Flow Field and Shape Inference in Magnetic Resonance Velocimetry Using Physics-Informed Neural Networks

Rami Cassia

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Downing College August 2021

Declaration

I, Rami Cassia of Downing College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

The software used in this thesis was written from scratch in Python, with predominant use of the JAX library. Ushnish Sengupta kindly provided me a copy of the code he was using as a starting point, and it is this code that I extensively refactored and modified to produce the research described in this thesis. Chapters 3, 4, and 5 show results produced using my code.

Code can be found at https://github.com/RamiCassia/MPhil-MLMI-Project.

This report contains no more than 13064 words.

Rami Cassia
August 2021

Abstract

Magnetic Resonance Velocimetry (MRV) is a non-invasive experimental technique widely used in biomedical engineering to measure the velocity field of a fluid. Whilst these measurements are dense, they are also noisy, having a low Signal-to-Noise Ratio (SNR). A common method of de-noising MRV fields is by scanning thousands of MRV images and averaging them. However, this method is impractical because it requires considerable time and energy to make the numerous scans. There have also been previous studies on MRV de-noising using a single image, and such methods have usually required the shape of the flow boundary to be known *a-priori*. This however requires a set of additional measurements, which can be expensive to obtain. Alternatively, and in principle, an MRV image can be de-noised via smoothing methods that simultaneously impose physical constraints on the flow. The physical constraints would be encapsulated in governing equations for mass and momentum in the case of fluid dynamics. In this research project, we attempt to implement this latter-most approach using a Physics-Informed Neural Network (PINN). The PINN uses the noisy MRV data alone to de-noise the velocity field, and thereafter flow shape is inferred by plotting streamlines. The de-noising is achieved by training the neural network with a composite loss function that incorporates the residuals of the Partial Differential Equations (PDEs) of fluid dynamics. The advantage of this method lies in the fact that it a) incorporates physical knowledge of flow, b) requires a single MRV scan, c) does not require the shape of the flow to be known *a-priori*, and d) is simple and flexible. The flexibility is demonstrated in this project as the de-noising method is applied to flows ranging in geometric and physical complexities. In particular, the de-noising algorithm is tested by assimilating both synthetic and experimental MRV measurements for flows that can be modelled by the Stokes and Navier-Stokes equations, for steady and unsteady flow, and in 2D or 3D. We find that we are able to reconstruct very noisy MRV signals, and recover the ground truth flow field and flow shape with reasonably high reconstruction accuracies.

Acknowledgements

I would like to thank my supervisor, Professor Matthew Juniper, for his constant support and guidance throughout this MPhil project. I want to also thank co-supervisor Ushnish Sengupta, and Alexandros Kontogiannis, for their continuous help in these last six months.

Dr Andrew Sederman and Scott Elgersma, from the Department of Chemical Engineering and Biotechnology, kindly provided experimental MRV data for use in this project, as well as details of the MRV experimental procedure, and so I extend my gratitude to them as well.

Table of contents

List of figures

List of tables

Nomenclature

Greek Symbols

ρ	Density
μ	Dynamic Viscosity
Ω	Domain
ζ	Flow Shape
ν	Kinematic Viscosity
δ	Noise
η	Non-adaptive Learning Rate
σ	Standard Deviation
ω	Vorticity
θ	Weight Matrix

Other Symbols

l	Characteristic Length
$\mathscr{L}$	Loss
p	Pressure
RES	Residual
Re	Reynolds Number
t	Time

u	Velocity
x	x-position
y	y-position
z	z-position

Acronyms / Abbreviations

BC	Boundary Condition
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
GPU	Graphics Processing Unit
IC	Initial Condition
MRI	Magnetic Resonance Imaging
MR	Magnetic Resonance
MRV	Magnetic Resonance Velocimetry
MSE	Mean-Squared Error
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
RF	Radio Frequency
SNR	Signal-to-Noise Ratio

Chapter 1

Introduction

In this chapter, a general overview of the research project is outlined along with the main contributions, followed by a description of the structure of this report.

1.1 Magnetic Resonance Velocimetry

Magnetic Resonance Imaging (MRI) and Magnetic Resonance Velocimetry (MRV)[1] are non-invasive imaging techniques that are used in biomedical engineering and in health assessment [Fukushima, 1999; van de Meent et al., 2009; Blythe et al., 2015; Gladden and Sederman, 2017]. MRV specifically is used for investigating irregularities of the cardiovascular system, such as the weakening of blood vessel walls and the resulting enlargement of the blood vessel (i.e. an aneurysm), as well as the narrowing and blockage of blood vessels (i.e. stenoses). MRV also provides a non-invasive way of obtaining measurements of unsteady velocity fields for complex, 3D flow geometries [Markl et al., 2012]. Additionally, Magnetic Resonance (MR) techniques are beneficial because they do not involve the use of harmful ionising radiation, unlike X-ray tomography.

However, a disadvantage of MR techniques is that a single MRI or MRV scan yields a noisy image[2]. The low Signal-to-Noise Ratio (SNR) results from inhomogeneities of the magnetic field, thermal noise within Radio Frequency (RF) coils, and non-linearity of the

[1] MRV can be considered a subset of MRI. In MRV, each point in an image or scan is a velocity value, rather than a pixel value.

[2] The terms 'scan', 'image', and 'field' are used interchangeably throughout this report, and refer to a domain of MRV values.

signal amplifier [Weishaupt et al., 2003]. Other factors include the procedure of image processing and patient-specific factors such as movements during scanning [Weishaupt et al., 2003]. The consequence of low SNR scans is the requirement to average over thousands of them to acquire a high SNR image, and as a result the acquisition time is long. One solution to this problem is to use MR protocols[3] with smaller acquisition times, but these protocols may cause MRV images to have irregular artefacts.

1.2 Previous Research

Research has therefore been invested in ways of achieving a high SNR image at a reduced acquisition time, by only using a single scan. The resulting reduction in acquisition time and number of scans is important as this implies a reduction in the operating and maintenance costs of the MR machine. The time reduction would also be beneficial to claustrophobic patients.

An example of research into improving MRV acquisition time involves utilising sparse sampling patterns to reconstruct signals [Lustig et al., 2007]. This approach is known as compressed sensing, and is based on the idea that knowledge of signal sparsity allows the signal to be de-noised using fewer samples than the Nyquist-Shannon theorem requires [Donoho, 2006]. More recently, use of deep variational neural networks for MRV reconstruction has shown substantial results [Hammernik et al., 2018; Vishnevskiy et al., 2020]. However, these approaches described do not account for physical knowledge of the systems they are reconstructing.

Since each point of an MRV image is a physical quantity (velocity), it is logical to incorporate physical knowledge into the reconstruction technique to potentially achieve more accurate reconstructions. One way involves utilising partial differential equations (PDEs) of fluid dynamics by formulating an inverse flow problem, whereby the modelled flow is matched to the measured flow by iteratively updating model parameters. In addition to de-noising MRV images, this method can be used to infer extra parameters such as pressure and viscosity. Reconstruction of time-varying 3D velocity fields in blood vessels has also been demonstrated to be possible using adjoint-based methods of solving PDEs, and this involves determining the inlet velocity Boundary Condition (BC) of the flow [Funke et al., 2019; Koltukluoğlu, 2019; Koltukluoğlu and Blanco, 2018].

[3] A MR protocol is a combination of various MR sequences. A sequence is a number of RF pulses and gradients that result in a set of images with a particular appearance.

1.3 This Project and its Contributions

Recently, Physics Informed Neural Networks (PINNs) have also been used to solve inverse problems in physics [Raissi et al., 2019]. The PINNs essentially incorporate physical aspects of the system into a neural network to leverage its capabilities. The incorporation of physics is in altering how the loss function is computed. This project takes inspiration from this PINN approach, by using them specifically for the purpose of MRV de-noising. We research using PINNs to reconstruct highly noisy MRV images in a simple and flexible way. The idea is to use the noisy MRV image as the reference during neural network training, so that the network smoothens the noisy image. The physics is incorporated by adding fluid dynamic PDE residuals into the loss function with some residual weighting. Having reconstructed the flow, the shape is inferred by plotting streamlines close to the flow boundary. The streamline would depict the trajectory a fluid particle takes close to the actual flow boundary, and is therefore a suitable representation of this boundary.

The main contributions of this research idea are:

- **Requirement of only a single noisy MRV image**: This eliminates the operating time and costs of taking thousands of scans and averaging them to reduce noise.
- **No requirement of prior knowledge of geometry**: Methods outlined in the previous section have assumed that the geometry of the system is known *a-priori*. The inclusion of prior knowledge of geometry in these methods introduces an added layer of complication for reconstruction. The PINN method used in this project does not require flow shape to be known beforehand, making it a simpler approach to apply to a range of different flow complexities. Rather, our method actually infers geometry after reconstruction. Our method is therefore practical in real applications because MR scans are often needed in order to find the geometry of a blood vessel, for example.
- **Scalable to more complicated flows**: Despite the demonstrated possibility, adjoint-based methods outlined in the previous section are harder to implement in complicated flows such as 3D unsteady flows, as the adjoints themselves are harder to code. The PINN approach is more easily scalable to more complicated flows, as it involves simply increasing dimensionality of input space, by adding an extra z-position and/or time variable, for example.
- **Physics-embedded machine learning approach to MRV de-noising**: The addition of residuals of fluid dynamic PDEs to the neural network data loss should in principle

drive the reconstruction towards a closer match to the ground truth velocity field, resulting in higher reconstruction accuracies.

1.4 Outline of Thesis

Following this introduction chapter, Chapter 2 provides the background of this research project. In particular, we detail the architecture and training of PINNs. Attention is also placed on the derivation of the composite loss function, including the form of the fluid dynamic PDEs used for residual computation. These PDEs are collectively known as the Navier-Stokes equations and can take several forms depending on geometrical and physical complexity, and so are briefly discussed. We also outline the nature of the data used for PINN training, and how synthetic and experimental data are generated. The metrics that are used to determine the effectiveness of our method are then summarised.

Thereafter, Chapters 3, 4 and 5 will present, for different flow scenarios, results generated by applying the methods and metrics outlined in Chapter 2. Chapter 3 explores reconstruction of 3D steady flow through a blood vessel with an aneurysm. We first assume the flow is purely laminar, and then account for inertial forces. The data used here is synthetic.

Chapter 4 explores reconstruction of 3D steady flow through a convergent nozzle. Unlike the previous chapter, the data used here is experimental and provided by the Department of Chemical Engineering and Biotechnology at Cambridge University. This chapter thus serves as an opportunity to test our method in a real-life scenario.

Chapter 5 explores the reconstruction of flow past a cylinder. Flow reconstruction in this chapter is simpler than in the previous chapters in the sense that the flow being examined here is in 2D, but more complicated in the sense that the flow is also unsteady. Once again, the data used in this chapter is synthetic.

Finally, the thesis ends with conclusions in Chapter 6, where improvements to the PINN approach are also suggested.

Chapter 2

Background

The purpose of this chapter is to provide the background of the research project. As described in Chapter 1, the idea is to use a PINN to reconstruct a noisy MRV image into a noiseless one. Therefore, Chapter 2 begins by describing the PINN architecture. Thereafter, focus is turned on the composite loss function that is to incorporate physical knowledge of the fluid flow - this stage involves describing the Navier-Stokes equations, and manipulating them into a form suitable for use in calculating the loss. Afterwards, practical aspects of PINN training, such as the optimisation algorithm to be used, are outlined. Then, the method of obtaining experimental and synthetic MRV data is discussed. The chapter ends by introducing success metrics for evaluating the effectiveness of the PINN method.

2.1 Overview

2.1.1 PINN Reconstructions: Principle and Architecture

In the last two decades deep neural networks have been used effectively in the fields of computer vision and natural language processing [LeCun et al., 2015]. Despite success in these areas, deep learning was not being widely used in scientific computing. Recently, however, methods have been developed to solve PDEs using deep learning, leading to a new field known as scientific machine learning [Baker et al., 2019]. Through scientific machine learning, promise has been demonstrated in replacing traditional numerical discretisation methods with a neural network that approximates the solution to PDEs [Lu et al., 2021].

The way to approximate the solution of PDEs using a neural network is to use automatic differentiation on the network output to compute PDE residuals, and then constrain the neural network to minimise these PDE residuals. This method does not involve computational meshing of the physical domain, which is beneficial because mesh generation can be time-consuming, technically difficult, and prone to errors such as creating cells with negative volume or cells that do not join together. The deep learning approach is therefore an advantage over mesh-based Finite Difference Methods (FDMs), Finite Volume Methods (FVMs), and Finite Element Methods (FEMs) in this regard [Lu et al., 2021].

Figure 2.1 depicts the use of a PINN to solve the diffusion equation [Lu et al., 2021]. The figure shows how the output of the neural network is modified in order to compute PDE residuals as well as residuals of the Initial Conditions (ICs) and Boundary Conditions (BCs). The residuals are used to calculate a physical loss, and optimising this physical loss solves the diffusion equation.

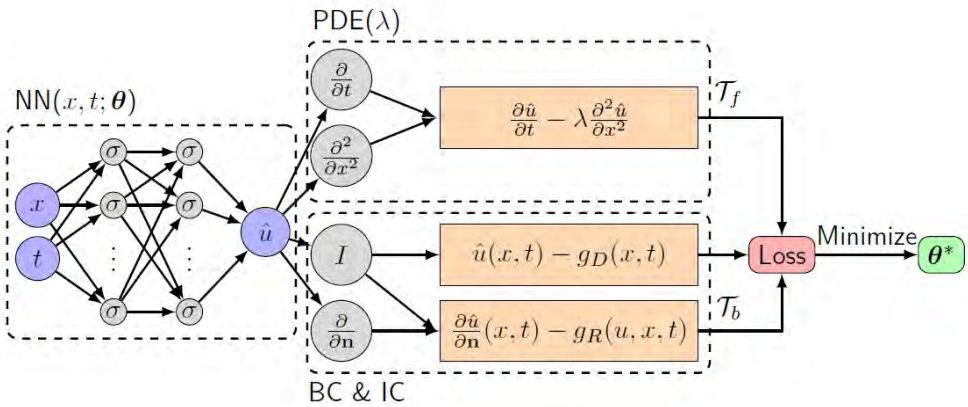

Fig. 2.1 Use of a PINN to solve the diffusion equation [Lu et al., 2021]. Automatic-differentiation is applied on the output of the network, in order to compute a physical loss that comprises the residuals of the PDEs, ICs and BCs.

