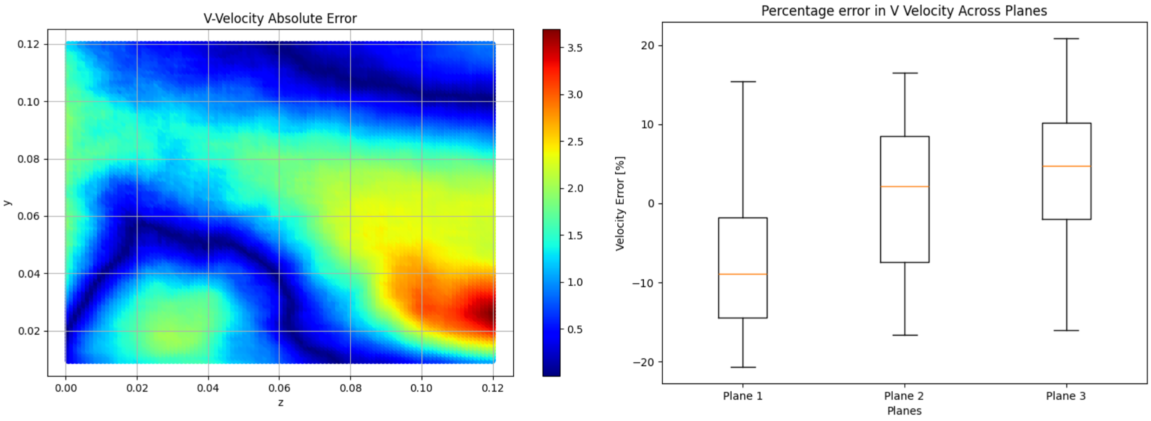


Figure 5.11: Absolute velocity error for vertical velocity flow field at distance $x/L = 0.306$ post-bump apex (left), and box-and-whisker plot of percent error across all planes.

When assessing the reconstruction error, as shown in Figure 5.11, one can see that the variations in error are less drastic, though a higher maximum percent error is observed than when compared to the streamwise velocity, u. This could be attributed to the smaller magnitude of vertical velocity, and its contribution to the loss function dictating the PINN's learning.

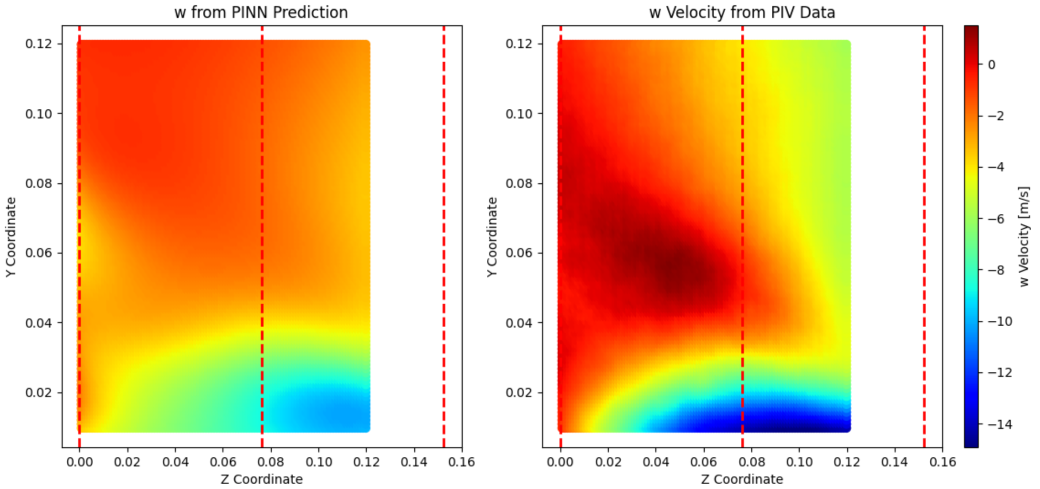


Figure 5.12: Side-by-side comparison of predicted (left) and experimental (right) spanwise velocity flow field at distance $x/L = 0.306$ post-bump apex.

Finally, discussing the side-by-side comparison for the third velocity component, w, one can see a more strained reconstruction. Naturally, since no data was used in training the w component velocity, the PINN relied entirely upon its governing equations to interpret the velocity field. As such, its accuracy relies on the success of the two prior velocity field reconstructions. While the trend is broadly intact, an inability in reconstructing the correct magnitudes is shown, even qualitatively.

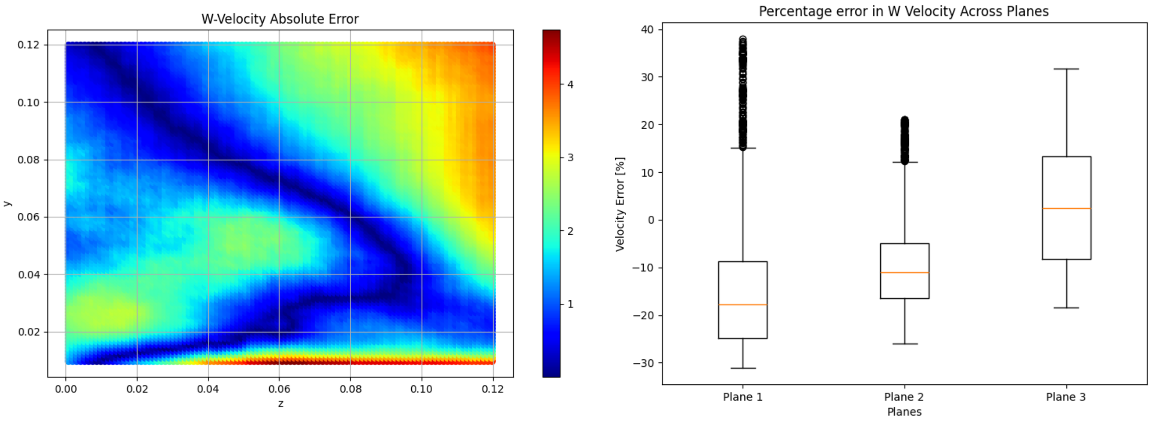
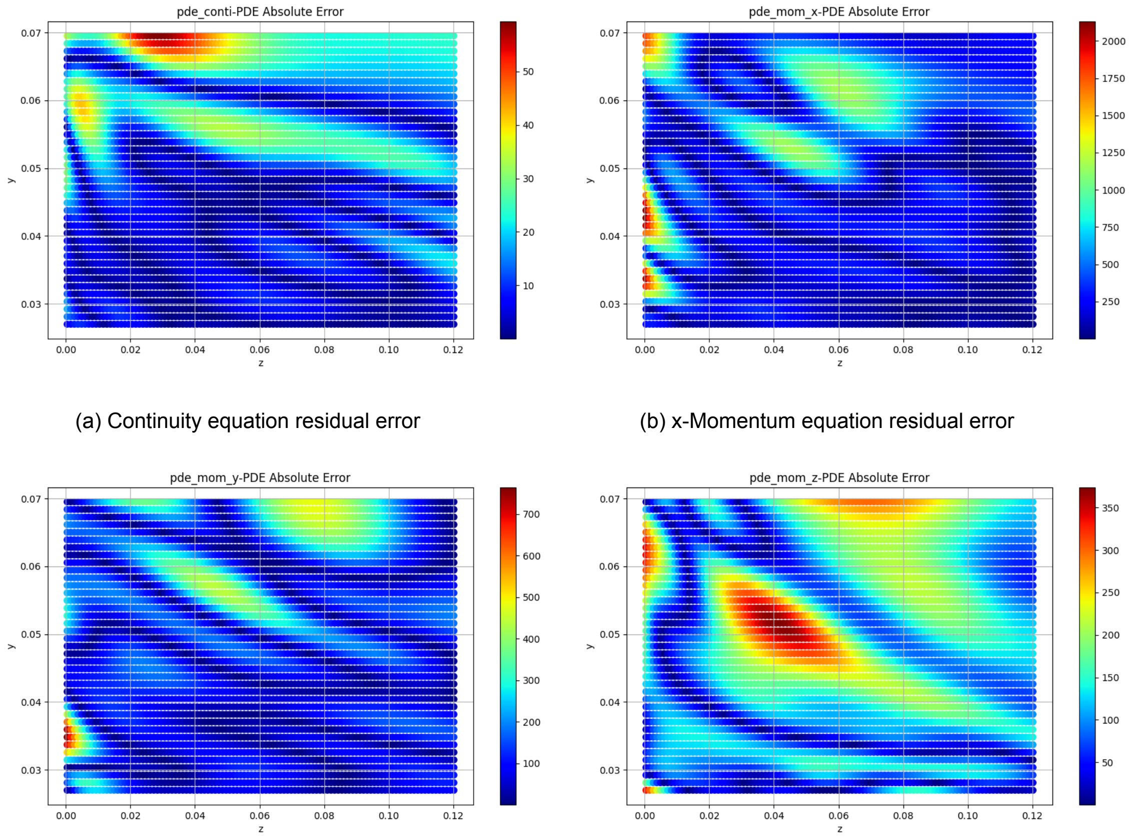


Figure 5.13: Absolute velocity error for spanwise velocity flow field at distance $x/L = 0.306$ post-bump apex (left), and box-and-whisker plot of percent error across all planes.

When quantitatively assessing the reconstruction, one can see an inability to predict the correct velocity magnitudes, particularly within the lower boundary layer where the largest error is seen. The boxplots also display a high number of errors deemed as outliers, with most data points being erroneous when quantitatively compared to the experimental baseline.

To better provide insight to the training process and potentially identify the cause for any discrepancies in velocity prediction, one can similarly plot the residual losses over the landscape to pinpoint the specific governing equation that is poorly represented in the solution. Figure 5.14 displays the residual errors of the governing equations across the first spanwise plane, located at $x/L = 0.208$.



(a) Continuity equation residual error

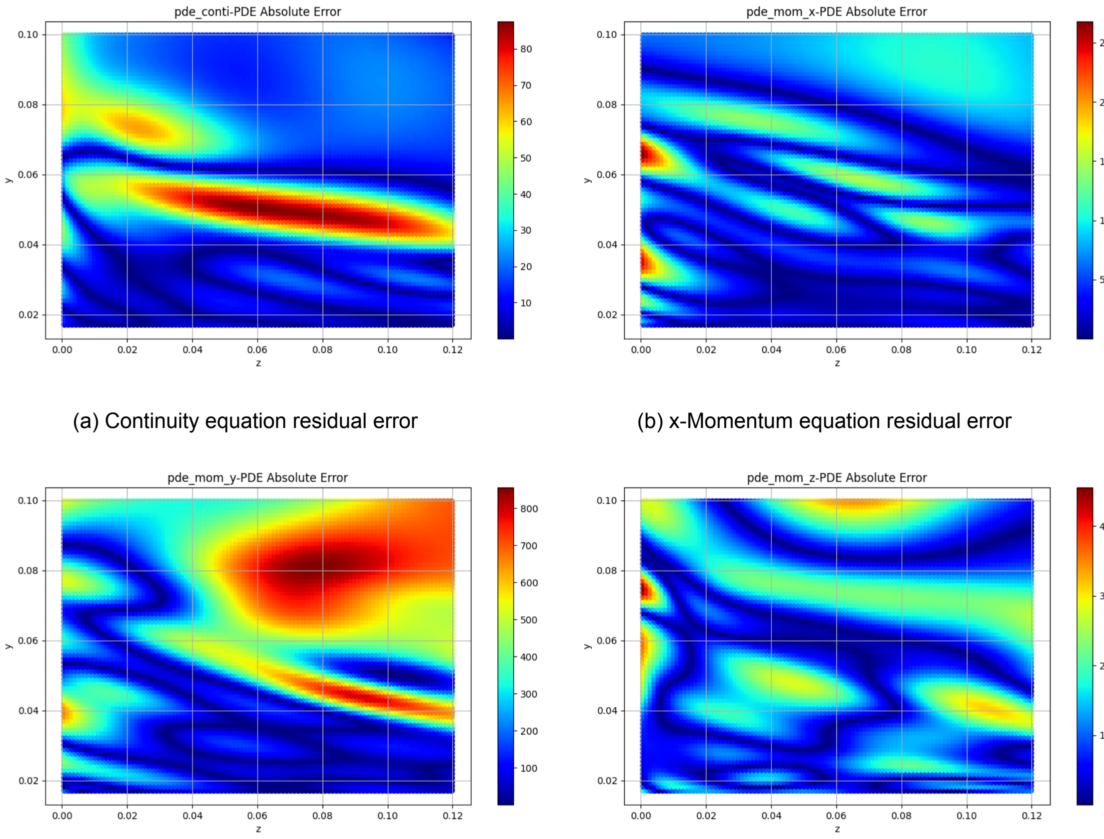
(b) x-Momentum equation residual error

(c) y-Momentum equation residual error

(d) z-Momentum equation residual error

Figure 5.14: Visualization of predicted governing equation errors for all components across first spanwise plane $(x/L = 0.208)$: (a) continuity equation and (b), (c), (d) show momentum equations in x, y, z, respectively.

As one can see, many of the errors in governing equations, particularly across the momentum equations take on similar forms, with zones of high error being shared across different momentum axes. The same process can be performed for the remaining planes of interest, shown in Figure 5.15 for $x/L = 0.250$, and Figure 5.16 for $x/L = 0.306$. One must realize that these provide only localized insights into the overall loss landscape, which is evaluated over the entire domain for the governing equations.