This research project essentially adopts the PINN approach for de-noising MRV images, where the PDEs of interest are those of fluid dynamics (Navier-Stokes equations). Unlike the approach depicted in Figure 2.1, we present slight modifications to the PINN algorithm, namely:

- We ignore the IC and BC residuals of our flows- these are assumed unknown *a-priori*.

- The loss function is a composite one that takes into account both the neural network data loss and the PDE residual loss, weighted against each other using a residual weight.

Thus, the generic structure of the PINN is as follows:

- The PINN is fully connected, with several hidden layers and a constant number of nodes per hidden layer.
- The input to the PINN is a point cloud of spatio-temporal co-ordinates, $\mathbf{x} = (x,\ y,\ z,\ t)$.
- The output of the PINN is the corresponding x-velocity, y-velocity, and z-velocity, i.e $\mathbf{u} = (u_x,\ u_y,\ u_z)$, at each point of the input point cloud.
- The loss function of the PINN is given by:

$$\mathscr{L} = (1-r)\cdot\mathscr{L}_{Data} + r\cdot\mathscr{L}_{Residual} \tag{2.1}$$

  Where $\mathscr{L}$ is the total loss, $\mathscr{L}_{Data}$ is the neural network data loss, $\mathscr{L}_{Residual}$ is the PDE residual loss, and r is the residual weight.

- The label used for calculating $\mathscr{L}_{Data}$ is the noisy MRV image that is to be de-noised.

Figure 2.2 depicts the PINN principle used in this project diagrammatically.

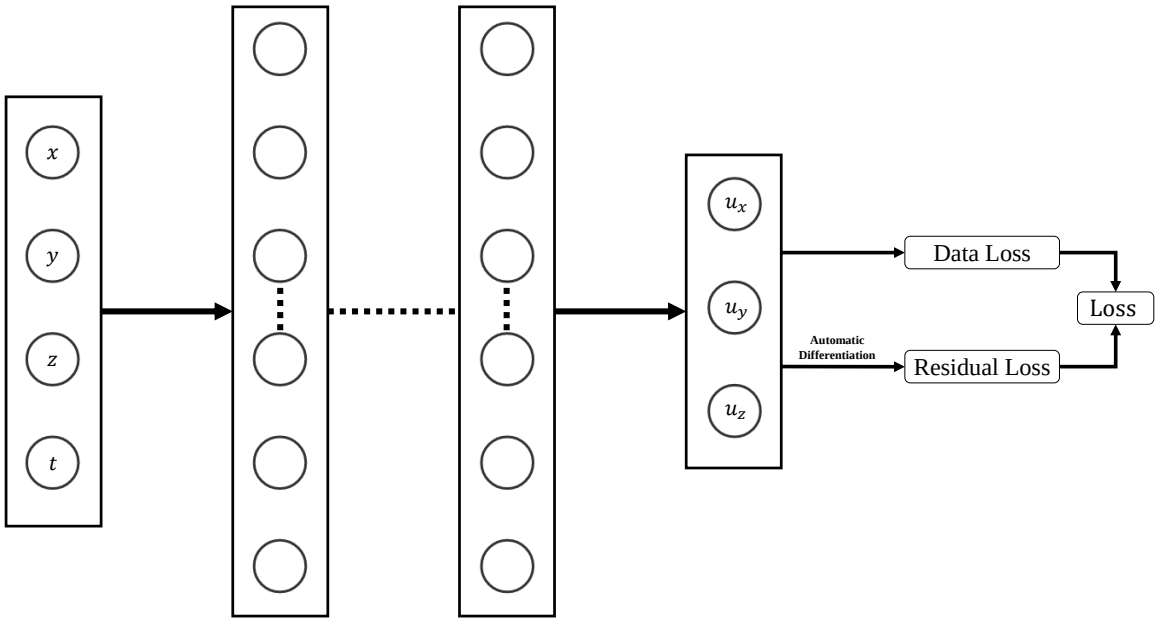

Fig. 2.2 The fully connected PINN architecture that is adopted by the project. The PINN takes input $(x,\ y,\ z,\ t)$ and outputs $(u_x,\ u_y,\ u_z)$. The output is used to calculate PDE residuals via automatic differentiation, and this residual loss is combined with the data loss (via some residual weight r) to yield the total, composite loss. De-noising is achieved by minimising the composite loss.

Therefore, by training a PINN to minimise the composite loss, we obtain a converged network weight matrix, θ, that is used to evaluate the de-noised MRV field:

$$\mathbf{u}_\Omega = F(\mathbf{x}_\Omega;\ \theta) \tag{2.2}$$

where the field or domain is denoted by Ω.

As discussed in Chapter 1, the advantage of this smoothing method lies in that it requires a single noisy MRV image, does not require prior knowledge of flow geometry, and embeds physical knowledge to achieve more accurate flow field reconstructions (and therefore more accurate shape inferences).

2.1.2 Flow Shape Inference

Having reconstructed the flow field, inferring flow shape is done by subsequently making use of the reconstructed flow field. One simple but effective method is to plot streamlines of the flow. We select initial points at the flow inlet and examine how particles at these points traverse the velocity field[1]. Assuming steady flow, then particle position at time $t+1$, i.e. $\mathbf{x}_{t+1} = (x,\ y,\ z)_{t+1}$, is given by:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \int_t^{t+1} \mathbf{u}\ dt \tag{2.3}$$

Thus, for each initialised point, a streamline is obtained. The position of initial points at the inlet are selected such that they approximately represent points at the boundary - i.e. these points should be where velocity magnitude is transitioning from high to low. In this way, streamlines that reasonably describe the shape of 2D or 3D flows are obtained.

The formulation of the loss function $\mathscr{L}$, and the training method of the PINN, are discussed in Sections 2.2 and 2.3 respectively.

2.2 Composite Loss Function

In Section 2.1, the loss function is introduced:

$$\mathscr{L} = (1-r)\cdot\mathscr{L}_{Data} + r\cdot\mathscr{L}_{Residual} \tag{2.4}$$

In this section, we formulate $\mathscr{L}_{Data}$ and $\mathscr{L}_{Residual}$.

2.2.1 Form of Data Loss

The de-noising method from Section 2.1 is essentially a regression task, and the typical data loss function that neural networks use in regression is the Mean-Squared Error (MSE), given by:

[1] Technically, this process describes how streaklines are created. However, streaklines and streamlines are identical in steady flow.

$$MSE = \frac{1}{N} \cdot \sum_{n=1}^{N} \left[(u_{x,n} - \hat{u}_{x,n})^2 + (u_{y,n} - \hat{u}_{y,n})^2 + (u_{z,n} - \hat{u}_{z,n})^2 \right] \tag{2.5}$$

where $u_{x,n}$, $u_{y,n}$ and $u_{z,n}$ are neural network predictions for the x-velocity, y-velocity and z-velocity at a particular position and time in the flow, and $\hat{u}_{x,n}$, $\hat{u}_{y,n}$ and $\hat{u}_{z,n}$ are the corresponding references (i.e. corresponding points in the noisy MRV image). The evaluation is done over N samples selected at random from the flow field.

A problem with Equation 2.5 is that it does not weigh the losses in the x, y, and z dimensions relative to each other. This could potentially reduce de-noising performance, since the extent of noise in the raw MRV image is different for velocities in the x, y, and z directions. Assuming Gaussian noise, a solution to this weighting problem is to weigh the squared-difference in each direction by the inverse noise variance in the respective direction. The form of the data loss $\mathscr{L}_{Data}$ is then:

$$\mathscr{L}_{Data} = \frac{1}{N} \cdot \sum_{n=1}^{N} \left[\frac{1}{\sigma_x^2} \cdot (u_{x,n} - \hat{u}_{x,n})^2 + \frac{1}{\sigma_y^2} \cdot (u_{y,n} - \hat{u}_{y,n})^2 + \frac{1}{\sigma_z^2} \cdot (u_{z,n} - \hat{u}_{z,n})^2 \right] \tag{2.6}$$

where σ_x^2, σ_y^2, and σ_z^2 are the noise variances in the x, y, and z directions. The relation of the noise variances to SNR is discussed in Section 2.4.

2.2.2 Form of Residual Loss

The residual loss $\mathscr{L}_{Residual}$ quantifies the residuals of physical PDEs. In the case of MRV de-noising, the PDEs of concern are those of fluid dynamics, known as the Navier-Stokes equations. Thus, in order to formulate $\mathscr{L}_{Residual}$, the Navier-Stokes equations are described first.

In the full-form, the Navier-Stokes equations describe how velocity, pressure, temperature, and density of a moving fluid are related. They are a set of coupled differential equations, meaning the equations must be solved simultaneously to determine the flow field given ICs and BCs [Batchelor and Batchelor, 2000].

One of the equations in the full Navier-Stokes is the continuity equation, which describes conservation of mass. There are an additional three momentum equations describing con-

servation of momentum, as well as one equation describing conservation of energy. The equations cover four independent variables (x, y, z, t) and six dependent variables (density, temperature, pressure, and three velocity components) [Anderson and Wendt, 1995].

In addition, for the momentum equations, there are different types of terms: convection, diffusion, and additional source terms [Ferziger et al., 2002]. Convection terms are responsible for transport of fluid momentum via ordered motion of flow. Diffusion terms are responsible for momentum transport via molecular mixing, i.e. via random motion of fluid molecules. Additional source terms refer to pressure gradients and body forces, for example.

If the momentum equations are dominated by convective and pressure terms, the resulting flow tends to turbulent flow. On the opposite extreme, if the momentum equations are dominated by diffusion terms, the resulting flow tends to laminar viscous flow [Batchelor and Batchelor, 2000]. The balance between these competing terms is quantified by the Reynolds Number Re, which is simply the ratio of inertial forces to viscous forces:

$$Re = \frac{\rho \cdot u \cdot l}{\mu} = \frac{u \cdot l}{\nu} \tag{2.7}$$

Where l is the characteristic length of the flow, μ is the kinematic viscosity, ν is the dynamic viscosity, and ρ is the density ($\nu = \mu/\rho$).

Moreover, extra assumptions about the nature of the flow can simplify the Navier-Stokes. For example, in all the flow cases in this project it is assumed that the flow is incompressible, isothermal, and of uniform viscosity. This removes the energy equation and reduces the Navier-Stokes to the following set of equations [Brennen, 2006]:

$$\nabla \cdot \mathbf{u} = 0 \tag{2.8}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \left(\frac{|\mathbf{u}|^2}{2} \right) - \mathbf{u} \times \omega = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} \tag{2.9}$$

Equation 2.8 is the continuity equation and states that divergence of the velocity field has to equal zero. Equation 2.9 is the momentum equation.

Equation 2.8 only makes use of velocity. Since MRV images only comprise velocity values, this means that Equation 2.8 is already in a suitable format for calculating the continuity

residual. On the other hand, Equation 2.9 involves terms other than velocity, namely vorticity ω and pressure p. Vorticity is the curl of velocity, i.e. $\omega = \nabla \times \mathbf{u}$. Since vorticity is derived from velocity, the presence of vorticity in the momentum equation is not a problem. However, Equation 2.9 needs to be manipulated into a form that excludes pressure, since this quantity is assumed unknown in all the encountered reconstruction problems.

Noting that $\nabla \cdot \mathbf{u} = 0$ and $\nabla \cdot \omega = 0$, we reformulate Equation 2.9 by making use of two identities [Brennen, 2006]:

$$\nabla^2 \omega = \nabla(\nabla \cdot \omega) - \nabla \times (\nabla \times \omega) = -\nabla \times (\nabla \times \omega) \tag{2.10}$$

$$\nabla \times (\mathbf{u} \times \omega) = (\omega \cdot \nabla)\mathbf{u} - \omega(\nabla \cdot \mathbf{u}) - (\mathbf{u} \cdot \nabla)\omega + \mathbf{u}(\nabla \cdot \omega) = (\omega \cdot \nabla)\mathbf{u} - (\mathbf{u} \cdot \nabla)\omega \tag{2.11}$$

Using Identities 2.10 and 2.11, the curl of Equation 2.9 is expressed in a form that excludes the pressure term [Brennen, 2006]:

$$\frac{\partial \omega}{\partial t} + (\mathbf{u} \cdot \nabla)\omega - (\omega \cdot \nabla)\mathbf{u} - \nu \nabla^2 \omega = \mathbf{0} \tag{2.12}$$

Both Equations 2.8 and 2.12 are now in a suitable format for computing the total PDE residual. In index notation, this total PDE residual at a particular point q in the flow domain is written as:

$$RES_q = \sum_{i=1}^{3} \left[\frac{\partial u_i}{\partial x_i} \right]^2 + \sum_{i=1}^{3} \left[\frac{\partial \omega_i}{\partial t} + \sum_{j=1}^{3} \left(u_j \cdot \frac{\partial u_i}{\partial x_j} - \omega_j \cdot \frac{\partial u_i}{\partial x_j} - \nu \cdot \frac{\partial^2 \omega_i}{\partial x_j \partial x_j} \right) \right]^2 \tag{2.13}$$

where indices $1,\ 2,\ 3$ represent spatial directions $x,\ y,\ z$.

Finally, the form of $\mathscr{L}_{Residual}$ is written as:

$$\mathscr{L}_{Residual} = \frac{1}{Q} \cdot \sum_{q=1}^{Q} min \left(RES_q,\ \left| \mathbf{u}_q \right|^2 \right) \tag{2.14}$$

where the residual is calculated at Q random points in the spatio-temporal flow domain. The idea behind setting $\mathscr{L}_{Residual}$ based on the minimum between RES_q and $\left|\mathbf{u}_q\right|^2$ is to help the training algorithm distinguish between points that are inside and outside of the flow boundary.

2.2.3 Form of Composite Loss

Finally, we summarise the previous two subsections to formulate $\mathscr{L}$:

$$\mathscr{L} = (1-r)\cdot\mathscr{L}_{Data} + r\cdot\mathscr{L}_{Residual} \tag{2.15}$$

$$\mathscr{L}_{Data} = \frac{1}{N}\cdot\sum_{n=1}^{N}\left[\frac{1}{\sigma_x^2}\cdot(u_{x,n}-\hat{u}_{x,n})^2 + \frac{1}{\sigma_y^2}\cdot(u_{y,n}-\hat{u}_{y,n})^2 + \frac{1}{\sigma_z^2}\cdot(u_{z,n}-\hat{u}_{z,n})^2\right] \tag{2.16}$$

$$\mathscr{L}_{Residual} = \frac{1}{Q}\cdot\sum_{q=1}^{Q} min\left(RES_q,\ \left|\mathbf{u}_q\right|^2\right) \tag{2.17}$$

$$RES_q = \sum_{i=1}^{3}\left[\frac{\partial u_i}{\partial x_i}\right]^2 + \sum_{i=1}^{3}\left[\frac{\partial \omega_i}{\partial t} + \sum_{j=1}^{3}\left(u_j\cdot\frac{\partial u_i}{\partial x_j} - \omega_j\cdot\frac{\partial u_i}{\partial x_j} - \nu\cdot\frac{\partial^2\omega_i}{\partial x_j\partial x_j}\right)\right]^2 \tag{2.18}$$

2.3 Training of PINN

An important aspect of MRV de-noising via PINNs is the training of the PINN itself. In this section, we consider in detail the practicalities of PINN training.

2.3.1 Optimisation Algorithm

Central to PINN training is the algorithm that minimises the loss function derived in Section 2.2. The usual optimisation algorithm of choice is Adagrad or Adam, which are considered state-of-the-art in numerous machine learning applications [Duchi et al., 2011; Kingma

and Ba, 2014]. These methods adaptively tune the learning rate for each parameter during optimisation by making use of second-order statistics [Anil et al., 2019]. Such methods not only offer superior convergence characteristics, but are desirable in large applications due to their moderate space and time requirements. This is because the space requirements of Adagrad and Adam are linear in the number of training parameters [Anil et al., 2019]. Despite this, modest memory overheads can still heavily limit the quality of very large trained models, and, as a result, the optimiser overhead tends to heavily restrict model size as well as training mini-batch size [Anil et al., 2019]. In dense MRV images there are millions of data points, implying that large PINN models are required for de-noising. This means that space and time usage should be as efficient as possible, and Adagrad and Adam may therefore not be optimal choices.

To counter this memory issue, optimisation of the loss function for all de-noising cases is carried out using a memory-efficient optimiser known as SM3 [Anil et al., 2019]. Algorithm 1 outlines the training procedure via SM3 [Anil et al., 2019]. As a brief explanation of the algorithm: at each training epoch (iteration) t, the loss function $\mathscr{L}_t(\theta_t)$ is determined according to Equation 2.15 via a forward pass through the PINN, using a randomly selected sample of points (position and time) in the flow domain. Then, a backward pass through the PINN is carried out to compute the gradient of $\mathscr{L}_t(\theta_t)$ with respect to θ_t. Thereafter, for each set S_r, a running sum $\varepsilon_t(r)$ of the maximal variance over all gradient entries $j \in S_r$ is maintained. Following this, for each parameter i, the minimum over all variables $\varepsilon_t(r)$ associated with sets $S_r \ni i$, is selected. The square-root of this minimum, denoted by $\psi_t(i)$, is used to determine the adaptive learning rate for the i_{th} gradient, i.e. $\eta/\sqrt{\psi_t(i)}$. Finally, parameter update $\theta_{t+1}(i)$ is computed using the loss gradient at the previous step $g_t(i)$ and the parameter at the previous step $\theta_t(i)$.

SM3 tries to retain the benefits of per-parameter adaptability whilst significantly reducing memory overhead [Anil et al., 2019]. The algorithm makes use of a collection of k non-empty sets $\{S_r\}_{r=1}^{k}$ so that $S_r \subseteq [d]$ and $\cup_r S_r = [d]$ [Anil et al., 2019]. Specifically, an index $i \in [d]$ may be contained in multiple sets S_r, and the algorithm maintains only one variable for each set S_r. What this abstraction implies is that, excluding the space required to store θ_t, the SM3

optimiser only requires $O(k)$ additional space, in comparison to the $O(d)$ additional space of standard adaptive optimisers [Anil et al., 2019].

Algorithm 1: PINN training via SM3 optimisation [Anil et al., 2019]

Parameters: Non-adapted learning rate η
Initialise θ_1; $\forall r \in [k] : \varepsilon_0(r) = 0$
for $t = 1 \dots T$ **do**
  Forward Pass to compute $\mathscr{L}_t(\theta_t)$
  Backward Pass to compute $g_t = \nabla \mathscr{L}_t(\theta_t)$
  for $r = 1 \dots k$ **do**
    Set: $\varepsilon_t(r) \longleftarrow \varepsilon_{t-1}(r) + max_{j \in S_r} \left[g_t^2(j) \right]$
  end
  for $i = 1 \dots d$ **do**
    Set: $\psi_t(i) \longleftarrow min_{r:S_r \ni i} \left[\varepsilon_t(r) \right]$
    Update: $\theta_{t+1}(i) \longleftarrow \theta_t(i) - \eta \cdot g_t(i) / \sqrt{\psi_t(i)}$
  end
end

2.3.2 Activation Function

As per Algorithm 1, an important part of training is the forward pass. This involves the input to each layer of the PINN being linearly transformed by multiplying with $\theta^{(l)}$, where l denotes a layer. The output of the linear transformation is then passed into an activation function $\phi(\cdot)$. The output of the final layer of the PINN is then the predicted velocity $\mathbf{u}$, which is used to compute the loss. The gradient of the loss is then computed with respect to the weights of the PINN.

Given the above description of a forward pass, one issue to consider is the choice of the activation function. Numerous possible activation functions $\phi(\cdot)$ are used in the machine learning community. The most common activation function is ReLU [Agarap, 2018]:

$$ReLU(a) = max(0,\ a) \tag{2.19}$$

Another option is the Swish activation function [Ramachandran et al., 2017]:

$$Swish(a) = \frac{a}{1 + e^{-\beta \cdot a}} \tag{2.20}$$

where β is a controllable parameter.

Figure 2.3 graphically illustrates these activation functions.

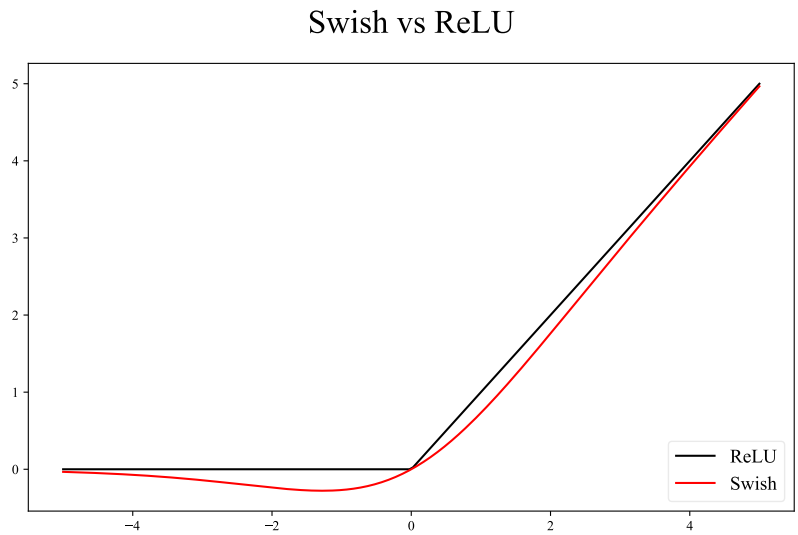

Fig. 2.3 Plots of ReLU and Swish. Swish is plotted with $\beta = 1$.

As per Equation 2.19 and Figure 2.3, ReLU amends negative inputs by forcing them to zero, and the gradient at positive inputs is constant and positive. ReLU is thus bounded from below and unbounded from above. The unboundedness property reduces the problem of slow training (saturation) due to small gradients, a problem sigmoid or tanh functions are notorious for [Agarap, 2018; Ramachandran et al., 2017]. This is primarily why ReLU is a popular choice. ReLU also does not involve exponentials or divisions and so is faster to compute [Nwankpa et al., 2018]. One downside of using ReLU neurons is that they can become trapped in the negative input space, becoming permanently inactive and outputting zero [Nwankpa et al., 2018].

From Figure 2.3, it can also be observed that Swish is bounded from below and unbounded from above, similar to ReLU. In fact, in the limit $\beta \to \infty$, Swish converges to a ReLU-like function [Ramachandran et al., 2017]. The similarity of Swish to ReLU helps avoid the saturation problem [Ramachandran et al., 2017; Glorot and Bengio, 2010a]. However, unlike ReLU, Swish is otherwise smooth and non-monotonic. This smoothness and non-monotonicity produces negative outputs for small negative inputs, unlike ReLU which thresholds all negative inputs to zero. This characteristic of Swish is vital in its success when used on very deep networks. Moreover, experiments conducted on various architectures and datasets indicate

that Swish consistently outperforms ReLU on very deep networks [Ramachandran et al., 2017]. Also, the use of Swish appears to make the neural network more robust to batch size changes, compared to ReLU [Ramachandran et al., 2017]. Additionally, because Swish is smooth, the output landscape of a neural network utilising Swish units is expected to be smoother than the output of a network using ReLU units, as per Figure 2.4 [Ramachandran et al., 2017]. In the flow cases encountered in this project, the desired reconstructed flow fields should generally have smooth variations in velocity, rather than sharp variations, so using Swish is more beneficial for de-noising. The smoothness of the network output also has a direct effect on the smoothness of the loss landscape, and, intuitively, a smoother loss landscape is easier to optimise since it is more traversable [Ramachandran et al., 2017]. This ultimately reduces network sensitivity to hyperparameters such as initialisation and learning rate.

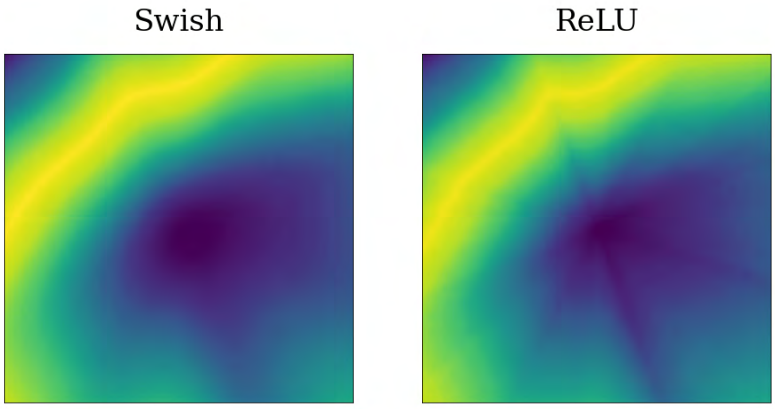

Fig. 2.4 Output landscape of a randomly initialised neural network, using Swish (left) and ReLU (right) activation functions [Ramachandran et al., 2017].

Given all these arguments for Swish, all the PINNs in this project adopt the Swish activation function, for all hidden and output layers.

2.3.3 Network Size

Another issue to consider are the number of layers and number of nodes per layer of the PINN. These must be sufficiently high to capture the full extent of the non-linearities of our problem and prevent underfitting [LeCun et al., 2015], whilst de-noising flow details at smaller flow-scales where this is required. However, a larger PINN architecture imposes

more computational overhead on the de-noising process. The size of the PINN model should be consistent with the data size of the noisy MRV image, as well as the required flow-scale resolution. In Chapters 3 and 5, the effect of number of hidden layers and nodes per layer on reconstruction performance is examined.

2.3.4 Weight Initialisation

Additionally, an important factor to consider is how to initialise the weights of the PINN. A common practice is to initialise the weights of each layer to ensure the variance of the output from a neural layer equals the variance of the inputs to it [Kumar, 2017]. In order for this to happen the following condition must be met:

$$VAR\left[\theta^{(l)}\right] = \frac{1}{n^{(l)}} \tag{2.21}$$

where $n^{(l)}$ is the number of nodes or neurons in layer l. Thus, initialisation for layer l is as follows [Glorot and Bengio, 2010b]:

$$\theta^{(l)} \sim \mathcal{N}\left(\mathbf{0},\, diag\left(1/n^{(l)}\right)\right) \tag{2.22}$$

The above initialisation scheme is known as Xavier Initialisation and helps avoid exploding and vanishing gradients in training [Glorot and Bengio, 2010b; Kumar, 2017]. The downside of this scheme is that it is more optimised for use with the sigmoid activation, and not optimised for ReLU-like functions such as Swish [Kumar, 2017]. A similarly motivated and reasoned initialisation scheme to Xavier's that better accommodates for the Swish function is He Initialisation [He et al., 2015]:

$$VAR\left[\theta^{(l)}\right] = \frac{2}{n^{(l)}} \tag{2.23}$$

$$\theta^{(l)} \sim \mathcal{N}\left(\mathbf{0},\, diag\left(2/n^{(l)}\right)\right) \tag{2.24}$$

All PINN weights in this project are therefore initialised using He Initialisation.

2.3.5 Residual Weight

Consideration must also be given to how the residual weight r in Equation 2.15 is set. This hyperparameter determines the balance between data and physics and therefore has a direct effect on de-noising performance. During training, the residual weight is activated via a step-change from zero to a desired number after the PINN achieves an adequate initial fit to the data.

In theory, there is an optimal setting of the residual weight that optimises reconstruction performance. A hyperparameter study of the residual weight is therefore performed for all reconstruction cases in this project. Additionally, an exponential decay variation of the residual weight from epoch 0 to the final epoch is also examined to determine if this is a more optimal scheme than the step-change approach.

2.3.6 Sampling Sizes for Data and Residual Losses

As explained in Section 2.2, points from the input space are randomly sampled to compute $\mathscr{L}_{Data}$ and $\mathscr{L}_{Residual}$.

For $\mathscr{L}_{Data}$, a large sampling size (or batch size) gives slower convergence but a more accurate loss-gradient estimate. A smaller batch size results in quicker convergence, but a noisier training process. From trial and error, a sampling size of 2048 gives a good compromise between the aforementioned competing factors, across all flow cases encountered.

The sampling size of $\mathscr{L}_{Residual}$ should be large enough to give a reasonable indication of the PDE residuals across the entire flow domain at any given training epoch, but should not be too large such that computing residuals incurs too high a computational cost. From trial and error, a sampling size of 512 is found to be a good compromise for all flow cases encountered.