(a) Continuity equation residual error

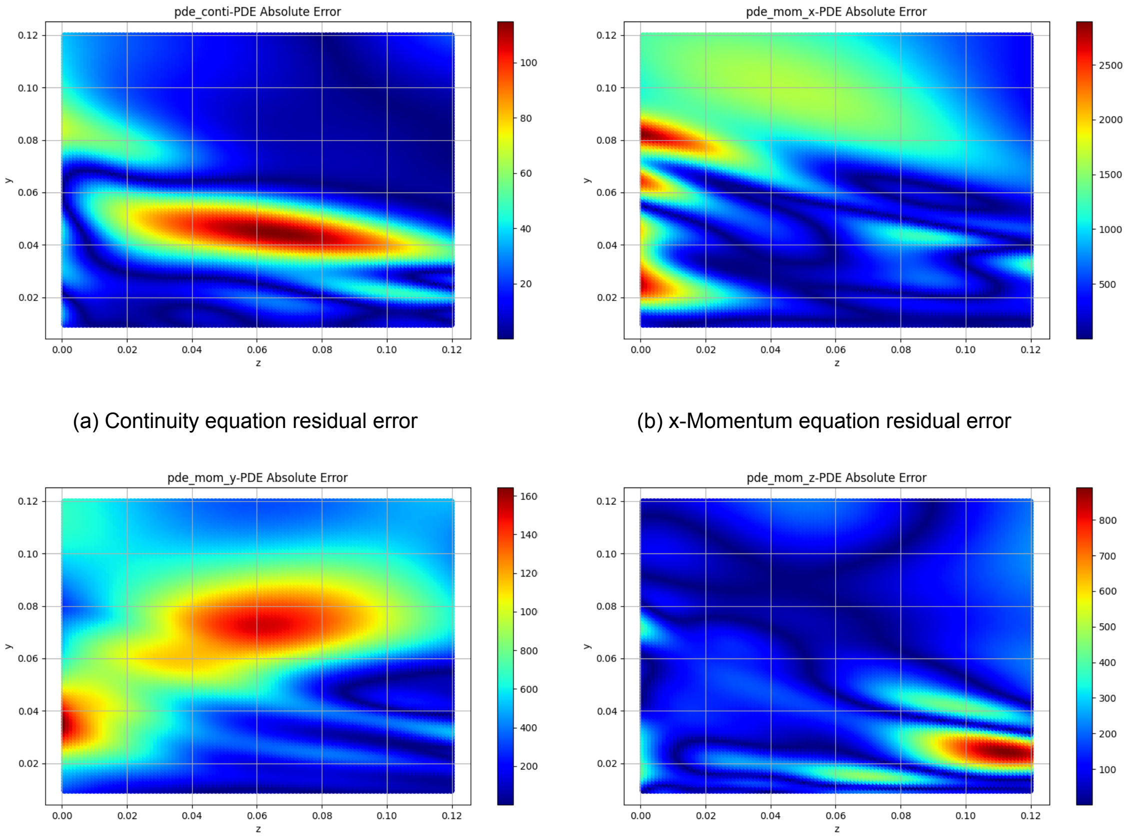
(b) x-Momentum equation residual error

(c) y-Momentum equation residual error

(d) z-Momentum equation residual error

Figure 5.15: Visualization of predicted governing equation errors for all components across second spanwise plane $(x/L = 0.250)$: (a) continuity equation and (b), (c), (d) show momentum equations in x, y, z, respectively.

The second plane, shown above, shows similar coherence across different residual equations, this time with a larger congruence between the continuity and the y-momentum equation. Nevertheless, all equations show a region of similar error across the center-line of the domain $(z/L = 0.0)$, where the physics, boundaries and training data must all be reconciled. Despite this, a strong relation between an individual loss component and the overarching validation errors could not be identified. This reiterates the idea that the pursuit towards a global optimum remains necessary, and potentially identifies the lack of collocation point density or architectural size as the limiting factors in this optimization.



(a) Continuity equation residual error

(b) x-Momentum equation residual error

(c) y-Momentum equation residual error

(d) z-Momentum equation residual error

Figure 5.16: Visualization of predicted governing equation errors for all components across third spanwise plane $(x/L = 0.306)$: (a) continuity equation and (b), (c), (d) show momentum equations in x, y, z, respectively.

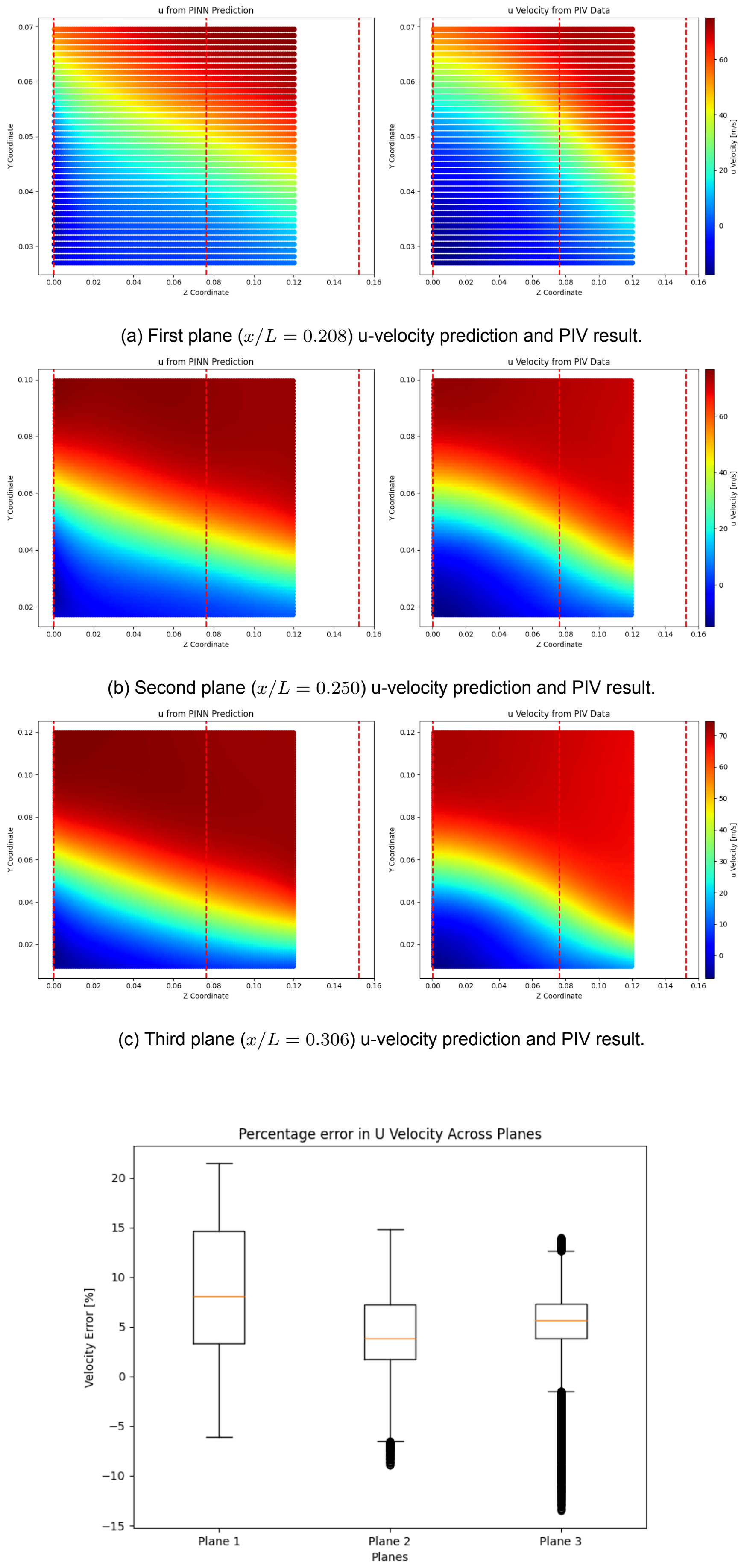
Finally, the third plane is shown in Figure 5.16, where some consistency is shown again in the continuity and momentum equation in x. Notably, these also present the largest residuals errors (note that the loss of the continuity equation was empirically increased by a factor of 10 before being returned in the global loss function), which may indicate the reason for their pervasiveness across planes, as opposed to the remaining momentum equations in y and z. What this section displays above all is the remaining magnitude of the residual equations at the conclusion of training, clearly indicating a lack of global convergence. This will be further discussed in section 5.3 and chapter 6, where it may be a leading contributor to the validation error.

5.3. Minimum Data Solutions and Ablation Study

In the interest of determining the minimum-data solution, several parameters were chosen as part of an ablation study to investigate the relative importance of various measurement techniques. For simplicity, this ablation study was limited to only varying parameters based on the scalars they represent. As such, one study was performed with pressure only, another with PIV data only (center plane), and another with PIV data only (all planes). The objective of this is to investigate the effect that specific data types contribute to the overall PINN solution, and obtain insights into the most relevant measurement techniques.

5.3.1. Training without Pressure Data

First, one can investigate the contributions of the pressure data to the overall solution, and observe the differences when training occurs without information about the pressure field. The pressure variable only appears in the momentum equations in its derivative form across each dimension. As such, it has less influence over the entire governing equations than the velocity gradients, which are present on multiple occasions in both the continuity and momentum equations. The velocity field shown in Figure 5.17, Figure 5.18, and Figure 5.19 display the predicted velocity in the spanwise plane when pressure data is absent from training, alongside the corresponding error for each plane investigated.



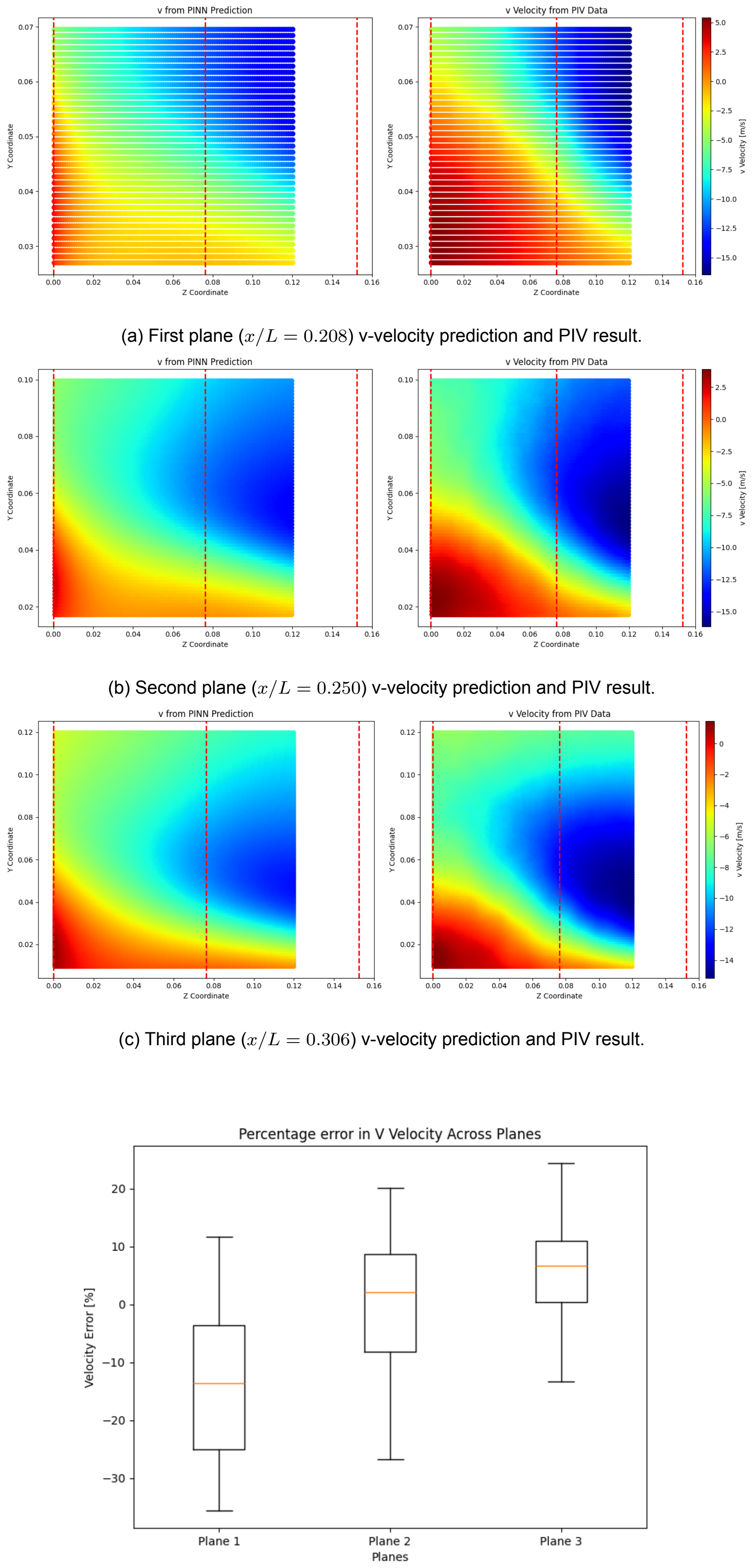
(a) First plane ($x/L = 0.208$) u-velocity prediction and PIV result.

(b) Second plane ($x/L = 0.250$) u-velocity prediction and PIV result.

(c) Third plane ($x/L = 0.306$) u-velocity prediction and PIV result.

(d) Error in streamwise velocity component across all planes

Figure 5.17: Pressure-absent predicted velocity field for u component across all spanwise planes: (a), (b), (c): component-wise velocities in x, y, z, respectively, (d): errors across all planes



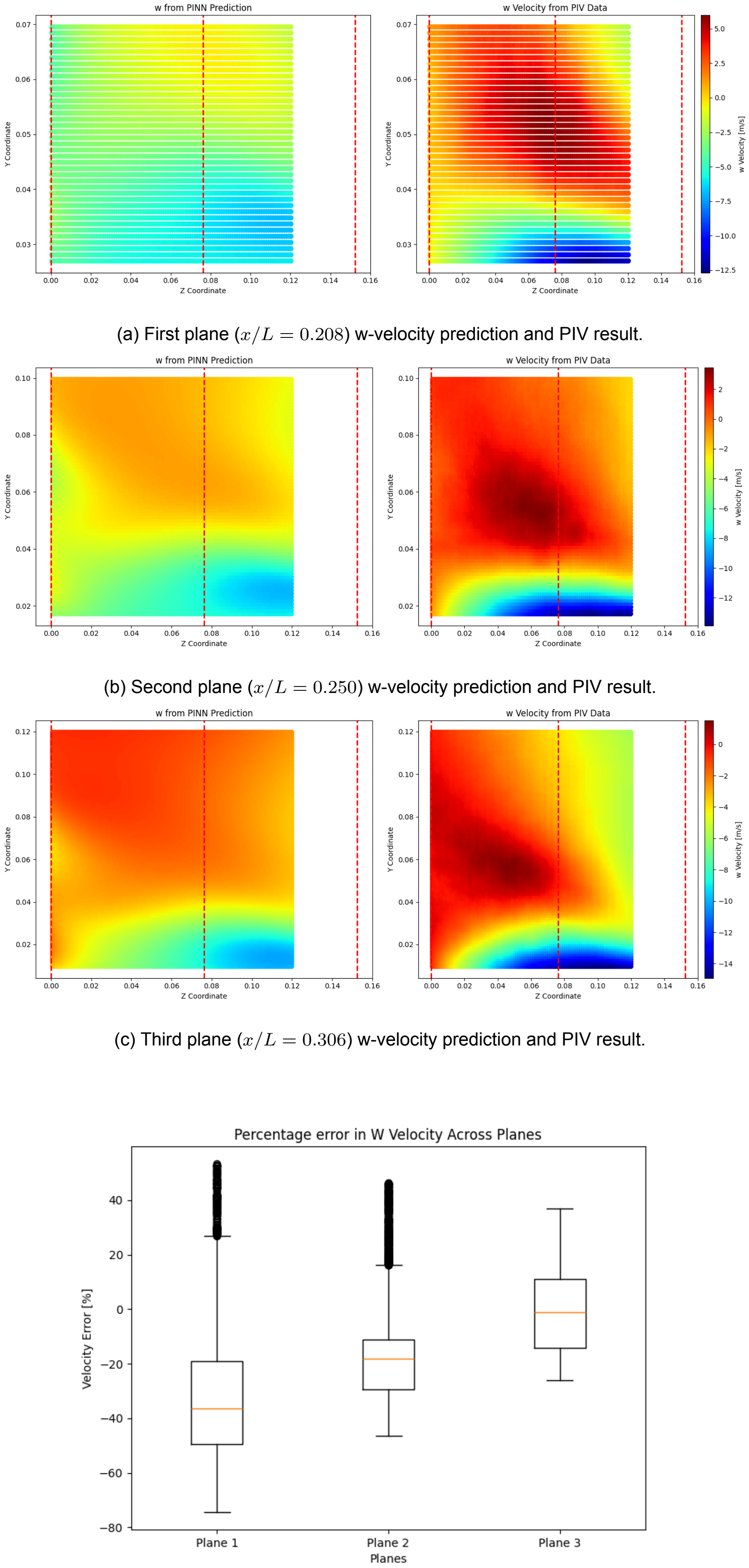
(a) First plane $(x/L = 0.208)$ v-velocity prediction and PIV result.

(b) Second plane $(x/L = 0.250)$ v-velocity prediction and PIV result.

(c) Third plane $(x/L = 0.306)$ v-velocity prediction and PIV result.

(d) Error in streamwise velocity component across all planes

Figure 5.18: Pressure-absent predicted velocity field for v component across all spanwise planes: (a), (b), (c): component-wise velocities in x, y, z, respectively, (d): errors across all planes



(a) First plane ($x/L = 0.208$) w-velocity prediction and PIV result.

(b) Second plane ($x/L = 0.250$) w-velocity prediction and PIV result.

(c) Third plane ($x/L = 0.306$) w-velocity prediction and PIV result.

(d) Error in streamwise velocity component across all planes

Figure 5.19: Pressure-absent predicted velocity field for w component across all spanwise planes: (a), (b), (c): component-wise velocities in x, y, z, respectively, (d): errors across all planes

As can be seen, qualitatively no large differences are observed, due to the reasons mentioned above. To better identify the effects, a quantitative assessment must be performed. This is best done by individually assessing the predictions in the streamwise and spanwise directions over coordinates for which static pressure data exists. These results are displayed side-by-side in Figure 5.20, where the predicted and real values can be seen on the same axes.

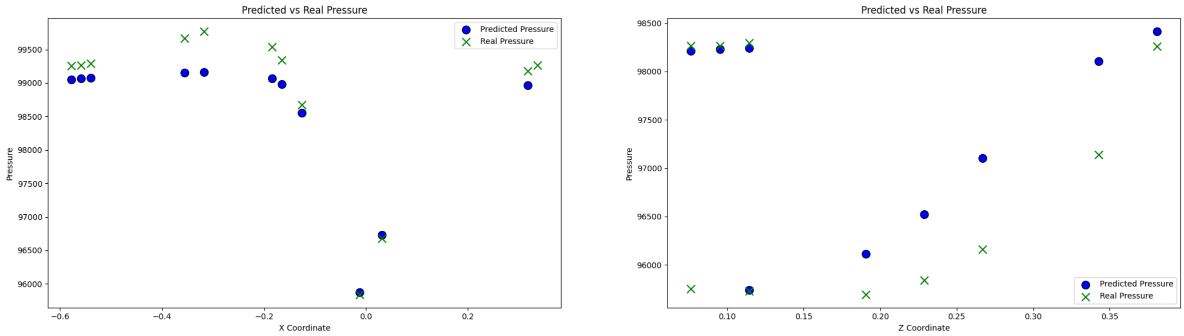


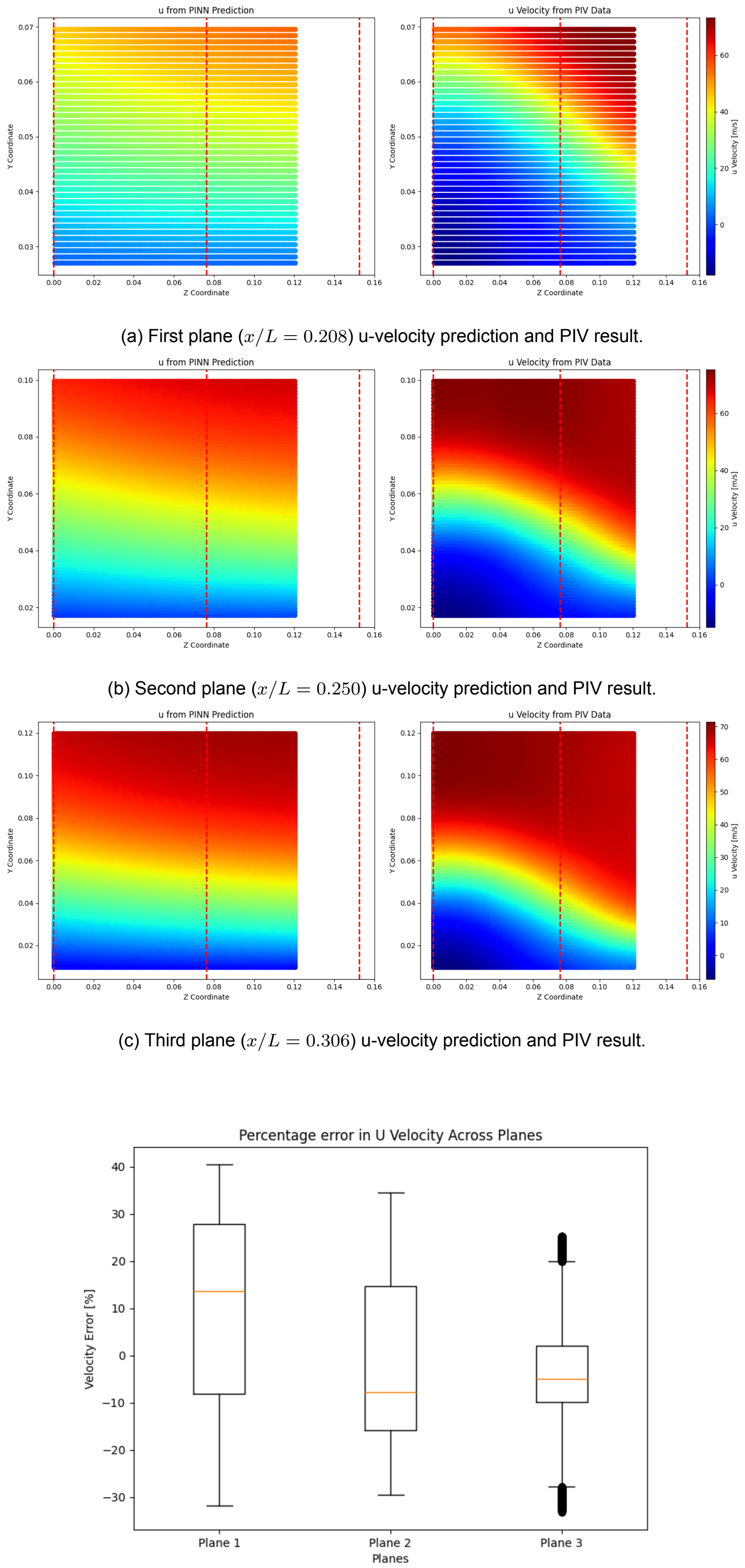
Figure 5.20: Difference between PINN-predicted and experimentally obtained static pressures over bump surface in streamwise (left) and lateral (right) axes.

As can be seen, the exact pressure field is not recreated, and this can only be attributed to an inability for the PINN to sufficiently resolve the momentum equations, in which the contribution of the pressure through its partial derivatives is uniquely present. Nevertheless, the pressure gradients in both axes, which are directly fed into the momentum equations, as opposed to the absolute pressure values show the correct trends. This may be a contributing factor to why the velocity reconstruction can remain sufficiently accurate.

5.3.2. Training Without PIV Data

Finally, the most data minimal outcome (obtained by training with pressure data only) was shown to be most difficult in producing a clear reconstruction of the velocity field.

The results are shown below for all velocity parameters in Figure 5.21, Figure 5.22, and Figure 5.23, respectively:



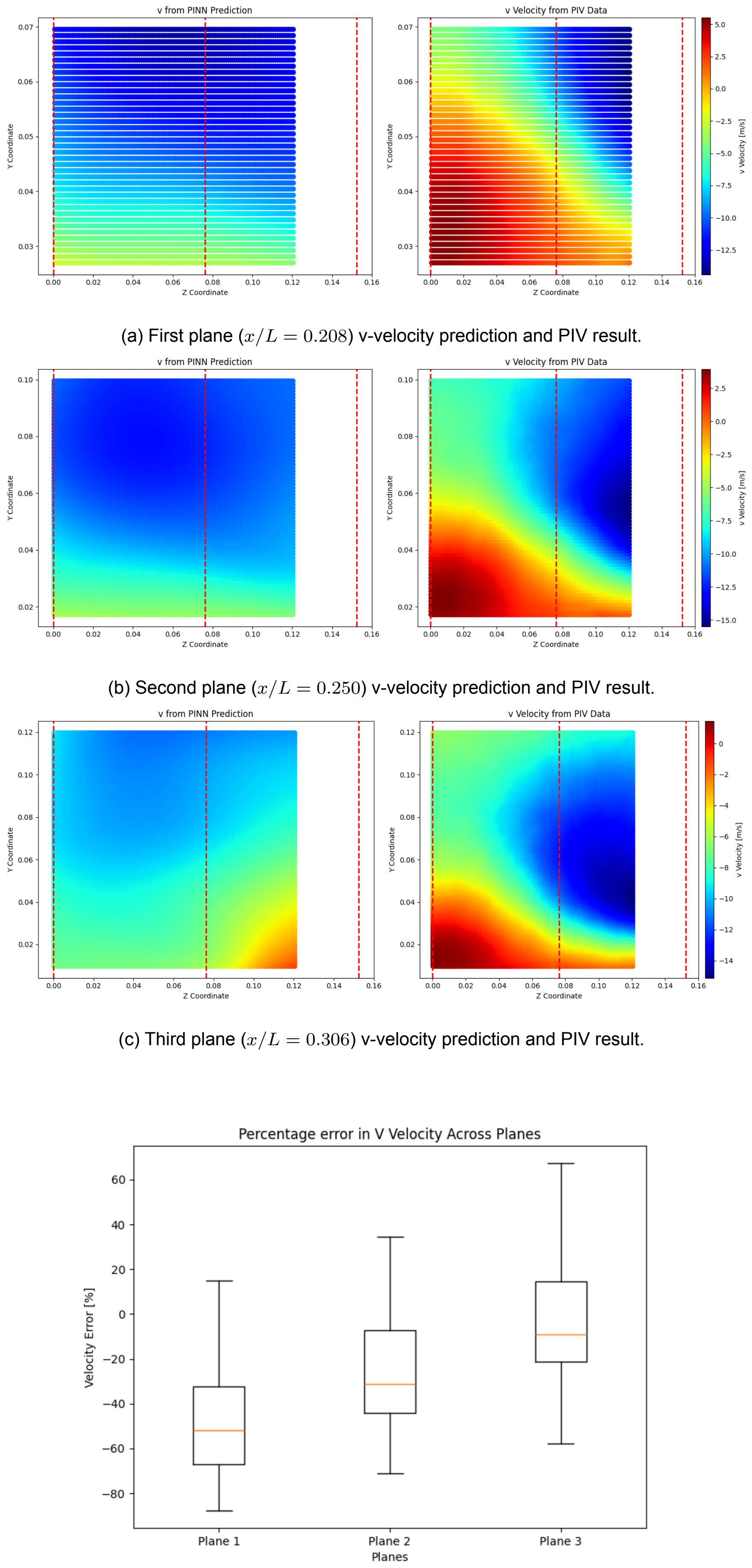
(a) First plane ($x/L = 0.208$) u-velocity prediction and PIV result.