2.4 Training Data

As described in Section 2.1, the reference velocity field for calculating $\mathscr{L}_{Data}$ is the noisy MRV image that is to be de-noised. It is therefore necessary to describe how these noisy MRV images are obtained, whether synthetically or experimentally.

2.4.1 Synthetic Data

In Chapters 3 and 5 we test the PINN approach on noisy MRV images that are generated synthetically. The process of generating this synthetic data is described below.

First, the ground truth velocity field over domain Ω, i.e. $\mathbf{u}^*_\Omega = (u^*_x,\ u^*_y,\ u^*_z)_\Omega$, is generated. This is done using FEM, where the system of discretised equations are solved using the Schur-complement method [Kontogiannis and Juniper, 2020; Zhang, 2006]. Moreover, if the flow field generated is 2D-axisymmetric, it is possible to rotate this axisymmetric data to create a 3D version of the flow. This is what is done to generate all the 3D (and steady) MRV fields in the project. Given f_{Axial} and f_{Radial}, which are interpolating functions for axial and radial velocity extracted from 2D-axisymmetric data, the 3D velocity field is generated using:

$$u^*_x(x,\ y,\ z) = f_{Axial}(x,\ r) \tag{2.25}$$

$$u^*_y(x,\ y,\ z) = f_{Radial}(x,\ r) \cdot \sin\gamma \tag{2.26}$$

$$u^*_z(x,\ y,\ z) = f_{Radial}(x,\ r) \cdot \cos\gamma \tag{2.27}$$

where r is radial position in the axisymmetric data, and rotation angle γ varies from 0 to 2π.

Given a ground truth velocity field, the corrupted or noisy MRV field is then generated according to:

$$\left(\hat{u}_{x/y/z}\right)_\Omega = \left(u^*_{x/y/z} + \delta_{x/y/z}\right)_\Omega \tag{2.28}$$

where the noise terms $\delta_{x/y/z}$ are sampled from Gaussian distributions:

$$\delta_{x/y/z} \sim \mathcal{N}\left(0,\ \sigma^2_{x/y/z}\right) \tag{2.29}$$

Hence, the variances σ^2_x, σ^2_y, and σ^2_z are predetermined for synthetic data, by making use of the definition of SNR. The SNR of an MRV image is the ratio between the mean of the ground truth and the noise variance that corrupts this ground truth:

$$SNR = \frac{1}{\sigma^2_{x/y/z} \cdot Vol_\Omega} \int u^*_{x/y/z} \cdot d(Vol_\Omega) \tag{2.30}$$

Hence, by arbitrarily defining the SNR, it is possible to determine σ_x^2, σ_y^2, and σ_z^2:

$$\sigma^2_{x/y/z} = \frac{1}{SNR \cdot Vol_\Omega} \int u^*_{x/y/z} \cdot d(Vol_\Omega) \tag{2.31}$$

These variances are then used to weigh the losses in each spatial direction, as per the formula for $\mathscr{L}_{Data}$ (Equation 2.16).

2.4.2 Experimental Data

In Chapter 4 we de-noise real, experimental MRV data for flow through a convergent nozzle. The noisy data is obtained by means of pumping a fluid mixture of water and glycerol through a converging nozzle, whereby the nozzle is carefully positioned between the bore of a 200 MHz superconducting magnet. The imaging process is then carried out using a 2D slice-selective spin-echo imaging sequence.

The MRV data is originally 2D, but we convert it to 3D by locating an assumed axisymmetric axis and rotating this noisy axisymmetric data:

$$\hat{u}_x(x,\ y,\ z) = f_{Axial}(x,\ r) \tag{2.32}$$

$$\hat{u}_y(x,\ y,\ z) = f_{Radial}(x,\ r) \cdot \sin\gamma \tag{2.33}$$

$$\hat{u}_z(x,\ y,\ z) = f_{Radial}(x,\ r) \cdot \cos\gamma \tag{2.34}$$

The problem with directly rotating noisy MRV data is that it results in cross-sectional ring artefacts forming in the rotated 3D field, which could severely affect reconstruction results. To help prevent these rings forming, points in the 2D-axisymmetric field that are beyond three standard deviations from the mean of the velocity field are replaced by the average of their surrounding points. This process is done before data rotation and is done separately for the axial and radial velocity components.

Another issue is the choice of values for σ_x^2, σ_y^2, and σ_z^2 in calculating $\mathscr{L}_{Data}$ as per Equation 2.16. Since the value of SNR is unknown for the experimental data, the variances cannot be determined from Equation 2.31. We solve this issue by manually tuning the variances until an optimal reconstruction performance is found.

2.5 Success Metrics

Finally, a description of the success metrics are outlined. First, the success of reconstructing flow from a noisy MRV field is computed by comparing the reconstructed field with the ground truth [Kontogiannis and Juniper, 2020]:

$$ACC_{x/y/z} = \frac{\| u_{x/y/z} - u^*_{x/y/z} \|_\Omega}{\| u^*_{x/y/z} \|_\Omega} \tag{2.35}$$

which is essentially an error difference evaluated over domain Ω.

Shape inference is a significant element of the project, so a suitable success metric must be defined for it. If ζ represents the predicted flow shape via a set of streamlines, and ζ^* is the ground truth shape, then the reconstruction accuracy is:

$$ACC_{Shape} = \frac{\| \zeta - \zeta^* \|}{\| \zeta^* \|} \tag{2.36}$$

In the next three chapters, we evaluate our PINN approach on noisy MRV images corresponding to different flow scenarios. In Chapter 3, the flow scenario is 3D flow through a blood vessel with an aneurysm. We first examine reconstructing viscous flow, and then examine the reconstruction of flow with inertial forces present.

Chapter 3

3D Steady Flow Through a Blood Vessel

In this chapter the PINN method of flow reconstruction is tested by reconstructing synthetic noisy MRV fields that represent 3D steady flows through a blood vessel with an aneurysm. The effect of the residual weight r (introduced in Subsection 2.1.1), the number of hidden layers, and the number of nodes per layer, on the flow and shape reconstruction accuracies, is investigated. Specifically, three different flow sub-scenarios are examined. In Section 3.2, the flow is assumed viscous, meaning inertial terms of the Navier-Stokes are ignored in calculating the residual. In Section 3.3, inertial terms are accounted for, where the flow is set at $Re \approx 100$. In Section 3.4, the reconstruction of flow at $Re \approx 1000$ is investigated.

3.1 Training Hyperparameters

All the PINNs in this chapter are trained over 10000 epochs using the SM3 algorithm, with a non-adapted learning rate $\eta = 0.001$. The Swish activation function is used in these PINNs, and the weights are He-initialised. At every training epoch, 2048 points are randomly sampled from the input space to evaluate $\mathscr{L}_{Data}$, and 512 points are randomly sampled to evaluate $\mathscr{L}_{Residual}$. Keeping the aforementioned hyperparameters constant, the residual weight, number of hidden layers, and number of nodes per layer are varied in this chapter. We explore two schemes for the residual weight - one in which it activates according to a step change and stays constant throughout training, and one in which it varies exponentially every 100th epoch:

$$r = 1 - exp\left[\frac{-(Epoch\,Number)}{200000}\right] \tag{3.1}$$

which varies the residual weight from 0% to 5% over 10000 epochs. Section 2.3 outlines more detail on PINN training, as well as justification of the hyperparameter choices.

3.2 Viscous Flow

3.2.1 Viscous Flow: Assumptions

The flow in this section is assumed 3D, steady, and viscous. This means the input to the PINN is $(x,\ y,\ z)$, and the output is $(u_x,\ u_y,\ u_z)$, which is a slight modification to Figure 2.2. The ground truth velocity field is also generated via FEM according to these assumptions, and this ground truth is corrupted to yield a field with an SNR of 0.3 (see Subsection 2.4.1 for more details of how the training data is generated). Additionally, Equation 2.18 for calculating the PDE residual at a given point in the flow field during PINN training reduces to:

$$RES_q = \sum_{i=1}^{3}\left[\frac{\partial u_i}{\partial x_i}\right]^2 - \sum_{i=1}^{3}\sum_{j=1}^{3}\left[\frac{\partial^2 \omega_i}{\partial x_j \partial x_j}\right]^2 \tag{3.2}$$

3.2.2 Viscous Flow: Results

Figure 3.1 plots the training loss history. Figures 3.2, 3.4, and 3.6 show the ground truth, noisy, and reconstructed velocity fields for the three velocity components. Figures 3.3, 3.5, and 3.7 show the absolute discrepancy between the ground truth and reconstructed velocity field for the three components. Figures 3.8 and 3.9 compare the ground truth shape and the reconstructed shape derived from the reconstructed flow field. Tables 3.1, 3.2, and 3.3 give an idea of how reconstruction accuracies of flow field and shape vary with the residual weighting r, number of hidden layers, and number of nodes per layer, respectively.

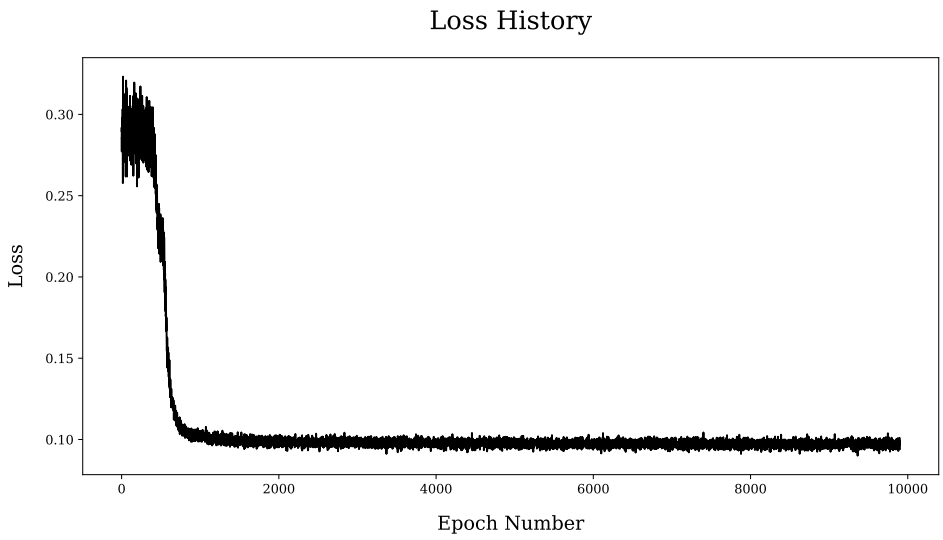

Fig. 3.1 Loss history during PINN training, for de-noising 3D steady viscous flow through a blood vessel with an aneurysm. The PINN uses a residual weighting of 1%, 30 hidden layers, and 225 nodes per layer.

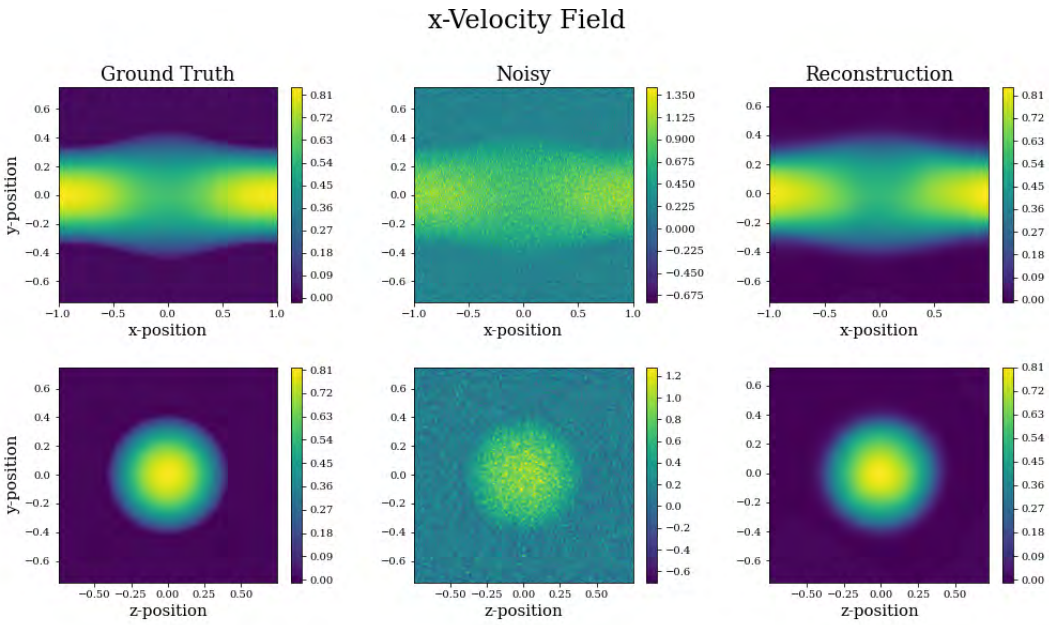

Fig. 3.2 Field of x-velocity for 3D steady viscous flow through a blood vessel with an aneurysm. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_x^*, obtained synthetically via FEM (ignoring inertial terms). The middle column is the noisy MRV field $\hat{u}_x$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_x, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

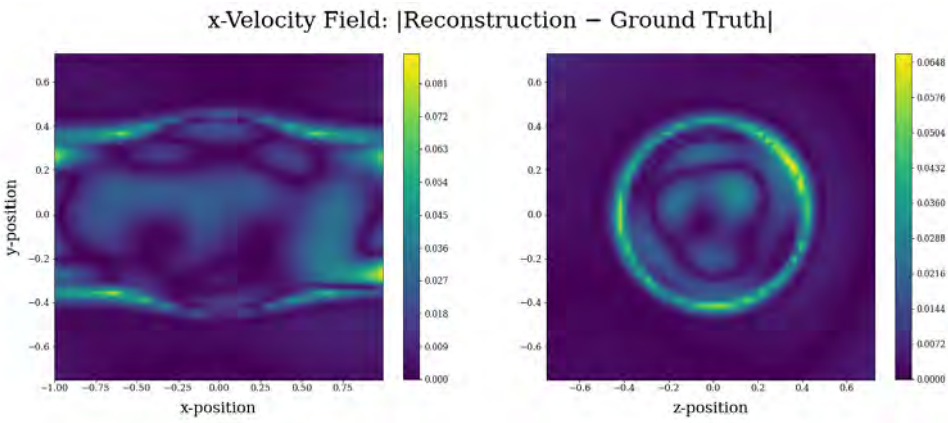

Fig. 3.3 Absolute discrepancy between the reconstruction and ground truth x-velocity fields from Figure 3.2. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