(b) Second plane ($x/L = 0.250$) u-velocity prediction and PIV result.

(c) Third plane ($x/L = 0.306$) u-velocity prediction and PIV result.

(d) Error in streamwise velocity component across all planes

Figure 5.21: PIV-absent predicted velocity field for u component across all spanwise planes: (a), (b), (c): component-wise velocities in x, y, z, respectively, (d): errors across all planes



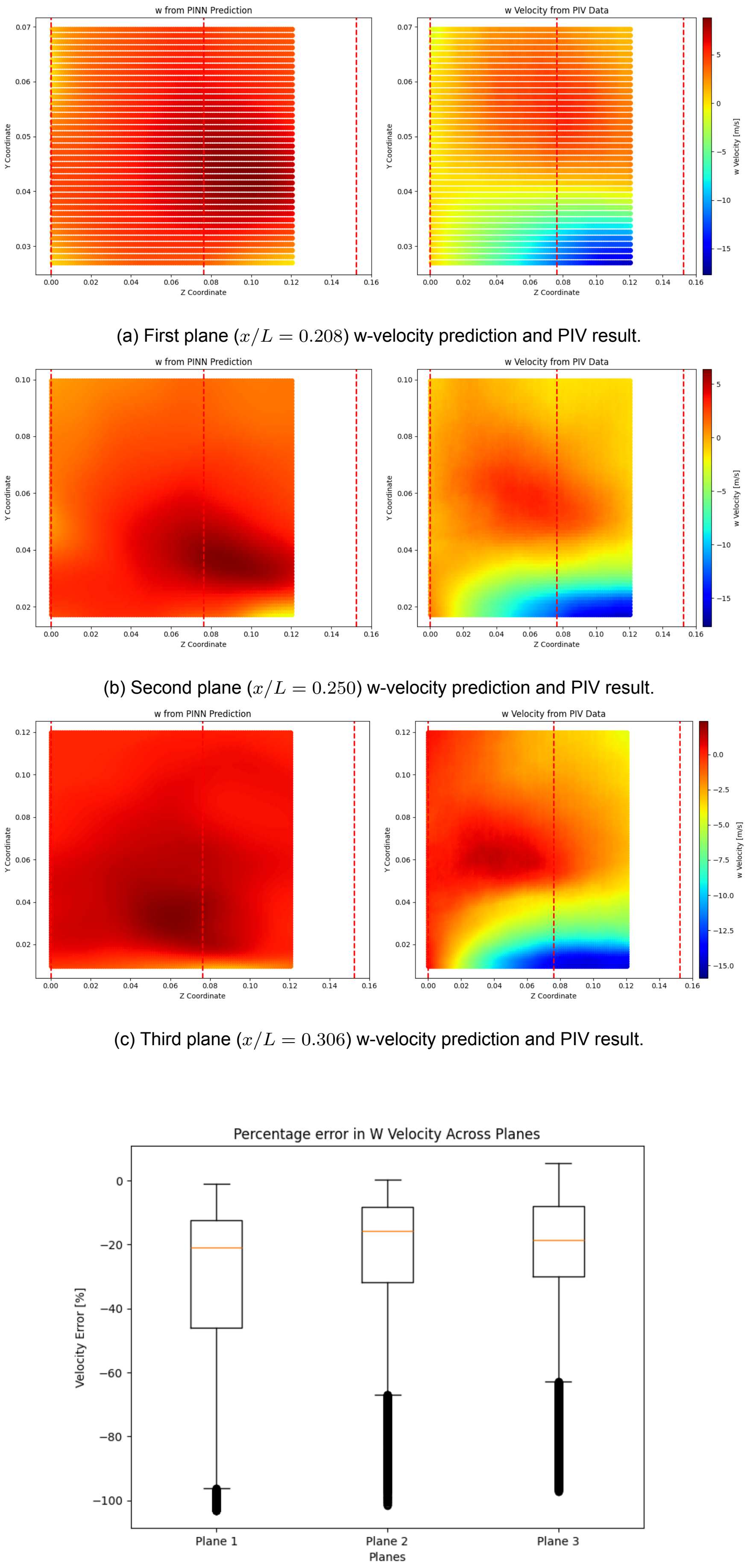
(a) First plane $(x/L = 0.208)$ v-velocity prediction and PIV result.

(b) Second plane $(x/L = 0.250)$ v-velocity prediction and PIV result.

(c) Third plane $(x/L = 0.306)$ v-velocity prediction and PIV result.

(d) Error in streamwise velocity component across all planes

Figure 5.22: PIV-absent predicted velocity field for v component across all spanwise planes: (a), (b), (c): component-wise velocities in x, y, z, respectively, (d): errors across all planes



(a) First plane ($x/L = 0.208$) w-velocity prediction and PIV result.

(b) Second plane ($x/L = 0.250$) w-velocity prediction and PIV result.

(c) Third plane ($x/L = 0.306$) w-velocity prediction and PIV result.

(d) Error in streamwise velocity component across all planes

Figure 5.23: PIV-absent predicted velocity field for w component across all spanwise planes: (a), (b), (c): component-wise velocities in x, y, z, respectively, (d): errors across all planes

As can be seen for the streamwise case, while the correct magnitudes are broadly reconstructed, there remains a lack of accuracy in the curvature and location of velocity gradients. Due to the lack of information in the upper regions of the boundary layer, one can see the PINN's dependence upon missing velocity information that cannot (at this level of complexity) be sufficiently reconstructed through the governing equations.

Extending the analysis to the vertical dimension, one can see that the velocity is poorly reconstructed, with a high validation error being shown across all planes. Since static pressure data was obtained exclusively over the surface of the bump, it provides little information over the velocity gradients in the y-axis, likely hindering the flow field reconstruction. The percentage errors obtained across all planes also remain excessively high (larger than the respectively errors in other components for this training cycle), hinting at the importance of information in the y-dimension.

Finally, when assessing the third component, one can see an overall failure to predict the general flow behavior in its respective dimension. While the magnitudes are broadly correct, the 3-dimensionality of the flow towards the lower y-axis is not reconstructed. One may point to the lack of variation expressed in the z-axis by the streamwise velocity, indicating that the inclusion (or density) of pressure taps was not sufficient for the PINN to extrapolate a velocity gradient in the z-axis. As expected, the validation error remains high across all planes for this configuration, and the results across all components point towards the importance of having information of all types across all spatial axes of the domain.

5.3.3. Performance comparison

To assess the performance of the PINN with training on each data type, the individual performances were assessed for each velocity component. The results are shown in Tables 5.2, and 5.3, where the values are obtained by performing a linear average over all planes, as depicted from the box and whisker plots.

Table 5.2: Average PINN velocity component prediction errors for various training data types over all planes.

	u-velocity	v-velocity	w-velocity
All data	4.18%	8.27%	12.4%
PIV data only	6.30%	10.1%	19.8%
Pressure data only	13.9%	30.5%	26.1%

As can be seen, there is a clear trend displaying the importance of both velocity and pressure data types, with the PINN configuration trained on all data leading to the lowest validation error across all velocity dimensions. Notable also is the increase in error from each velocity axis to the next, with the u-velocity displaying the lowest validation error, next to the v-velocity, and finally the w-velocity. The only exception is for that of the pressure-only trained PINN results, where lack of information given in the y-axis may have caused the y-component velocity to suffer in its reconstruction.

Table 5.3: Maximum PINN velocity component prediction errors for various training data types over all planes.

	u-velocity	v-velocity	w-velocity
All data	11.1%	20.9%	38.0%
PIV data only	21.5%	35.6%	74.4%
Pressure data only	40.5%	87.7%	103.2%

In similar fashion to the average errors by training data type, the maximum error shows similar trends, with provision of all data types leading to the lowest error across the board. In contrast to the average results, for the case of only training on PIV data, the extremes are far more exaggerated in the maximum errors, where the removal of pressure data leads to nearly a factor 2 increase in the maximum error observed. Additionally, the trend exposed for the mean values is also repeated for the maximum validation errors, where the streamwise velocity, u, displays the lowest validation error, followed by v, and w, respectively. Due to the pervasiveness of this trend across all types of training data, one may point to the lower magnitudes of the y- and z-component velocities, and the consequence of this on its loss magnitude during training. Despite each component being normalized at every epoch, the normalization values are obtained at the start of training, and thus the optimization may move away from an equal attention to all parameters. This relates again to the difficulty in asserting a globally minimum solution, discussed further in chapter 6.

5.4. Surface Shear Stress Reconstruction at M=0.1

Given the flow field previously shown in section 5.2, and the low relative error in the lower boundary layer, the partial derivatives of the velocity can be used to predict the shear stress topology over the entire surface (using Equation 4.2). This remains one of the stronger, yet untapped potentials of PINNs, which is their ability to extrapolate flow features from which it is untrained. In both experimental and numerical attempts, modeling the flow shearing over the surface remains a difficult task. The results of this endeavor are shown in Figure 5.24, where it is compared with the results obtained through experimental campaigns. Note that the PINN prediction obtained in this segment is trained on boundary conditions and experimental data corresponding to a freestream velocity of $M = 0.1$. The reason for this is the inability for the PINN to reproduce a sensible shear stress reconstruction at higher velocities.

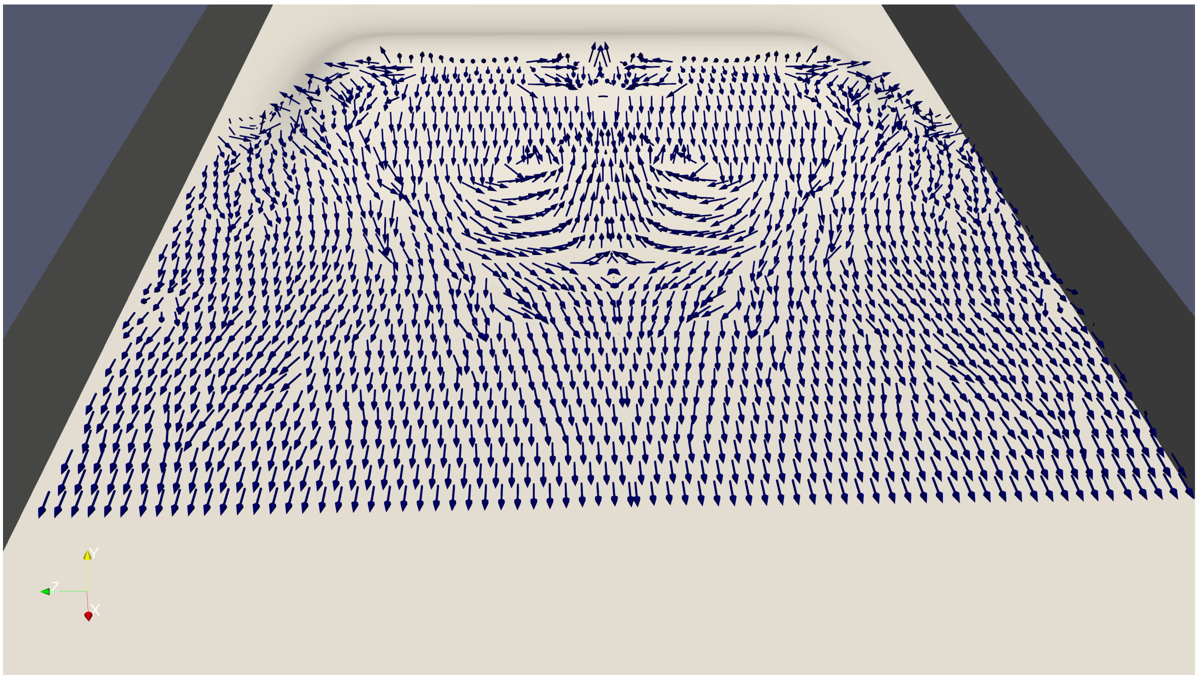


Figure 5.24: Traction vectors of surface shear stresses downstream of bump taken from PINN prediction.