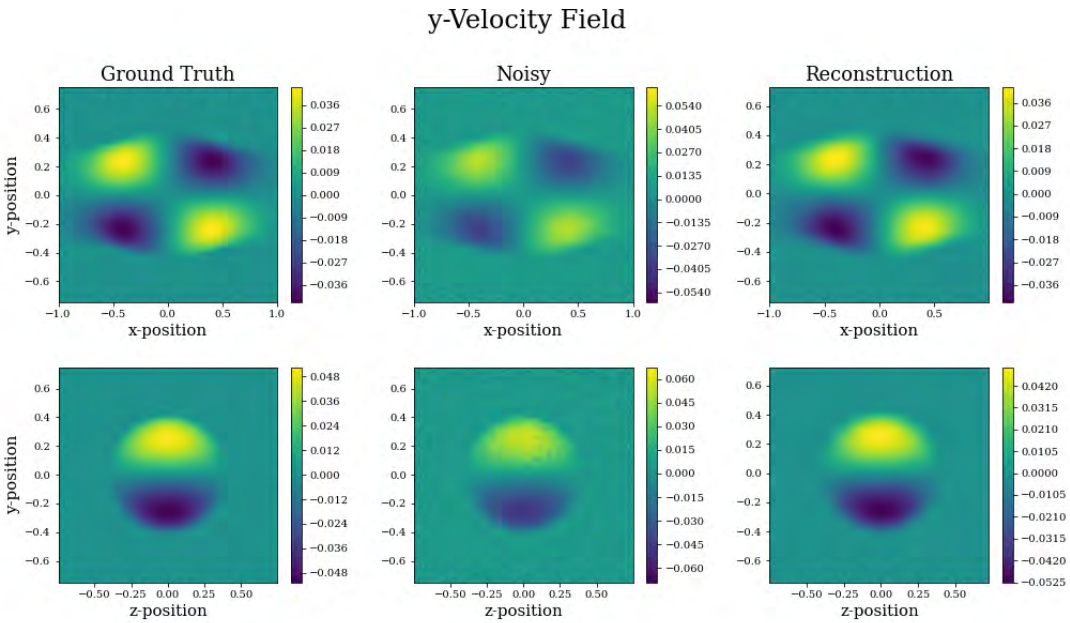

Fig. 3.4 Field of y-velocity for 3D steady viscous flow through a blood vessel with an aneurysm. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_y^*, obtained synthetically via FEM (ignoring inertial terms). The middle column is the noisy MRV field $\hat{u}_y$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_y, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

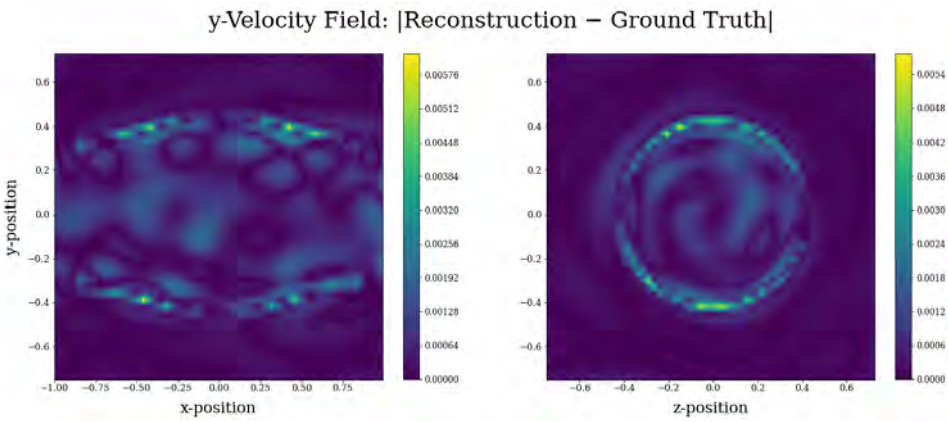

Fig. 3.5 Absolute discrepancy between the reconstruction and ground truth y-velocity fields from Figure 3.4. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

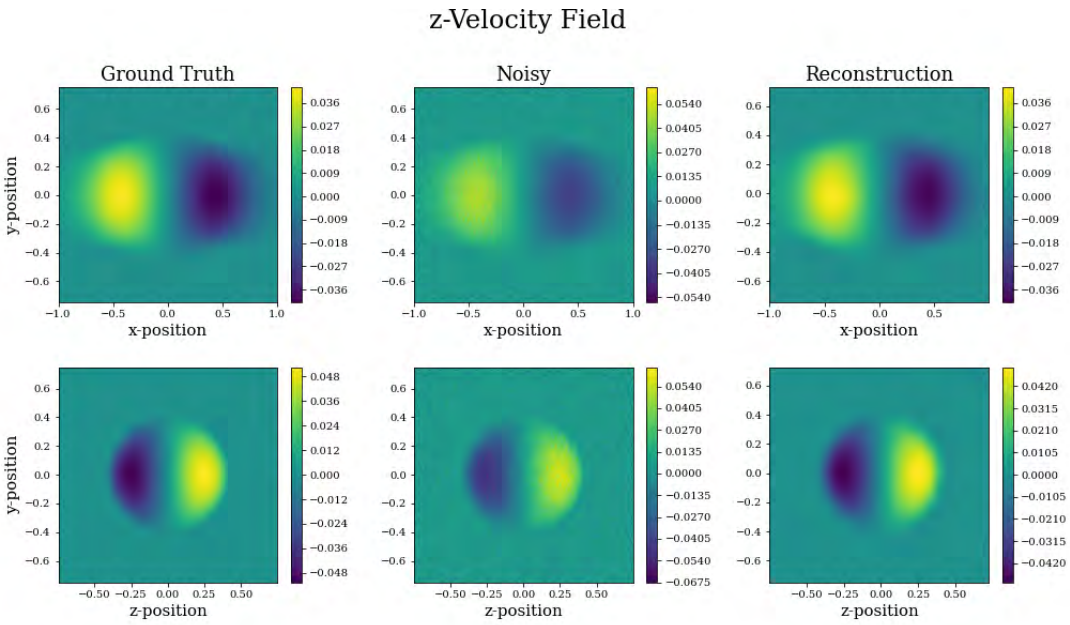

Fig. 3.6 Field of z-velocity for 3D steady viscous flow through a blood vessel with an aneurysm. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_z^*, obtained synthetically via FEM (ignoring inertial terms). The middle column is the noisy MRV field $\hat{u}_z$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_z, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

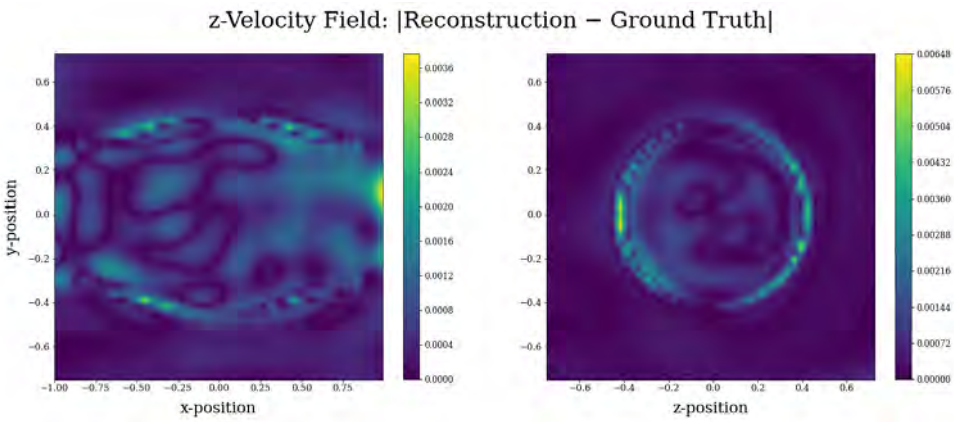

Fig. 3.7 Absolute discrepancy between the reconstruction and ground truth z-velocity fields from Figure 3.6. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

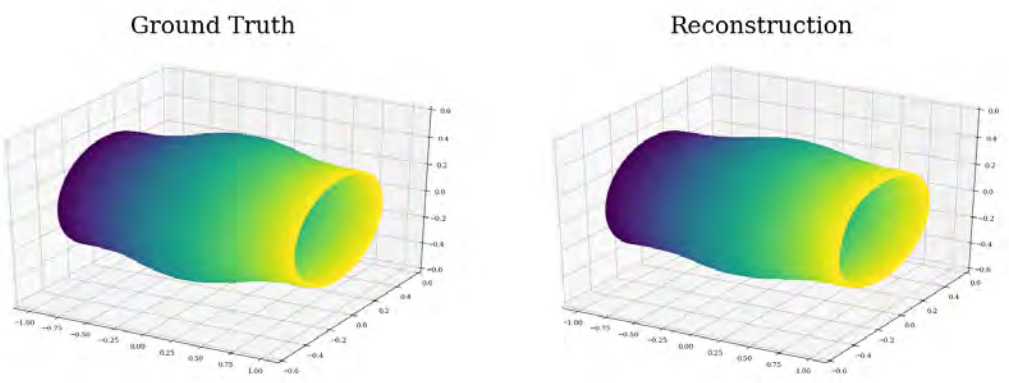

Fig. 3.8 Ground truth flow shape and the reconstructed flow shape, for 3D steady viscous flow through a blood vessel with an aneurysm. The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field. The colour-map represents x-position.

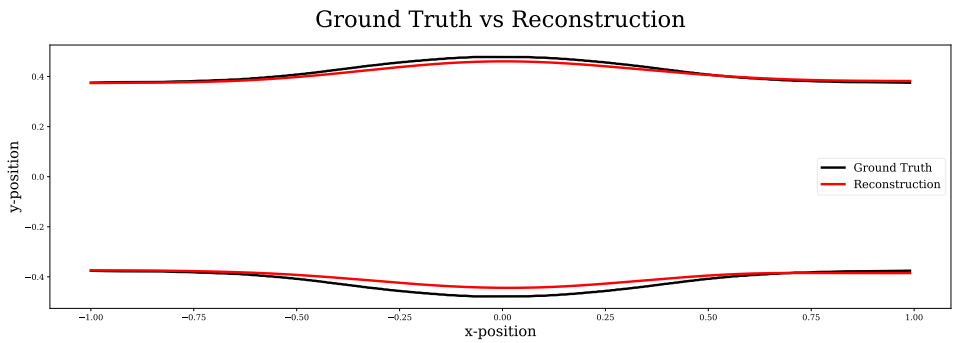

Fig. 3.9 A comparison of the ground truth flow shape and the reconstructed flow shape, for 3D steady viscous flow through a blood vessel with an aneurysm. The shapes are plotted on the xy plane where $z = 0$. The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field.

Table 3.1 Effect of residual weight r on reconstruction accuracies of the flow field and flow shape, for 3D steady viscous flow through a blood vessel with an aneurysm. For all cases, the number of hidden layers is 30, and the number of nodes per layer is 225. A residual weight of 0% means the neural network is no longer physics-informed. 'Exp Variation' refers to variation of residual weight as per Equation 3.1.

Residual Weight (%)	ACC_x (%)	ACC_y (%)	ACC_z (%)	ACC_{Shape} (%)
0	94.72	94.13	93.40	97.39
1	94.51	94.08	93.39	97.42
3	93.58	93.99	93.38	97.50
5	92.21	93.90	93.37	97.57
Exp Variation	92.40	93.65	93.41	97.56

Table 3.2 Effect of number of hidden layers on reconstruction accuracies of the flow field and flow shape, for 3D steady viscous flow through a blood vessel with an aneurysm. For all cases, the number of nodes per layer is 225, and the residual weight is set to 1%.

Hidden Layers	ACC_x (%)	ACC_y (%)	ACC_z (%)	ACC_{Shape} (%)
30	94.51	94.08	93.39	97.42
35	93.02	91.08	91.91	92.91
40	95.12	95.49	94.78	96.79
45	93.16	93.83	93.35	92.75

Table 3.3 Effect of number of nodes per layer on reconstruction accuracies of the flow field and flow shape, for 3D steady viscous flow through a blood vessel with an aneurysm. For all cases, the number of hidden layers is 30, and the residual weight is set to 1%.

Nodes per Layer	ACC_x (%)	ACC_y (%)	ACC_z (%)	ACC_{Shape} (%)
225	94.51	94.08	93.39	97.42
250	94.89	94.32	94.52	97.03
275	94.29	93.39	93.35	97.21
300	94.99	93.32	94.20	97.33
325	95.86	92.83	95.45	94.78

3.3 Flow at 100 Reynolds Number

3.3.1 Flow at 100 Reynolds Number: Assumptions

The flow in this section is assumed 3D, steady, and at $Re \approx 100$. This means the input to the PINN is $(x,\ y,\ z)$, and the output is $(u_x,\ u_y,\ u_z)$, which is a slight modification to Figure 2.2. The ground truth velocity field is also generated via FEM according to these assumptions, and this ground truth is corrupted to yield a field with an SNR of 0.3 (see Subsection 2.4.1 for more details of how the training data is generated). Additionally, Equation 2.18 for calculating the PDE residual at a given point in the flow field during PINN training reduces to:

$$RES_q = \sum_{i=1}^{3} \left[\frac{\partial u_i}{\partial x_i} \right]^2 + \sum_{i=1}^{3} \sum_{j=1}^{3} \left[u_j \cdot \frac{\partial u_i}{\partial x_j} - \omega_j \cdot \frac{\partial u_i}{\partial x_j} - \nu \cdot \frac{\partial^2 \omega_i}{\partial x_j \partial x_j} \right]^2 \tag{3.3}$$

where the kinematic viscosity ν is set at 0.01, implying that:

$$Re = \frac{u \cdot l}{\nu} \approx 100 \tag{3.4}$$

3.3.2 Flow at 100 Reynolds Number: Results

Figure 3.10 plots the training loss history. Figures 3.11, 3.13, and 3.15 show the ground truth, noisy, and reconstructed velocity fields for the three velocity components. Figures 3.12, 3.14, and 3.16 show the absolute discrepancy between the ground truth and reconstructed velocity field for the three components. Figures 3.17 and 3.18 compare the ground truth shape and the reconstructed shape derived from the reconstructed flow field. Table 3.4 gives an idea of how reconstruction accuracies of flow field and shape vary with the residual weighting r.