As can be seen, the PINN is able to intelligently interpret the general flow features in regions where the boundary layer is sufficiently recovered. In the region of the separation bubble and flow reversal, the traction vectors over the surface are qualitatively reconstructed to a sufficient degree.

When comparing to the experimental results obtained using fluorescent oil flow imaging, the overall vector direction strongly resembles the shearing topology observed in Figure 2.3. Here one can directly compare the results with time-averaged streamlines, depicting the locations of separation, reattachment and foci. While the location of separation remains difficult to distinguish in the PINN-reconstructed plot, the locations of reattachment, and foci are well visualized. Table 5.4 summarizes the locations as shown in the experimental results, contrasted with those of the PINN prediction. One must be reminded of the differences in streamwise velocity used in the Oil Flow data and the PINN results, explaining why the locations of foci and reattachment occur further downstream for the PINN case, (since the growth of the separation bubble and severity of recirculation with increased freestream velocity is well documented, see chapter 2). Despite this, surprising agreement is shown between the experimental data and PINN prediction, especially when reminded of the relative error in spanwise PIV predictions. For locations predicted experimentally, only a precision of $\pm$ 2 [mm] is available due to the resolution of the measurement technique; the PINN obeys a similar reconstruction prediction as a result of the density of vectors available over the surface. As a direct result of the aforementioned inaccuracy, the PINN's prediction for the location of initial separation, S1, is set to the center-line. Furthermore, perfect symmetry is assumed due to the simulation of a single half-domain in the z-dimension.

Table 5.4: Locations of experimentally-obtained and PINN-predicted separation, reattachment and foci points [9].

	S1	S2	F1	F2	Units	Velocity U_∞
PINN	(0.0, 0.0)	(325, 0.0)	(122.5, 75.0)	(122.5, -75.0)	(x, z) [mm]	M=0.1
Oil Flow	(6.0, 0.0)	(360, 0.0)	(124.7, 113.8)	(129.3, -109.6)	(x, z) [mm]	M=0.2

In addition to a qualitative plot, the traction vectors can also be colored according to the magnitudes of their vectors. The result of this is shown in Figure 5.25, where the largest magnitudes are observed over the bump apex. This empirically aligns with expectations since shearing will be significant before separation, followed by near-zero in value at the point of separation, as located by S1 in Figure 2.3. While the coloring is scaled with the resultant magnitude in all dimensions, it is worth noting that the streamwise component contains the highest influence due to being larger in magnitude.

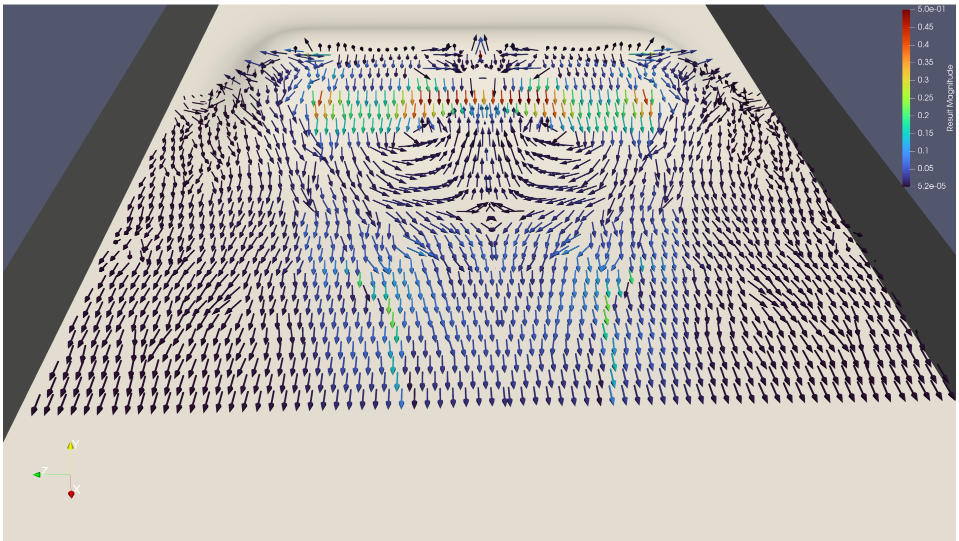


Figure 5.25: Traction vectors of surface shear stresses over bump taken from PINN prediction, colored by magnitude of resultant traction vector.

What one can notice in addition to the large shearing shortly after the apex, is the sudden drop in shearing at the location of flow reversal. In this area, the shearing takes on a near-zero value, owing to the existence of the separation bubble, echoed by the works of Uzun [20]. Furthermore, one can then plot the curvature of the friction coefficient over the surface, as obtained by the shear stress magnitudes. The results of this are shown in Figure 5.26, where an agreement in trend is shown with previous numerical studies (see 2.8, [20]).

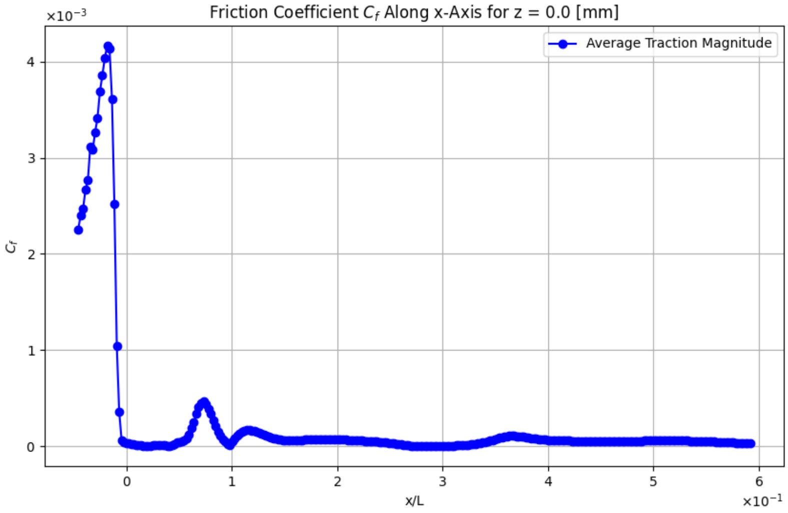


Figure 5.26: Traction vectors of surface shear stresses over bump taken from PINN prediction, colored by magnitude of resultant traction vector.

As can be seen, the expected surge in friction coefficient slightly before the bump apex at $x/L = 0.0$ can be observed, with a reversion to near-zero shortly after. The second, smaller peak is owed to the surge in traction vector magnitude, as seen in Figure 5.25, while the final hump, observed at $x/L \approx 0.37$ broadly corresponds to the area of reattachment (note that an x/L value of 0.37 corresponds to a distance of $x = 0.33$ [mm], as output by the PINN in Table 5.4. In regards to the magnitude of the coefficient of friction, one can refernce Table 2.1, where numerous boundary layer properties were outlined in previous experimental studies. One such parameter is the friction coefficient observed in the developed boundary layer ($x/L = -0.469$) upstream of the bump apex, where a value of 3.10$\cdot 10^{-3}$ was observed. While the PINN prediction for the friction coefficient only begins at an apex upstream distance of $x/L = 0.05$, the equal magnitude mark an encouraging result. Refering to previous advancements in PINN methods applied to PIV data, [37] claim challenges in mirroring the magnitude of the friction coefficient in adverse pressure gradient separated flows. The relative success of this work in mirroring the magnitude despite an unresolved boundary layer shows promising advancements in the possibilities of PINN methods when complemented with sufficient data.

Part III

Closure

6 Conclusion

In this thesis, the potential of Physics-Informed Neural Networks (PINNs) to enhance sparse experimental wind tunnel data and reconstruct dense, 3D flow fields has been thoroughly investigated. By leveraging the Gaussian Boeing bump geometry and the experimental data obtained at Caltech's Lucas Wind Tunnel, University of Notre Dame, and University of Washington, the study aimed to address the challenges of modeling complex, turbulent flow phenomena, such as flow separation, reattachment, and surface shear stress. The ability of PINNs to extract meaningful information from low-fidelity, 2D experimental data and reconstruct higher-dimensional velocity and pressure fields has been a central focus of this work.

The research questions posed at the outset provided a clear structure for exploring the capabilities of PINNs. Specifically, the study assessed the influence of data resolution, the accuracy of reconstructed flow fields, and the feasibility of predicting turbulent shear stress topologies at high Reynolds numbers of up to 3 million.

The findings of this thesis highlight both the strengths and limitations of Physics-Informed Machine Learning in turbulent flow modeling, contributing valuable insights into its role within aerospace design and experimental fluid mechanics. The following sections provide a discussion on the key results, implications, and avenues for future research.

3D flow reconstruction from 2D data

the results presented in this study offer a comprehensive assessment of the Physics-Informed Neural Network's (PINN) ability to reconstruct 3D flow fields from sparse, low-fidelity experimental data. By utilizing downstream spanwise PIV planes for validation, the PINN's performance was rigorously evaluated across all three velocity components: streamwise, vertical, and spanwise.

The findings demonstrate that the PINN successfully captured the dominant trends and structures of the flow field, particularly in regions where training data or boundary conditions were explicitly provided. For the streamwise velocity component, the reconstruction exhibited strong qualitative agreement with experimental data, with minimal error at the center-line and bottom boundaries. Errors tended to grow in regions where the model relied solely on the governing physics, yet these errors remained generally below 10% across the validated planes.

The reconstruction of the vertical velocity component v showed similar qualitative success, though discrepancies were observed in predicting the magnitudes of maxima and minima. Quantitative analysis revealed a slightly higher percentage error compared to u, likely due to the smaller magnitude of v and its relative contribution to the loss function. Although these were normalized in their contribution to the loss, controlling the evolution of each loss term throughout training and ensuring each term is sufficiently optimized remains a difficult task for PINNs.

The greatest challenges emerged in reconstructing the spanwise velocity component, w, where no direct training data was available. Here, the PINN relied entirely on the governing equations to infer the velocity field, resulting in significant inaccuracies, particularly within the lower boundary layer. While the global flow trends were maintained, the inability to accurately reconstruct velocity magnitudes highlights the limitations of the current approach when sparse or incomplete data is provided. The error analysis further revealed a high prevalence of outliers, underscoring the difficulty in resolving this component without additional training data or enhanced physics-based constraints.

In summary, the results confirm that while PINNs exhibit strong potential for reconstructing dense flow fields from sparse measurements, their accuracy is highly dependent on the availability of training data and the complexity of the target flow field. Streamwise and vertical components can be reconstructed with reasonable accuracy, especially in regions with training data and boundary constraints. However, the spanwise velocity component remains a challenge, and this investigation could not conclude whether this could be solved through additional investment in computational effort, or whether the optimization landscape remains ill-defined for a global solution of such scale. Since the accuracy of the physical residuals is dependent upon the accuracy with which partial derivatives are computed, it remains possible that further improvements to the network density, architecture or training scheme can yield improvements in regions absent of experimental data. Most obvious, would be the inclusion of additional training data, which can now be obtained through the experimental setup in the Lucas Wind Tunnel. Additional planes of imaging will directly improve the reconstruction accuracy, with data availability remaining the limiting factor in machine learning methods.

Data Minimal Solutions

The results presented in this section highlight the critical role of data diversity in achieving accurate predictions using Physics-Informed Neural Networks (PINNs). Across all configurations, the inclusion of both pressure and PIV data significantly enhanced the reconstruction of velocity fields in all three spatial dimensions. The errors for the configuration trained on all data types were consistently lower compared to those using only pressure or PIV data, with the streamwise velocity (u) achieving the highest accuracy, followed by the spanwise (v) and vertical (w) components.

When training solely on pressure data, the PINN struggled to resolve velocity gradients, particularly in the y- and z-directions, as the static pressure information provided insufficient information for reconstructing these components. Conversely, training on only PIV data also led to larger errors due to the absence of pressure gradients, which are critical for resolving momentum equations in all dimensions. The observed trends emphasize the limitations of using incomplete data types and highlight the necessity of a balanced dataset for effective PINN training. The optimization towards a global minimum remains a challenge in utilizing PINNs for practical applications.

The analysis also reveals the inherent difficulty in reconstructing the w-component of velocity, which displayed the highest error magnitudes across all configurations. This may be attributed to the lower relative magnitudes of v- and w-velocities and their reduced influence on the overall loss function during training. This imbalance suggests the need for tailored strategies, such as normalization schemes or weighting adjustments, to ensure equal attention to all velocity components.

Overall, the findings demonstrate the importance of comprehensive datasets that integrate both pressure and velocity measurements to fully leverage the potential of PINNs for accurate and robust flow field reconstructions. Contrary to previous claims that the addition of pressure does not yield an improved solution, these insights are crucial for guiding future efforts in improving PINN architectures and training methodologies for complex fluid dynamics problems.

Reconstruction of the surface shear stress topology

The results presented with regards to the surface shear stresses demonstrate the potential of Physics-Informed Neural Networks (PINNs) to reconstruct shear stress topologies over complex flow fields, even in regions untrained by the model. The PINN's predictions align qualitatively well with experimental results, particularly in regions where the boundary layer has recovered. The accurate depiction of flow features, including reattachment points and foci, highlights the model's capability to extrapolate flow characteristics from limited data.

Despite challenges in predicting exact separation locations, the PINN provides valuable insights into surface shear stress distributions. The ability to visualize traction vectors and their magnitudes, as well as to derive the curvature of the friction coefficient, underscores the utility of PINNs in analyzing surface flows. This approach not only complements experimental and numerical methods but also opens avenues for leveraging PINNs in more advanced flow prediction tasks, especially for scenarios where traditional methods face limitations.

The findings suggest that with further refinement, particularly at higher Mach numbers, PINNs may hold promise as robust tools for fluid dynamics research, providing a blend of predictive accuracy and computational efficiency.

Evaluation

Ultimately, this study highlights the capacity for PINNs to reconstruct highly turbulent, complex 3D flows when provided with limited experimental data. The ability for PINNs to extrapolate a velocity-component for which no training data is provided is assessed, where the maximum achievable accuracy is identified for the given dataset. Limitations in the PINN's flowfield reconstruction without increases to computational power or experimental data were established. The importance of diverse training data, as well as adequate spatial distribution of such data across all dimensions is established as a prerequisite to steer the optimization towards a global solution and reduction in validation error. The extension of the reconstructed flow field to the reconstruction of surface shear stresses over the Gaussian Boeing bump are also examined, with encouraging agreement to experimental and numerical results.

Much is left to be desired from PINNs before they can rival contemporary numerical solvers, which remain desirable within industry for their strong combinations of accuracy, cost, and iterability with decades of development and established practices. However, the straightforward capacity for PINNs to integrate experimental data remains unrivaled when compared to the difficulty of performing the same data acquisition in conventional CFD schemes. Despite the shortcomings of PINNs in establishing a global minimum solution, the young and rapidly developing space of Physics-Informed Machine Learning methods remain an attractive contender to bridge the middle-ground between experimental data and numerical solutions. The next chapter will discuss recommendations based on this investigation that may pave the way to a future where continued development for machine learning methods can establish a use-case for itself.

7

Further Work

This section identifies key areas for improving and extending the application of Physics-Informed Neural Networks (PINNs) from their use in predicting 3D flows and surface shear stresses using sparse static pressure and 2D2C Particle Image Velocimetry (PIV) data. A brief comment on continued work at Lucas Wind Tunnel is given, after which the recommendations for improvement are discussed and categorized into two main areas: aerodynamic applications and machine learning improvements.

Continued work at Lucas Wind Tunnel

While this thesis was capable of reproducing flow features with limited training data, the greatest improvement in accuracy would be yielded from a greater availability of data to train on. This remains the largest benefit from an integrated training/testing pipeline at Lucas Wind Tunnel, where the PINN can be iteratively studied for its accuracy with varying quantities fo training data. Recommended further work at Lucas Wind Tunnel will consist of continued PIV imaging on streamwise planes not imaged previously, providing additional input data to the network, and assessing the improvement on accuracy that this yields. Furthermore, the study and enhancement of alternative measurement techniques, which has remained the principal interest of Caltech's collaboration with Boeing over the Gaussian geometry, can investigate the PINN's capacity for integrating less conventional data types (e.g. tufts) and developing a flow geometry.

Areas of Improvement for Aerodynamic Analysis

Extending the PINN framework to accommodate multiple flow configurations is an essential step toward demonstrating its robustness and ability for generalization. Applying the model across variations in geometry and inflow conditions would provide valuable insights into its adaptability to diverse aerodynamic scenarios. By varying parameters such as freestream velocity, surface geometry, and Reynolds number, one could assess how effectively the PINN captures boundary layer development, separation and reversal. This capability would bring PINNs closer to being a general-purpose tool for aerodynamic analysis.

Another promising avenue is integrating alternative data types into the PINN framework to broaden its utility in experimental aerodynamics. For example, using tufts to capture qualitative directional flow data could provide PINNs with a novel input source. This approach would allow the network to accurately predict surface-level velocity fields from data that traditional methods often overlook due to their qualitative nature. Such an extension would open unique opportunities for PINNs in experimental studies that rely on unconventional, or low-cost datasets.

Validating PINN predictions against high-fidelity CFD results is also crucial for benchmarking their accuracy. Cross-validation with CFD solvers would help identify discrepancies and provide insights into areas where the PINN framework requires refinement. This iterative process of comparison and improvement would enhance the reliability of PINNs, particularly in regions with sparse or noisy input data. Moreover, such validation efforts could highlight the conditions under which PINNs can serve as viable alternatives to CFD solvers, particularly in resource-constrained environments.

A key limitation of this work lies in its focus on steady-state flows. Extending the PINN methodology to unsteady and dynamic flow conditions is a logical progression for future research. Incorporating temporal terms into the governing equations or training on time-resolved datasets would enable PINNs to predict transient phenomena, and by addressing these time-dependent behaviors, PINNs could significantly expand their applicability to real-world aerodynamic challenges.

The utility of PINNs in industrial applications, such as aerodynamic shape design, remains an ultimate goal in their application. Straightforward data-acquisition and prediction capabilities make PINNs particularly valuable for industrial design processes, where contemporary numerical methods can possess large barriers to entry for reliable analyses. The ability to optimize parameters such as lift-to-drag ratios or reduce flow separation over surfaces remain crucial for the improved design of aerospace structures. Collaborating with industrial partners could facilitate the application of PINNs to complex aerodynamic problems, showcasing their potential to streamline design cycles and improve accuracy where numerical methods struggle.

Areas of Improvement for Machine Learning

One of the primary limitations of the current PINN framework is the resolution of its predictions, which is constrained by the computational grid. Enhancing the grid density could significantly improve the accuracy and detail of predicted flow fields. The reliance of the PINN on accurate partial derivative computation has been well established in the goal for a globally minimum solution, and lower errors in governing equations. Not unlike a CFD mesh, improved grid spacing would allow the network to better capture essential aerodynamic phenomena.

While the inclusion of the Fourier Feature Layer has substantially improved the network's ability to resolve complex flow features, there remains considerable scope for architectural enhancements. Increasing the size of the network, adopting alternative architectures such as residual-based attention mechanisms, or implementing multi-optimizer schemes could further enhance the expressivity and convergence of PINNs. Adaptive loss weighting schemes that dynamically balance the contributions of physics, data, and boundary condition losses could lead to more efficient optimization processes and better final solutions.

Finally, addressing the computational cost of training PINNs is essential for their widespread adoption. While computational expense is a limiting factor, many industrial partners possess the resources needed to train larger and more complex networks. Exploring distributed training and parallelization strategies could make the training process more efficient, enabling the possibility to deploy PINNs on larger datasets or more intricate flow configurations. By addressing these challenges, the PINN framework can continue to evolve and provide increasingly powerful tools for aerodynamic and fluid dynamics research.

By addressing these recommendations, future research can build upon this work, advancing the fidelity and utility of PINNs for both academic and industrial fluid dynamics applications. This dual focus on aerodynamic challenges and machine learning advancements ensures a holistic approach to overcoming the current limitations of PINNs.

References

[1] Owen Williams et al. “Experimental study of a CFD validation test case for turbulent separated flows”. In: *AIAA Scitech 2020 Forum*. 2020, p. 0092.

[2] Owen J. Williams et al. “Characterization of separated flowfield over Gaussian speed-bump CFD validation geometry”. In: *AIAA Scitech 2021 Forum*. DOI: 10.2514/6.2021-1671. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2021-1671. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2021-1671.

[3] Owen J. Williams et al. “Comparison of hill-type geometries for the validation and advancement of turbulence models”. In: *AIAA SCITECH 2022 Forum*. DOI: 10.2514/6.2022-1032. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2022-1032. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2022-1032.

[4] J. Kim, S. J. Kline, and J. P. Johnston. “Investigation of a Reattaching Turbulent Shear Layer: Flow Over a Backward-Facing Step”. In: *Journal of Fluids Engineering* 102.3 (Sept. 1980), pp. 302–308.

[5] B. Armaly et al. “Experimental and Theoretical Investigation of Backward-Facing Step Flow”. In: *Journal of Fluid Mechanics* 127 (Jan. 1983), pp. 473–496. DOI: 10.1017/S0022112083002839.

[6] B. Wasistho and K. D. Squires. “Prediction of turbulent separation over a backward-facing smooth ramp”. In: *Journal of Turbulence* 6 (2005), N1. DOI: 10.1080/14685240500055012. eprint: https://doi.org/10.1080/14685240500055012. URL: https://doi.org/10.1080/14685240500055012.

[7] Daniel Simmons, Flint Thomas, and Thomas Corke. “Evidence of Surface Curvature Effects in Smooth Body Flow Separation Experiments”. In: June 2019. DOI: 10.2514/6.2019-2849.

[8] Jonathan Naughton, David Greenblatt, and S. Viken. “Skin Friction Measurements on the NASA Hump Model”. In: *Aiaa Journal - AIAA J* 44 (June 2006), pp. 1255–1265. DOI: 10.2514/1.14192.

[9] Patrick Gray et al. *Turbulence Model Validation Through Joint Experimental Computational Studies of Separated Flow Over A Three-Dimensional Tapered Bump: Part I - Experimental Investigation*. 2023.

[10] Patrick D. Gray et al. “Experimental and Computational Evaluation of Smooth-Body Separated Flow over Boeing Bump”. In: *AIAA AVIATION 2023 Forum*. DOI: 10.2514/6.2023-3981. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2023-3981. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2023-3981.

[11] J P Slotnick et al. *CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences*. 2014.

[12] Andrew Cary, Andrew Dorgan, and Mori Mani. “Towards Accurate Flow Predictions Using Unstructured Meshes”. In: *19th AIAA Computational Fluid Dynamics*. DOI: 10.2514/6.2009-3650. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2009-3650. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2009-3650.

[13] Daniel Simmons, Flint Thomas, and Thomas Corke. “A Smooth Body, Large-Scale Flow Separation Experiment”. In: Jan. 2018. DOI: 10.2514/6.2018-0572.

[14] Patrick D. Gray et al. “Benchmark Characterization of Separated Flow Over Smooth Gaussian Bump”. In: *AIAA AVIATION 2022 Forum*. DOI: 10.2514/6.2022-3342. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2022-3342. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2022-3342.

[15] A. E. Perry and H. G. Hornung. “Some aspects of three-dimensional separation. II - Vortex skeletons”. In: 1984. URL: https://api.semanticscholar.org/CorpusID:126955195.

[16] A. E. Perry and Min S. Chong. “Topology of flow patterns in vortex motions and turbulence”. In: *Flow Turbulence and Combustion* 53 (1994), pp. 357–374. URL: https://api.semanticscholar.org/CorpusID:122132406.

[17] Daniel J. Simmons. “An Experimental Investigation of Smooth-Body Flow Separation”. In: (July 2020). DOI: 10.7274/zs25x63732m. URL: https://curate.nd.edu/articles/thesis/An_Experimental_Investigation_of_Smooth-Body_Flow_Separation/24862563.

[18] Jean Delery. *Three-dimensional separated flow topology critical points, separation lines and vortical structures*. City, State or Country: London : ISTE ; Hoboken, N.J. : Wiley, 2013., 2013. URL: https://search.library.wisc.edu/catalog/9911071186302121.

[19] Matthew Lakebrink. *Turbulence Model Validation through Joint Experimental / Computational Studies of Separated Flow Over a Three-Dimensional Tapered Bump: Part II - Computational Investigation*. 2024.

[20] Ali Uzun and Mujeeb R Malik. “High-fidelity simulation of turbulent flow past Gaussian bump”. In: *Theor. Comput. Fluid Dyn* 35 (2022), pp. 293–329.

[21] Salvatore Cuomo et al. “Scientific machine learning through physics–informed neural networks: Where we are and what’s next”. In: *Journal of Scientific Computing* 92.3 (2022), p. 88.

[22] 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.

[23] Shengze Cai et al. “Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks”. In: *Journal of Fluid Mechanics* 915 (Mar. 2021). ISSN: 1469-7645. DOI: 10.1017/jfm.2021.135. URL: http://dx.doi.org/10.1017/jfm.2021.135.

[24] Shengze Cai et al. “Physics-Informed Neural Networks for Heat Transfer Problems”. In: *Journal of Heat Transfer* 143.6 (Apr. 2021), p. 060801. ISSN: 0022-1481. DOI: 10.1115/1.4050542. eprint: https://asmedigitalcollection.asme.org/heattransfer/article-pdf/143/6/060801/6688635/ht_143_06_060801.pdf. URL: https://doi.org/10.1115/1.4050542.

[25] 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 (Jan. 2022), p. 017116. ISSN: 1070-6631. DOI: 10.1063/5.0078143. eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0078143/16629223/017116_1_online.pdf. URL: https://doi.org/10.1063/5.0078143.

[26] Sifan Wang et al. *An Expert’s Guide to Training Physics-informed Neural Networks*. 2023. arXiv: 2308.08468 [cs.LG].

[27] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: 1912.01703 [cs.LG].

[28] Nasim Rahaman et al. *On the Spectral Bias of Neural Networks*. 2019. arXiv: 1806.08734 [stat.ML].

[29] Sifan Wang, Hanwen Wang, and Paris Perdikaris. “On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks”. In: *Computer Methods in Applied Mechanics and Engineering* 384 (Oct. 2021), p. 113938. ISSN: 0045-7825. DOI: 10.1016/j.cma.2021.113938. URL: http://dx.doi.org/10.1016/j.cma.2021.113938.

[30] Shaikhah Alkhadhr and Mohamed Almekkawy. “A Combination of Deep Neural Networks and Physics to Solve the Inverse Problem of Burger’s Equation”. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. 2021, pp. 4465–4468. DOI: 10.1109/EMBC46164.2021.9630259.