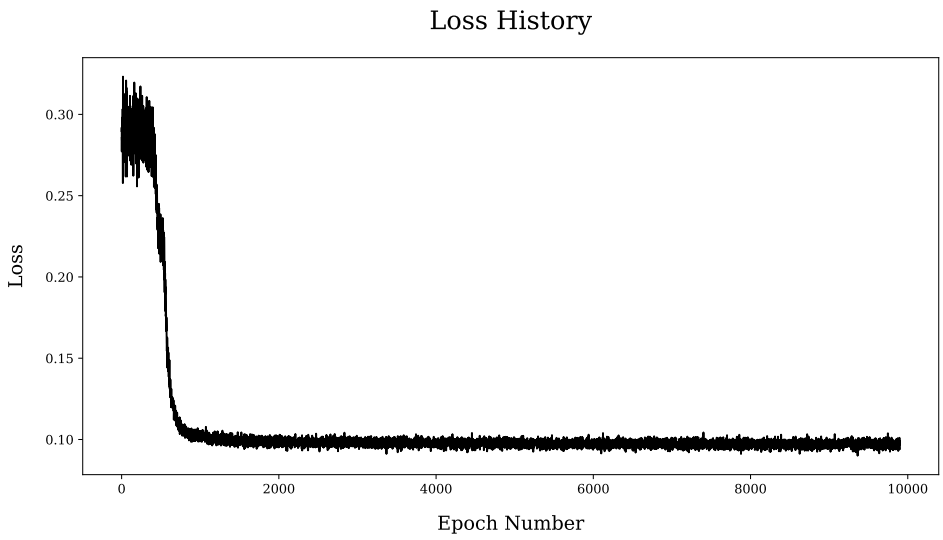

Fig. 3.10 Loss history during PINN training, for de-noising 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 100$. The PINN uses a residual weighting of 1%, 30 hidden layers, and 225 nodes per layer.

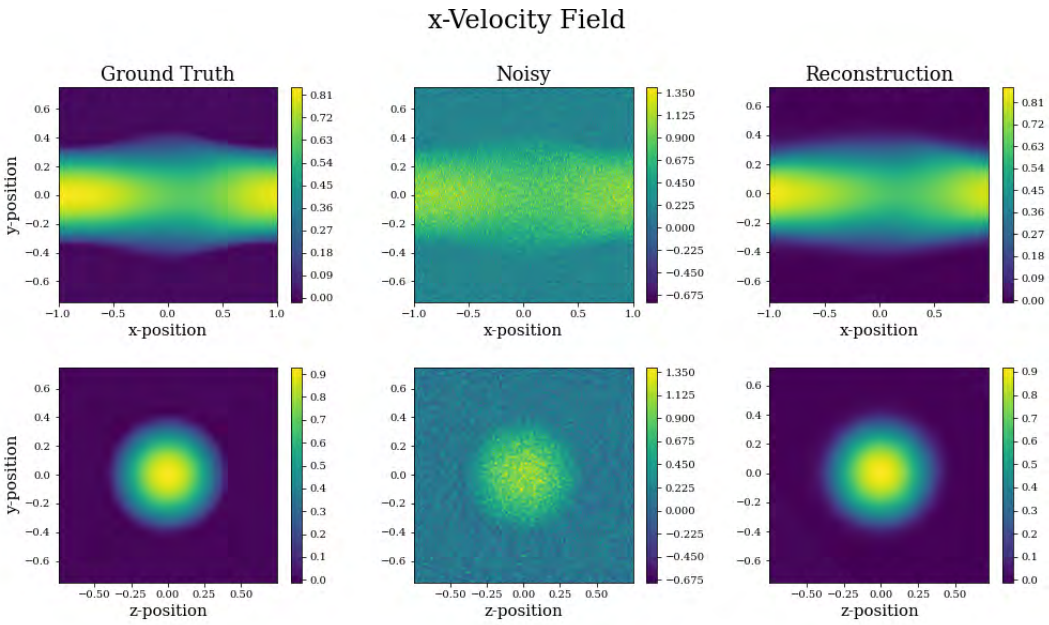

Fig. 3.11 Field of x-velocity for 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 100$. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_x^*, obtained synthetically via FEM. The middle column is the noisy MRV field $\hat{u}_x$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_x, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

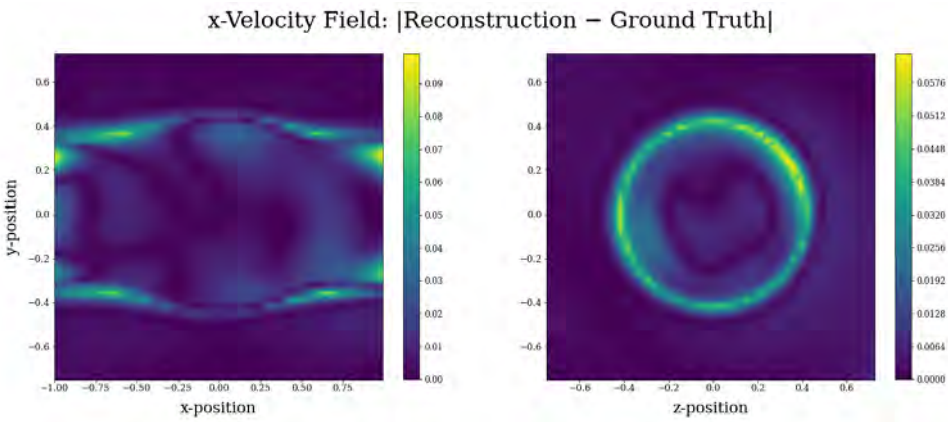

Fig. 3.12 Absolute discrepancy between the reconstruction and ground truth x-velocity fields from Figure 3.11. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

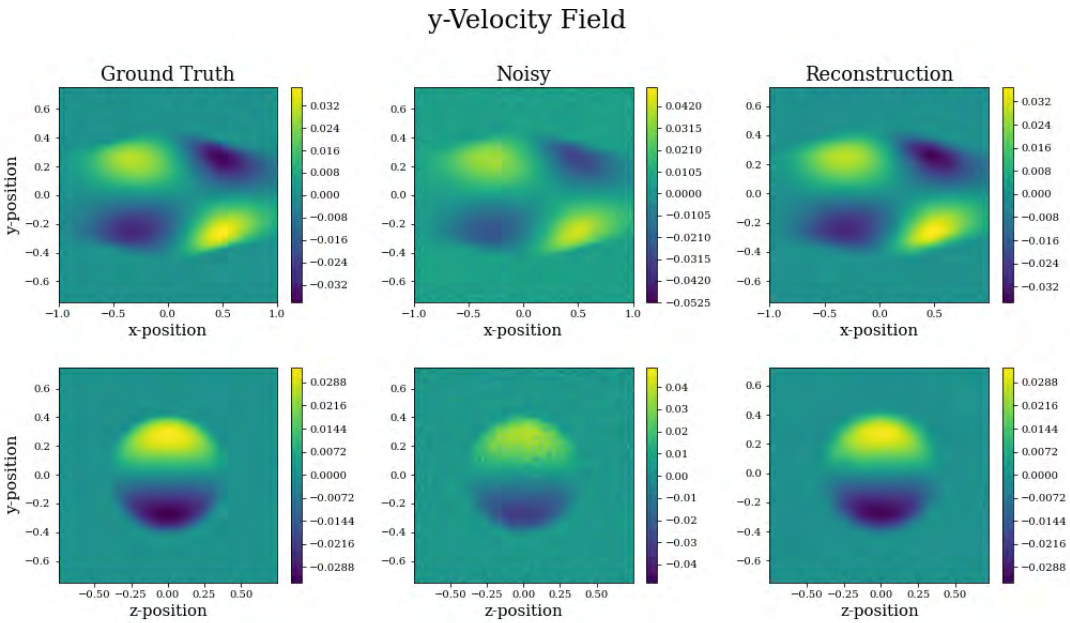

Fig. 3.13 Field of y-velocity for 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 100$. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_y^*, obtained synthetically via FEM. The middle column is the noisy MRV field $\hat{u}_y$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_y, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

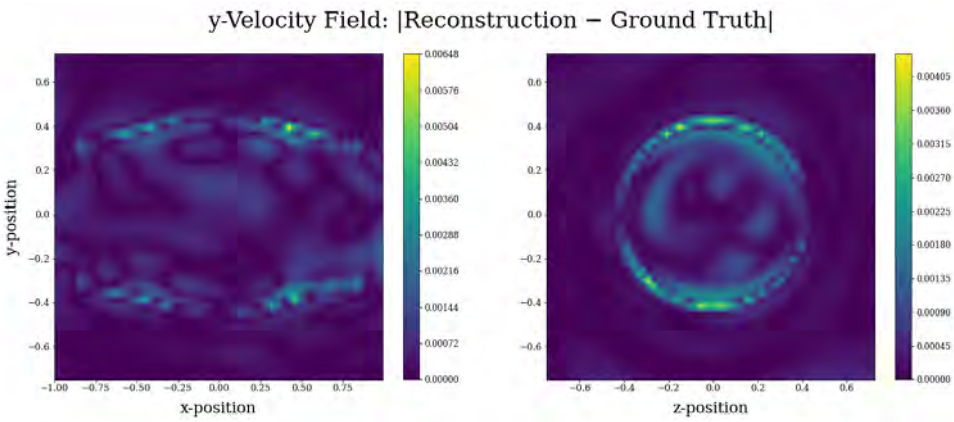

Fig. 3.14 Absolute discrepancy between the reconstruction and ground truth y-velocity fields from Figure 3.13. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

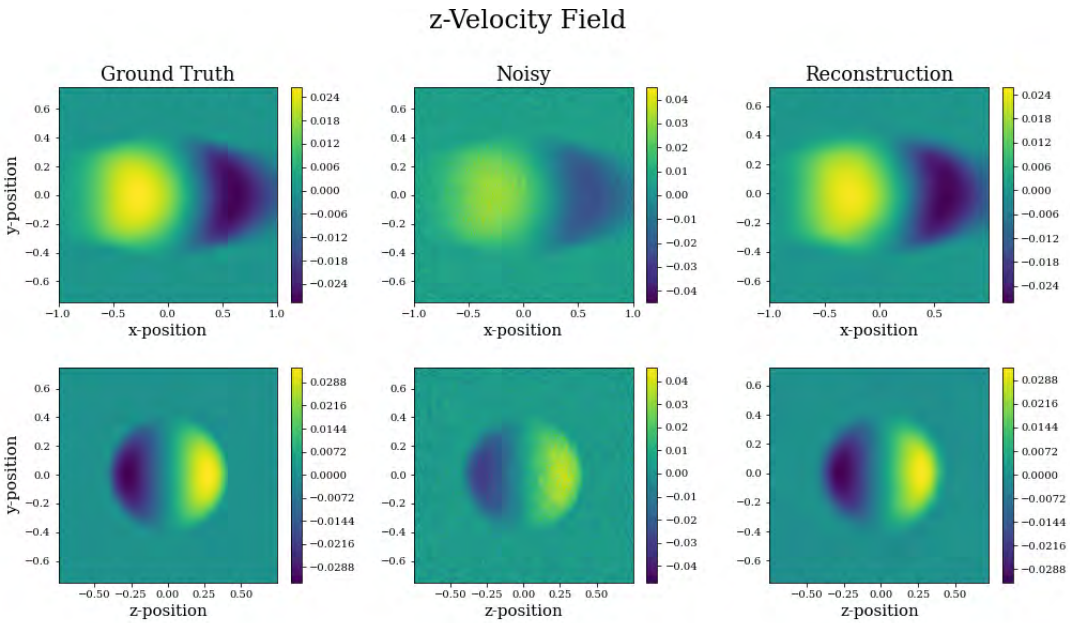

Fig. 3.15 Field of z-velocity for 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 100$. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_z^*, obtained synthetically via FEM. The middle column is the noisy MRV field $\hat{u}_z$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_z, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

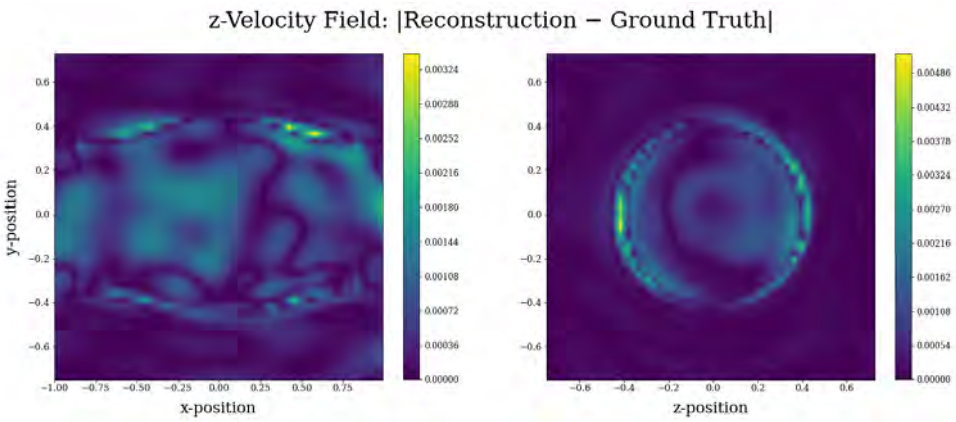

Fig. 3.16 Absolute discrepancy between the reconstruction and ground truth z-velocity fields from Figure 3.15. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

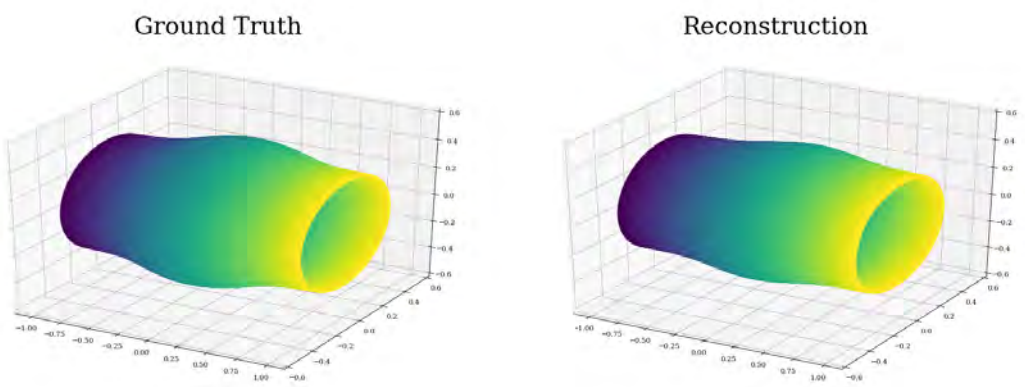

Fig. 3.17 A comparison of the ground truth flow shape and the reconstructed flow shape, for 3D steady flow through a blood vessel with an aneurysm ($Re \approx 100$). The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field. The colour-map represents x-position.