[31] 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 (Mar. 2024), p. 065304. DOI: 10.1088/1361-6501/ad3307. URL: https://dx.doi.org/10.1088/1361-6501/ad3307.

[32] Amirhossein Arzani, Jian-Xun Wang, and Roshan M. D'Souza. "Uncovering near-wall blood flow from sparse data with physics-informed neural networks". In: *Physics of Fluids* 33.7 (July 2021), p. 071905. ISSN: 1070-6631. DOI: 10.1063/5.0055600. eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0055600/19768281/071905_1_online.pdf. URL: https://doi.org/10.1063/5.0055600.

[33] Shengfeng Xu et al. "A practical approach to flow field reconstruction with sparse or incomplete data through physics informed neural network." In: *Acta Mech Sinica* 39.3 (Nov. 2022), p. 322302. ISSN: 1614-3116. DOI: 10.1007/s10409-022-22302-x. URL: https://doi.org/10.1007/s10409-022-22302-x.

[34] Di Fan et al. "Comparative assessment for pressure field reconstruction based on physics-informed neural network". In: *Physics of Fluids* 35.7 (July 2023), p. 077116. ISSN: 1070-6631. DOI: 10.1063/5.0157753. eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0157753/18051136/077116_1_5.0157753.pdf. URL: https://doi.org/10.1063/5.0157753.

[35] Muharrem Hilmi Aksoy et al. "Reconstruction of flow field with missing experimental data of a circular cylinder via machine learning algorithm". In: *Physics of Fluids* 35.11 (Nov. 2023), p. 114110. ISSN: 1070-6631. DOI: 10.1063/5.0176637. eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0176637/18218127/114110_1_5.0176637.pdf. URL: https://doi.org/10.1063/5.0176637.

[36] 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. URL: https://dx.doi.org/10.1088/1361-6501/aca9eb.

[37] B. Steinfurth and J. Weiss. "Assimilating experimental data of a mean three-dimensional separated flow using physics-informed neural networks". In: *Physics of Fluids* 36.1 (Jan. 2024), p. 015131. ISSN: 1070-6631. DOI: 10.1063/5.0183463. eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0183463/18626498/015131_1_5.0183463.pdf. URL: https://doi.org/10.1063/5.0183463.

[38] S. Hanrahan, M. Kozul, and R.D. Sandberg. "Studying turbulent flows with physics-informed neural networks and sparse data". In: *International Journal of Heat and Fluid Flow* 104 (2023), p. 109232. ISSN: 0142-727X. DOI: https://doi.org/10.1016/j.ijheatfluidflow.2023.109232. URL: https://www.sciencedirect.com/science/article/pii/S0142727X23001315.

[39] M. Raffel et al. *Particle Image Velocimetry: A Practical Guide*. Jan. 2007. ISBN: 978-3-540-72307-3. DOI: 10.1007/978-3-540-72308-0.

[40] William Thielicke and René Sonntag. "Particle Image Velocimetry for MATLAB: Accuracy and enhanced algorithms in PIVlab". In: *Journal of Open Research Software* 9.1 (May 2021), p. 12. DOI: 10.5334/jors.334. URL: https://doi.org/10.5334%2Fjors.334.

[41] Julian M. Humml et al. "Development of machine learning tools for aerospace design: wind tunnel investigations on a speed bump model". In: *AIAA SCITECH 2024 Forum*. DOI: 10.2514/6.2024-2834. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2024-2834. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2024-2834.

[42] Jose Florido et al. "Investigating Guiding Information for Adaptive Collocation Point Sampling in PINNs". In: *Computational Science – ICCS 2024*. Springer Nature Switzerland, 2024, pp. 323–337. ISBN: 9783031637599. DOI: 10.1007/978-3-031-63759-9_36. URL: http://dx.doi.org/10.1007/978-3-031-63759-9_36.

[43] Vincent Vanhoucke, Andrew W. Senior, and Mark Z. Mao. "Improving the speed of neural networks on CPUs". In: 2011. URL: https://api.semanticscholar.org/CorpusID:15196840.

[44] Pratik Rathore et al. "Challenges in Training PINNs: A Loss Landscape Perspective". In: *Preprint* (2024). Available on arXiv. URL: https://arxiv.org/abs/2402.01868.

A

Workflow diagram

(See next page)

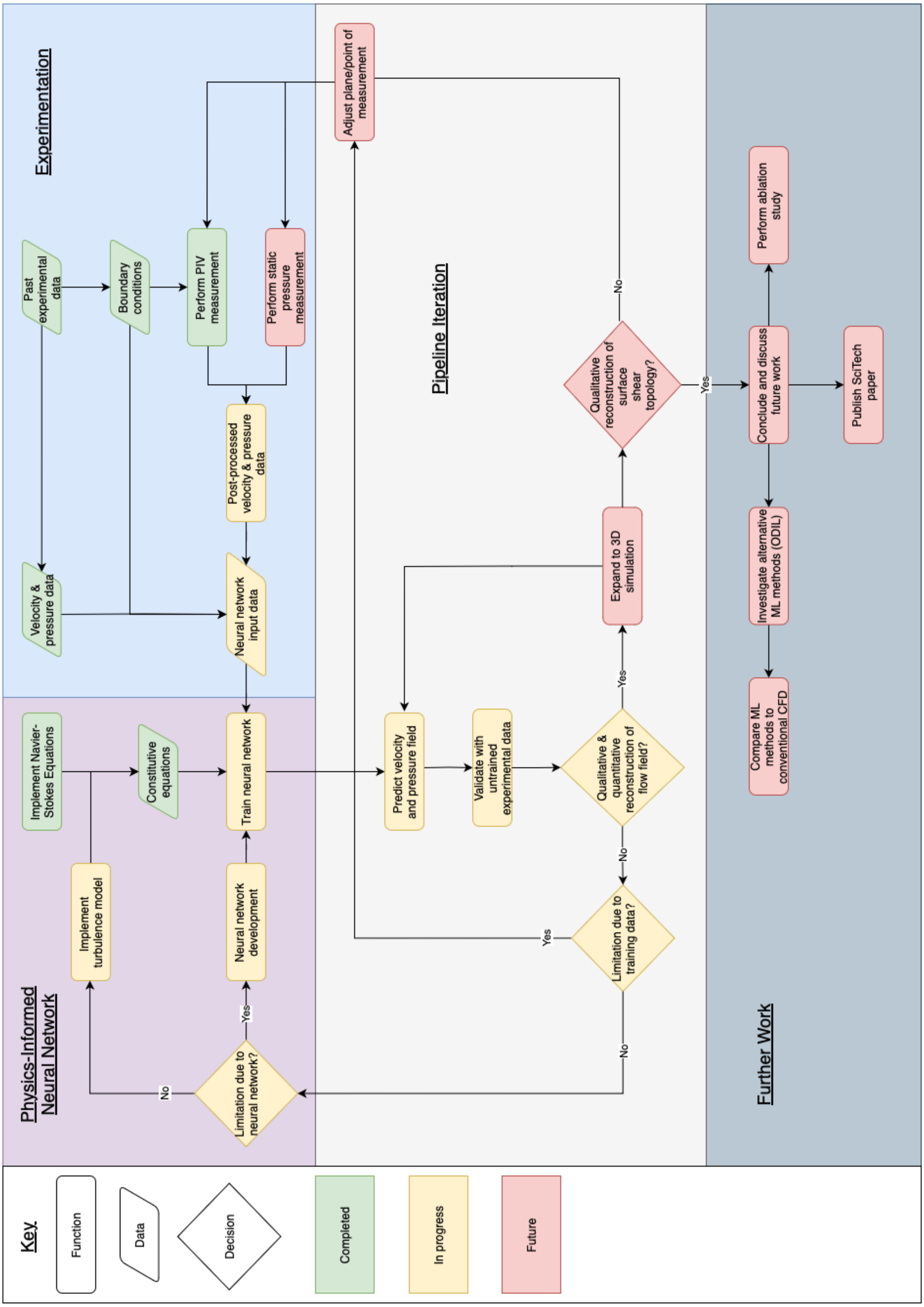

Experimentation
Physics-Informed Neural Network
Pipeline Iteration
Further Work
Past experimental data
Boundary conditions
Perform PIV measurement
Perform static pressure measurement
Adjust plane/point of measurement
Post-processed velocity & pressure data
Velocity & pressure data
Neural network input data
Implement Navier-Stokes Equations
Constitutive equations
Train neural network
Implement turbulence model
Neural network development
Limitation due to neural network?
Yes
No
Predict velocity and pressure field
Validate with untrained experimental data
Qualitative & quantitative reconstruction of flow field?
Yes
No
Expand to 3D simulation
Qualitative reconstruction of surface shear topology?
Yes
No
Limitation due to training data?
Yes
No
Perform ablation study
Conclude and discuss future work
Publish SciTech paper
Investigate alternative ML methods (ODIL)
Compare ML methods to conventional CFD
Key
Function
Data
Decision
Completed
In progress
Future

Figure A.1: Flowchart depicting work/data flow contextualized within past, current and future progress.

B

Learning Decay Rate Scheduler

Manually defined exponential decay schedule of learning rate:

```python
def lr_lambda(epoch: int) -> float:
    # Learning rate decay
    if epoch < 5000:
        return 1.0  # Keep initial learning rate for first 5000 epochs
    elif epoch < 10000:
        return 0.5  # Reduce the learning rate by half after 7500 epochs
    elif epoch < 20000:
        return 0.1
    elif epoch < 30000:
        return 0.05
    elif epoch < 40000:
        return 0.01
    elif epoch < 65000:
        return 0.005
    else:
        return 0.0025  # Further reduce the learning rate after 10000 epochs
```

Similarly, a continuous exponential decay rate scheduler was also investigated, which yielded similar results. Nevertheless, since the manual scheduler had already been established and performed slightly better, it was used over the simpler alternative, shown below:

```python
def lr_lambda(epoch):
    decay_steps = 2000
    decay_rate = 0.9
    # Apply exponential decay
    return decay_rate ** (epoch / decay_steps)
```

C

PINN velocity predictions

Here the remaining velocity predictions by the PINN are displayed, for each plane not shown previously. Note again, that the final plane ($x/L = 0.361$) was discarded due to its lack of coherence to the streamwise PIV data.

C.1. Streamwise velocity, u:

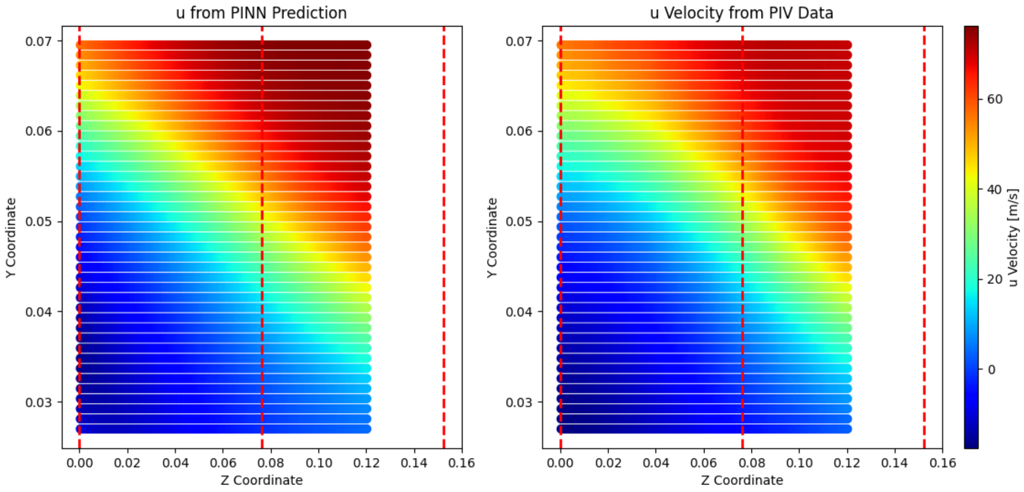


Figure C.1: Side-by-side comparison of predicted (left) and experimental (right) streamwise velocity flow field at distance $x/L = 0.208$ post-bump apex.

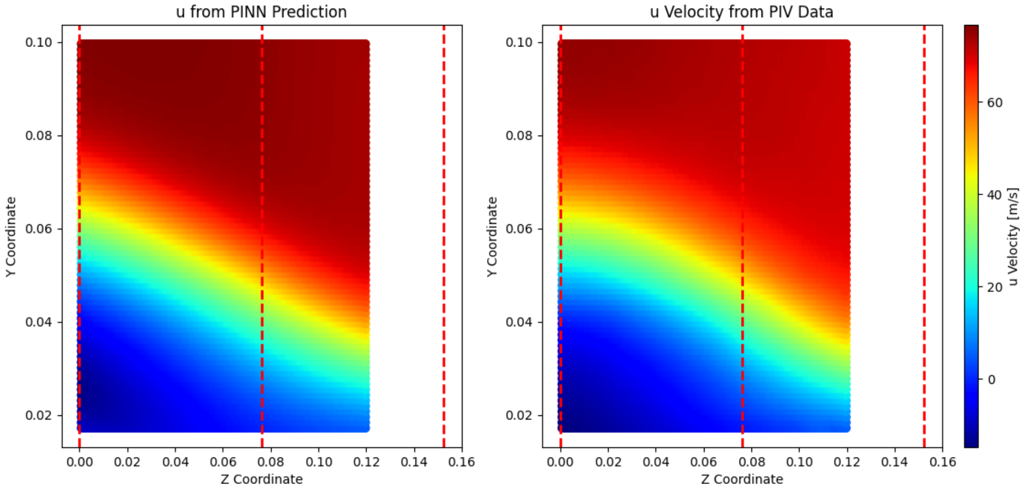


Figure C.2: Side-by-side comparison of predicted (left) and experimental (right) streamwise velocity flow field at distance $x/L = 0.250$ post-bump apex.

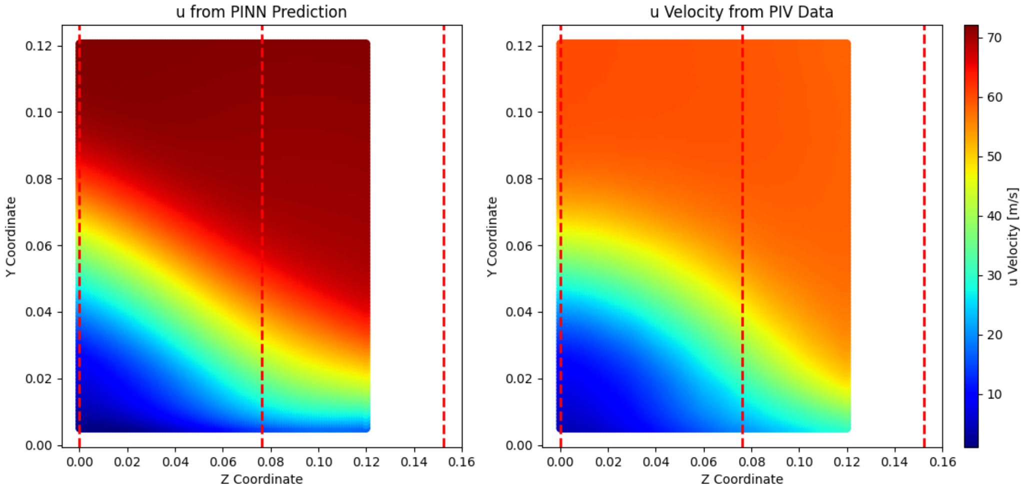


Figure C.3: Side-by-side comparison of predicted (left) and experimental (right) streamwise velocity flow field at distance $x/L = 0.361$ post-bump apex.

C.2. Streamwise velocity, v:

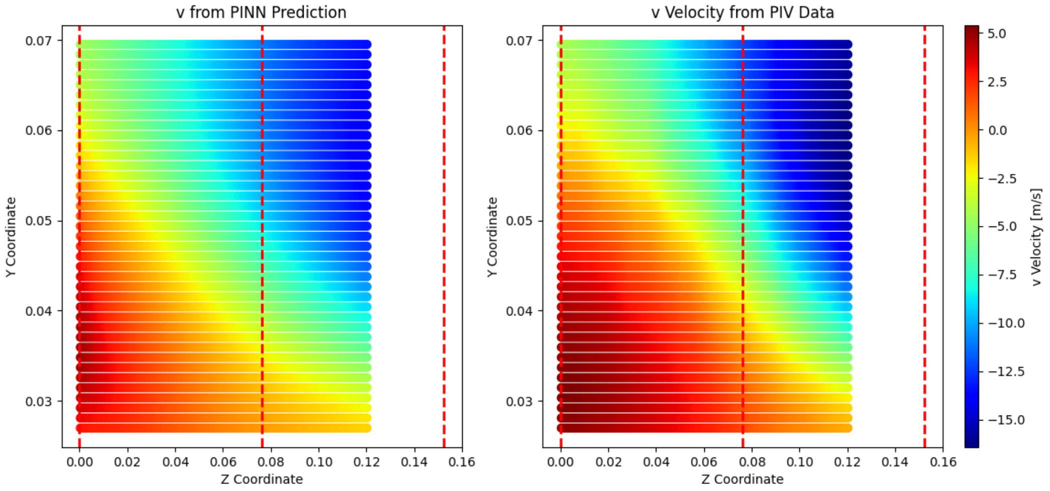


Figure C.4: Side-by-side comparison of predicted (left) and experimental (right) vertical velocity flow field at distance $x/L = 0.208$ post-bump apex.

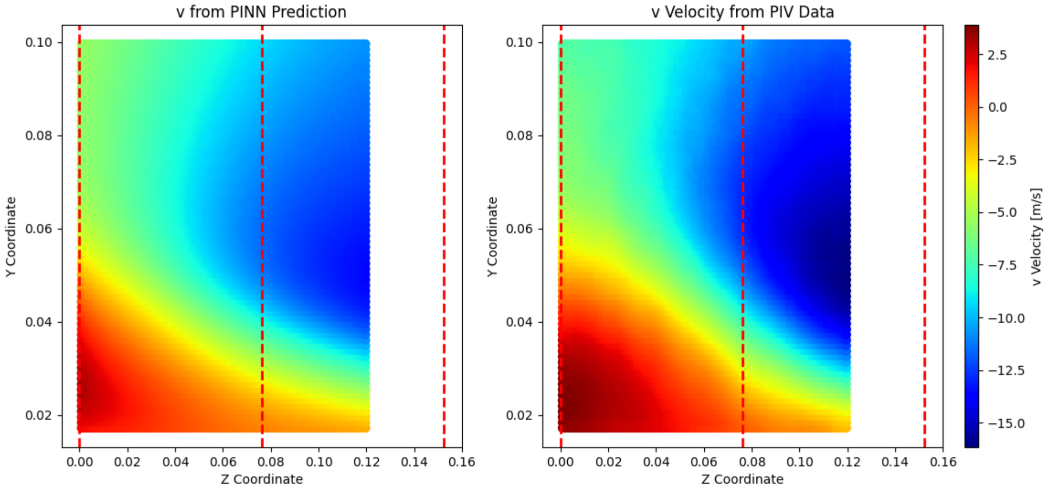


Figure C.5: Side-by-side comparison of predicted (left) and experimental (right) vertical velocity flow field at distance $x/L = 0.250$ post-bump apex.

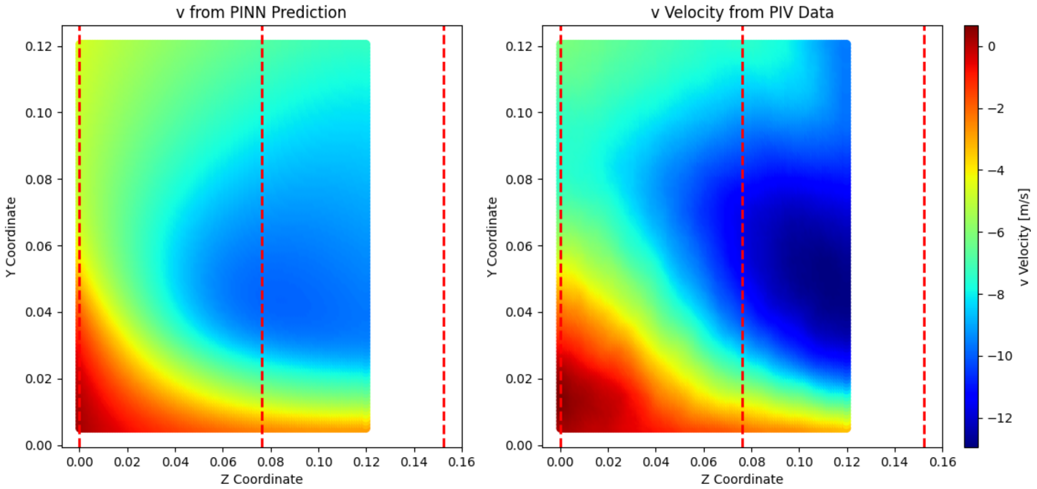


Figure C.6: Side-by-side comparison of predicted (left) and experimental (right) vertical velocity flow field at distance $x/L = 0.361$ post-bump apex.

C.3. Streamwise velocity, w:

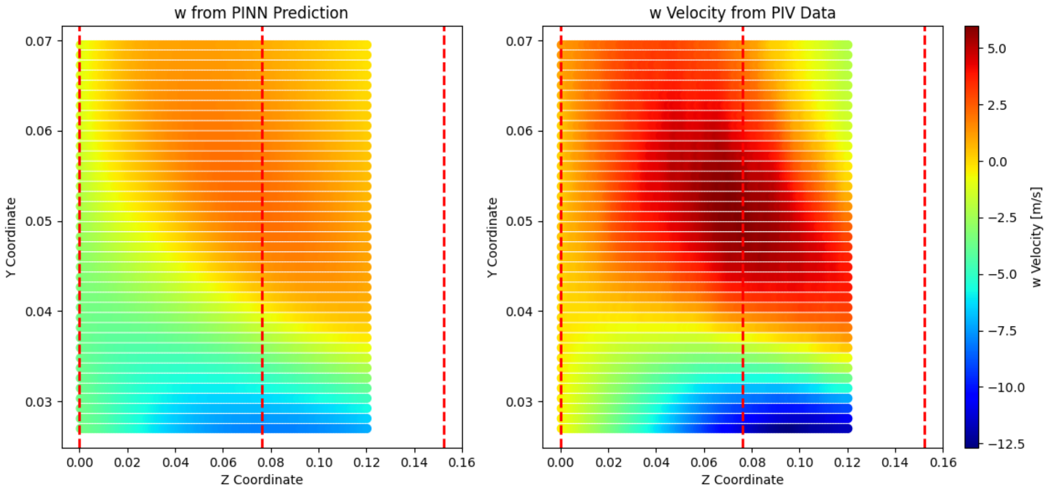


Figure C.7: Side-by-side comparison of predicted (left) and experimental (right) spanwise velocity flow field at distance $x/L = 0.208$ post-bump apex.

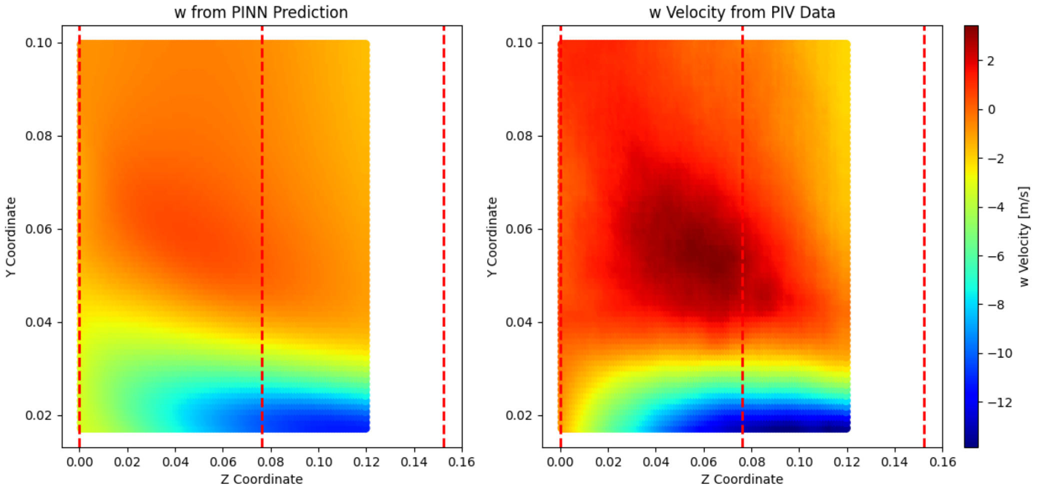


Figure C.8: Side-by-side comparison of predicted (left) and experimental (right) spanwise velocity flow field at distance $x/L = 0.250$ post-bump apex.

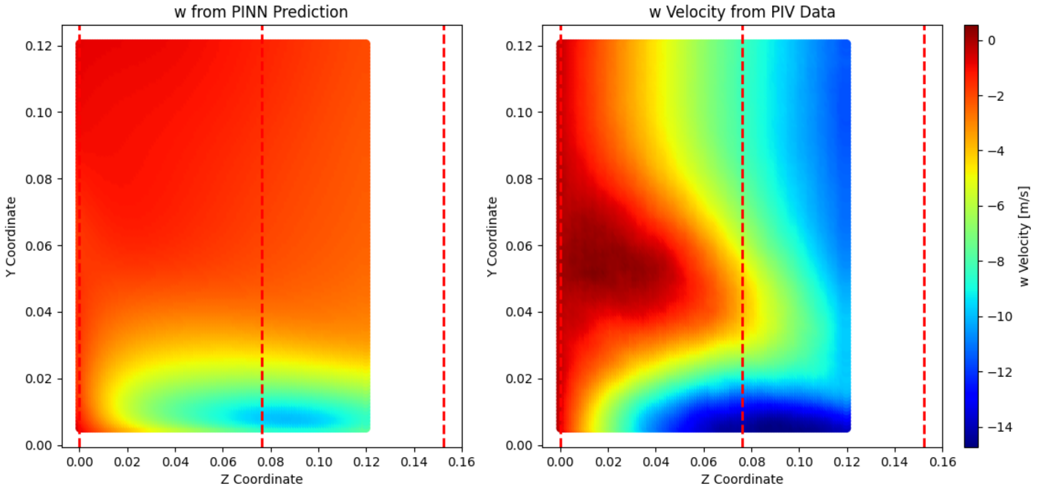


Figure C.9: Side-by-side comparison of predicted (left) and experimental (right) spanwise velocity flow field at distance $x/L = 0.361$ post-bump apex.