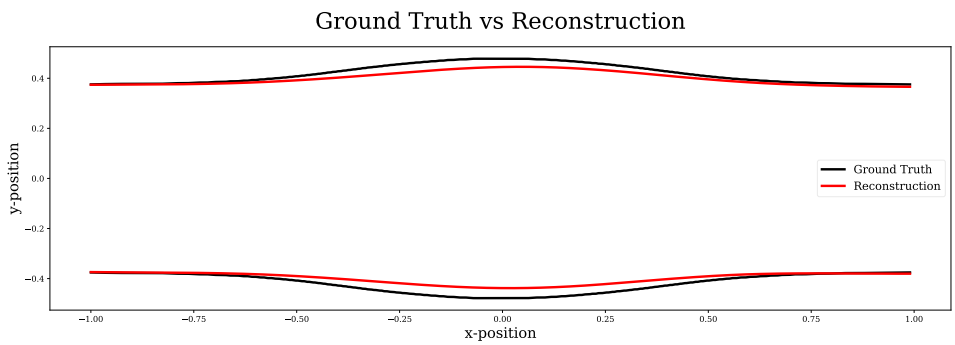

Fig. 3.18 A comparison of the ground truth flow shape and the reconstructed flow shape, for 3D steady flow through a blood vessel with an aneurysm ($Re \approx 100$). The shapes are plotted on the xy plane where $z = 0$. The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field.

Table 3.4 Effect of residual weight r on reconstruction accuracies of the flow field and flow shape, for 3D steady flow through a blood vessel with an aneurysm ($Re \approx 100$). For all cases, the number of hidden layers is 30, and the number of nodes per layer is 225. A residual weight of 0% means the neural network is no longer physics-informed. 'Exp Variation' refers to variation of residual weight as per Equation 3.1.

Residual Weight (%)	ACC_x (%)	ACC_y (%)	ACC_z (%)	ACC_{Shape} (%)
0	94.56	93.29	93.17	96.14
1	94.62	93.36	93.12	96.07
3	94.33	93.42	93.16	96.53
5	94.33	93.52	93.24	96.53
Exp Variation	94.19	93.31	93.03	96.54

3.4 Flow at 1000 Reynolds Number

3.4.1 Flow at 1000 Reynolds Number: Assumptions

The flow in this section is assumed 3D, steady, and at $Re \approx 1000$. This means the input to the PINN is $(x,\ y,\ z)$, and the output is $(u_x,\ u_y,\ u_z)$, which is a slight modification to Figure 2.2. The ground truth velocity field is also generated via FEM according to these assumptions, and this ground truth is corrupted to yield a field with an SNR of 0.3 (see Subsection 2.4.1 for more details of how the training data is generated). Additionally, Equation 2.18 for calculating the PDE residual at a given point in the flow field during PINN training reduces to:

$$RES_q = \sum_{i=1}^{3} \left[\frac{\partial u_i}{\partial x_i}\right]^2 + \sum_{i=1}^{3}\sum_{j=1}^{3} \left[u_j \cdot \frac{\partial u_i}{\partial x_j} - \omega_j \cdot \frac{\partial u_i}{\partial x_j} - \nu \cdot \frac{\partial^2 \omega_i}{\partial x_j \partial x_j}\right]^2 \tag{3.5}$$

where the kinematic viscosity ν is set at 0.001, implying that:

$$Re = \frac{u \cdot l}{\nu} \approx 1000 \tag{3.6}$$

3.4.2 Flow at 1000 Reynolds Number: Results

Figure 3.19 plots the training loss history. Figures 3.20, 3.22, and 3.24 show the ground truth, noisy, and reconstructed velocity fields for the three velocity components. Figures 3.21, 3.23, and 3.25 show the absolute discrepancy between the ground truth and reconstructed velocity field for the three components. Figures 3.26 and 3.27 compare the ground truth shape and the reconstructed shape derived from the reconstructed flow field. Table 3.5 gives an idea of how reconstruction accuracies of flow field and shape vary with the residual weighting r.

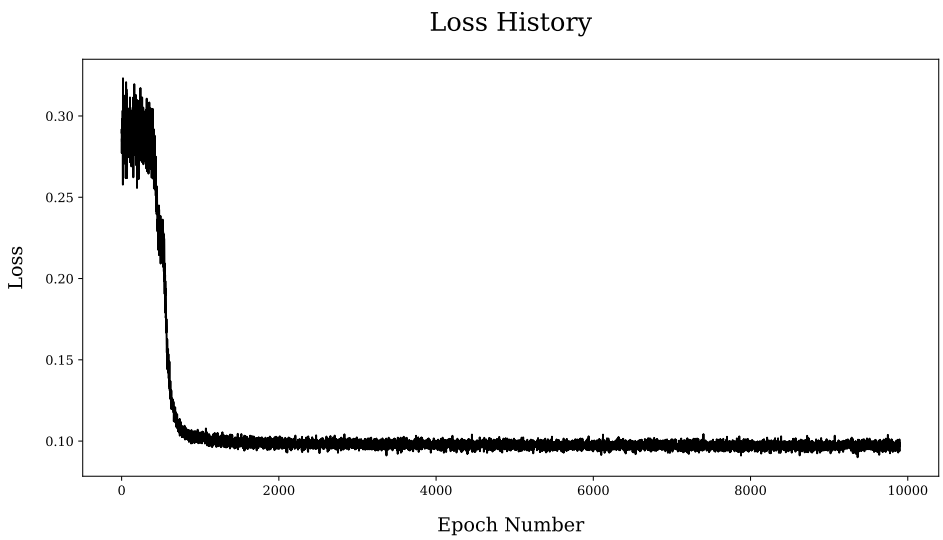

Fig. 3.19 Loss history during PINN training, for de-noising 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 1000$. The PINN uses a residual weighting of 1%, 30 hidden layers, and 225 nodes per layer.

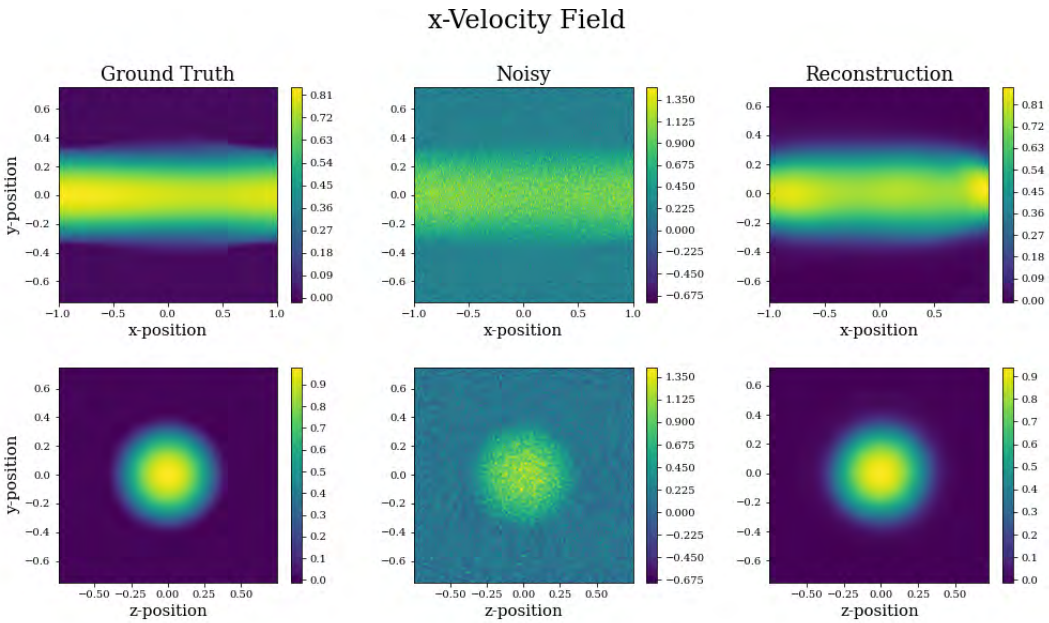

Fig. 3.20 Field of x-velocity for 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 1000$. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_x^*, obtained synthetically via FEM. The middle column is the noisy MRV field $\hat{u}_x$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_x, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

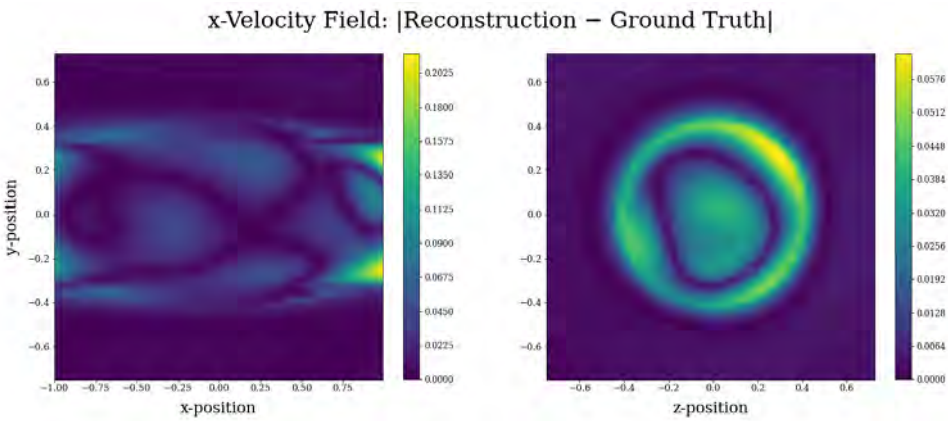

Fig. 3.21 Absolute discrepancy between the reconstruction and ground truth x-velocity fields from Figure 3.20. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

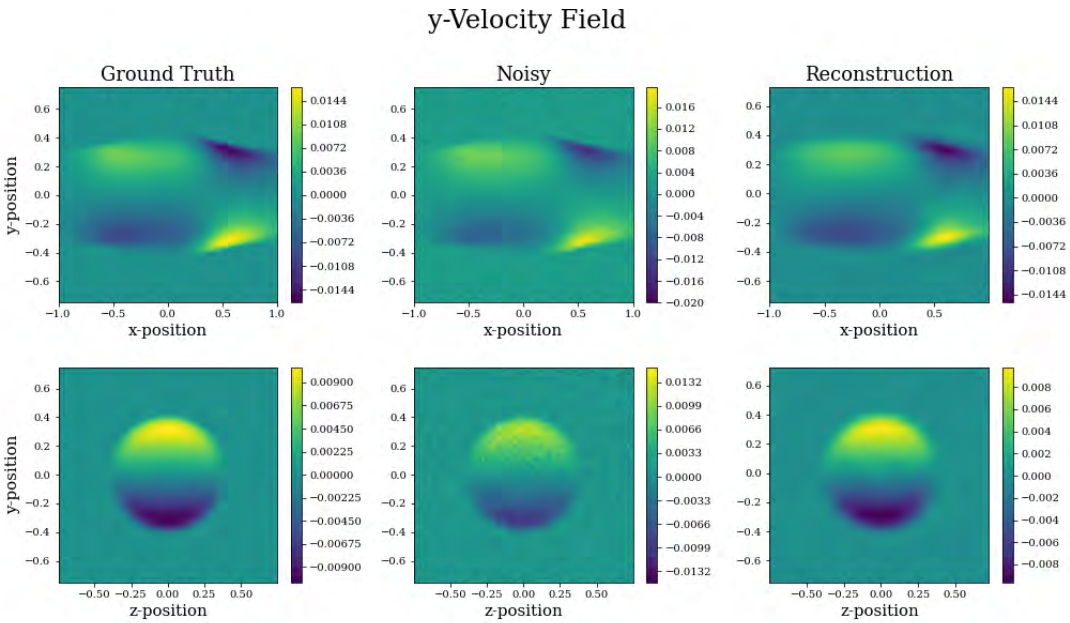

Fig. 3.22 Field of y-velocity for 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 1000$. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_y^*, obtained synthetically via FEM. The middle column is the noisy MRV field $\hat{u}_y$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_y, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

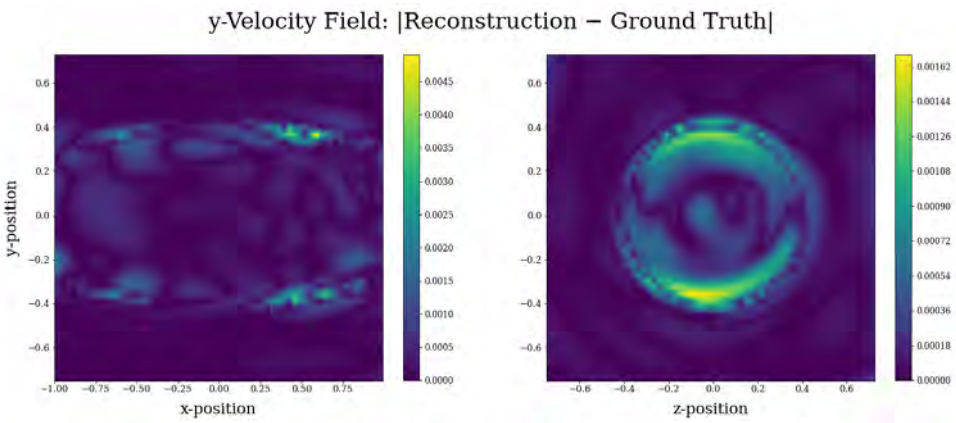

Fig. 3.23 Absolute discrepancy between the reconstruction and ground truth y-velocity fields from Figure 3.22. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

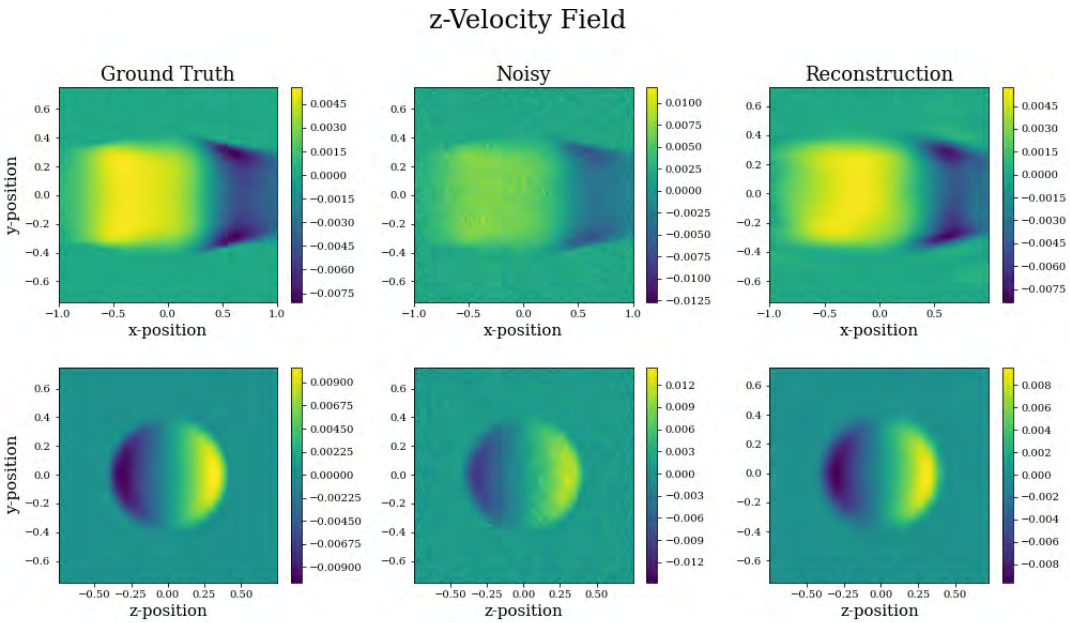

Fig. 3.24 Field of z-velocity for 3D steady flow through a blood vessel with an aneurysm, at $Re \approx 1000$. The top row of contours represent velocities at a reference yx plane where $z = 0.1$. The bottom row of contours represent the velocities at a reference yz plane where $x = -0.44$. The left column represents the ground truth u_z^*, obtained synthetically via FEM. The middle column is the noisy MRV field $\hat{u}_z$, obtained by corrupting the ground truth with noise. The right column is the PINN reconstruction u_z, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

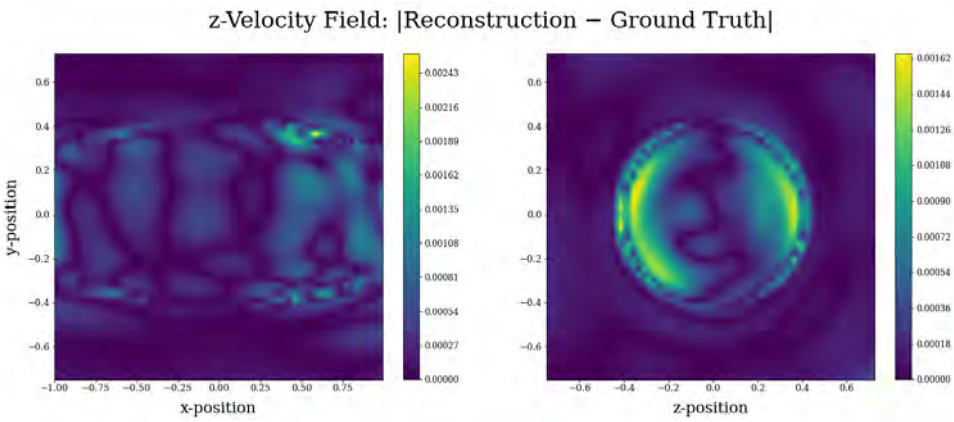

Fig. 3.25 Absolute discrepancy between the reconstruction and ground truth z-velocity fields from Figure 3.24. The left plot represents the discrepancy at a reference yx plane where $z = 0.1$. The right plot represents the discrepancy at a reference yz plane where $x = -0.44$.

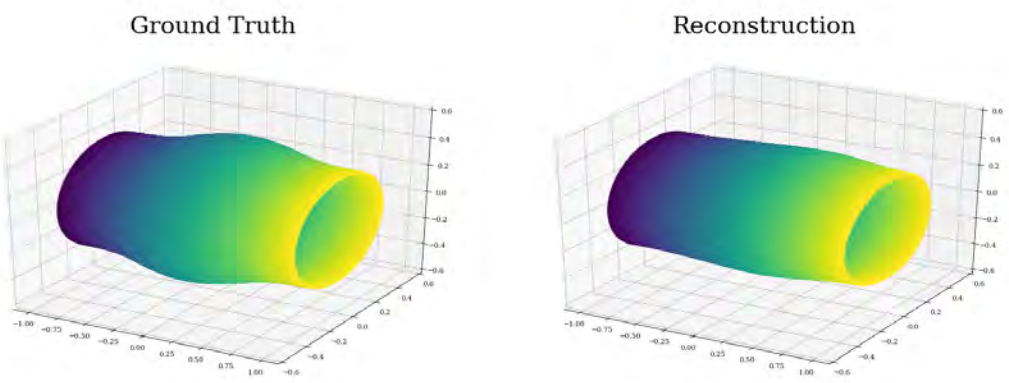

Fig. 3.26 A comparison of the ground truth flow shape and the reconstructed flow shape, for 3D steady flow through a blood vessel with an aneurysm ($Re \approx 1000$). The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field. The colour-map represents x-position.

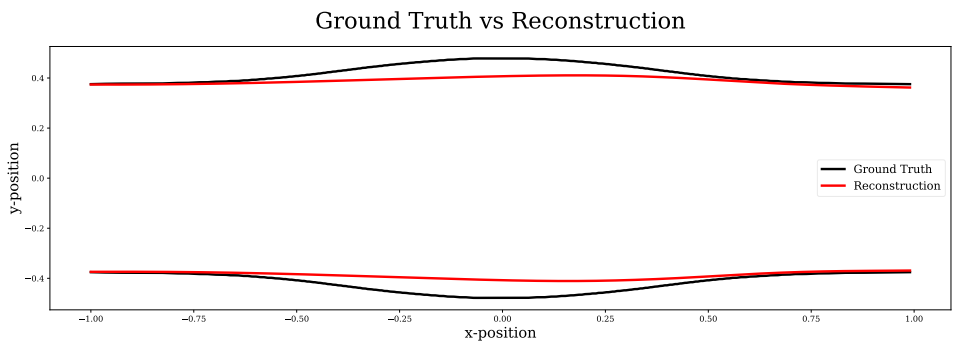

Fig. 3.27 A comparison of the ground truth flow shape and the reconstructed flow shape, for 3D steady flow through a blood vessel with an aneurysm ($Re \approx 1000$). The shapes are plotted on the xy plane where $z = 0$. The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field.

Table 3.5 Effect of residual weight r on reconstruction accuracies of the flow field and flow shape, for 3D steady flow through a blood vessel with an aneurysm ($Re \approx 1000$). For all cases, the number of hidden layers is 30, and the number of nodes per layer is 225. A residual weight of 0% means the neural network is no longer physics-informed. 'Exp Variation' refers to variation of residual weight as per Equation 3.1.

Residual Weight (%)	ACC_x (%)	ACC_y (%)	ACC_z (%)	ACC_{Shape} (%)
0	90.53	85.70	84.78	94.56
1	90.89	85.53	84.80	94.51
3	91.08	85.19	84.70	94.41
5	90.97	85.00	84.60	94.36
Exp Variation	90.89	85.21	84.56	94.38

3.5 Discussion

Figures 3.2, 3.4, 3.6, 3.11, 3.13, 3.15, 3.20, 3.22, and 3.24 indicate that flow field reconstructions for viscous and $Re \approx 100$ flow cases are reconstructed faithfully from a visual point of view, and that the reconstruction is reasonable for flow at $Re \approx 1000$. It should be noted however that the flow gradients appear to be over-smeared in the vicinity of the boundary. This is expected because Swish activation functions encourage a smoother PINN output landscape (Figure 2.4).

Figures 3.3, 3.5, 3.7, 3.12, 3.14, 3.16, 3.21, 3.23, and 3.25 indicate high discrepancy between ground truths and reconstructions near the vicinity of the flow boundary. Furthermore the measured PDE residuals are high in these regions. This could be due to the over-smearing mentioned earlier and due to not incorporating any BC residuals into the loss function.

Boundary shapes for viscous flow (Figures 3.8 and 3.9) and flow at $Re \approx 100$ (Figures 3.17 and 3.18) also appear to be reconstructed faithfully, although the boundary bump of the aneurysm is less pronounced compared to the ground truth shapes. This could also be due to the smearing of the flow gradients in the reconstructed flow fields. Moreover, the bump is almost completely gone in the shape inference for the flow at $Re \approx 1000$, as per Figures 3.26 and 3.27, which is a concern from a diagnostic point of view. The flattening of the aneurysmal bump in the reconstructed shape may be due to the method of inferring the boundary via streamlines. It appears that, at high flow speeds (i.e. high Re number) where

viscous effects are relatively small, the shape of streamlines diverge from the actual shape of the physical boundary in the case of flow through a bulging blood vessel.

Despite the gradient smearing in the reconstructed flow fields, and the de-bulging of the blood vessel in the reconstructed shape, Tables 3.1, 3.2, 3.3, 3.4 and 3.5 indicate reconstruction accuracies ranging from 85% to 97%.

For viscous flow, Table 3.1 indicates that a zero residual weighting (i.e. a normal neural network) results in the most accurate flow fields, but that a residual weighting of 5% results in the most accurate inferred shape. It is unclear why this residual weighting performs optimally for ACC_{Shape}, but sub-optimally for ACC_x, ACC_y, and ACC_z. However this observation somewhat demonstrates the potential of PINNs. Table 3.2 seems to indicate more accurate flow field reconstructions when the number of hidden layers is higher (see 40 hidden layers for example), but shape reconstructions seem to drop off with number of layers. A similar trend is observed with nodes per layer as in Table 3.3, with the use of 325 nodes per layer resulting in the highest flow field accuracies but the lowest shape accuracy.

Tables 3.4 and 3.5 give an idea of the variation of reconstruction accuracies with residual weighting for flow cases at $Re \approx 100$ and $Re \approx 1000$ respectively. The tables indicate that, when all parameters are kept constant except residual weighting, it is possible to achieve higher reconstruction accuracies on at least one of ACC_x, ACC_y, ACC_z, ACC_{Shape} with a non-zero residual weighting, albeit not all of them simultaneously. In theory, an optimal set of hyperparameters should result in higher reconstruction accuracies for a given non-zero residual weighting, in comparison to a neural network with the same hyperparameters and zero residual weighting. This is because the PINN encourages a lower residual loss minimum that better agrees with physics. It is therefore unclear why improvements are not simultaneous over all success metrics when a PINN is used. Possible reasons could be inadequate hyperparameter tuning, or error introductions when rotating 2D-axisymmetric data into 3D data (for example from any interpolations, see Subsection 2.4.1).

In the next chapter, we turn to de-noising experimental MRV data that represents 3D steady flow through a convergent nozzle.

Chapter 4

3D Steady Flow Through a Nozzle

In this chapter, the PINN method of flow reconstruction is tested by reconstructing an experimentally obtained noisy MRV field that represents 3D steady flow through a convergent nozzle. Specifically, the effect of the residual weight parameter r introduced in Subsection 2.1.1 is examined.

4.1 Training Hyperparameters

All the PINNs in this chapter are trained over 10000 epochs using the SM3 algorithm, with a non-adapted learning rate $\eta = 0.001$. The Swish activation function is used in these PINNs, and the weights are He-initialised. At every training epoch, 2048 points are randomly sampled from the input space to evaluate $\mathscr{L}_{Data}$, and 512 points are randomly sampled to evaluate $\mathscr{L}_{Residual}$. Keeping the aforementioned hyperparameters constant, the residual weight r is varied in this chapter. We explore two schemes for the residual weight - one in which it activates according to a step change and stays constant throughout training, and one in which it varies exponentially every 100th epoch:

$$r = 1 - exp\left[\frac{-(Epoch\,Number)}{200000}\right] \tag{4.1}$$

which varies the residual weight from 0% to 5% over 10000 epochs. Section 2.3 outlines more detail on PINN training, as well as justification of the hyperparameter choices.

4.2 Flow Assumptions

The flow in this section is assumed 3D, steady, and at $Re \approx 162$. This means the input to the PINN is (x, y, z), and the output is (u_x, u_y, u_z), which is a slight modification to Figure 2.2. A high SNR MRV field that is experimentally obtained is assumed to have the aforementioned flow assumptions, and is used as the 'ground truth' for evaluating reconstruction accuracies. Also, a low SNR MRV field- also experimentally obtained - is used as the noisy reference field during PINN training (see Subsection 2.4.2 on more details of the experimental procedure and how experimental data is pre-processed). Additionally, Equation 2.18 for calculating the PDE residual at a given point in the flow field during PINN training reduces to:

$$RES_q = \sum_{i=1}^{3} \left[\frac{\partial u_i}{\partial x_i} \right]^2 + \sum_{i=1}^{3} \sum_{j=1}^{3} \left[u_j \cdot \frac{\partial u_i}{\partial x_j} - \omega_j \cdot \frac{\partial u_i}{\partial x_j} - \nu \cdot \frac{\partial^2 \omega_i}{\partial x_j \partial x_j} \right]^2 \tag{4.2}$$

where the kinematic viscosity ν is set at 1/162, implying that:

$$Re = \frac{u \cdot l}{\nu} \approx 162 \tag{4.3}$$

4.3 Results

Figure 4.1 plots the training loss history. Figures 4.2, 4.4, and 4.6 show the ground truth, noisy, and reconstructed velocity fields for the three velocity components. Figures 4.3, 4.5, and 4.7 show the absolute discrepancy between the ground truth and reconstructed velocity field for the three components. Figures 4.8 and 4.9 compare the ground truth shape and the reconstructed shape derived from the reconstructed flow field. Table 4.1 gives an idea of how reconstruction accuracies of flow field and shape vary with the residual weighting r.

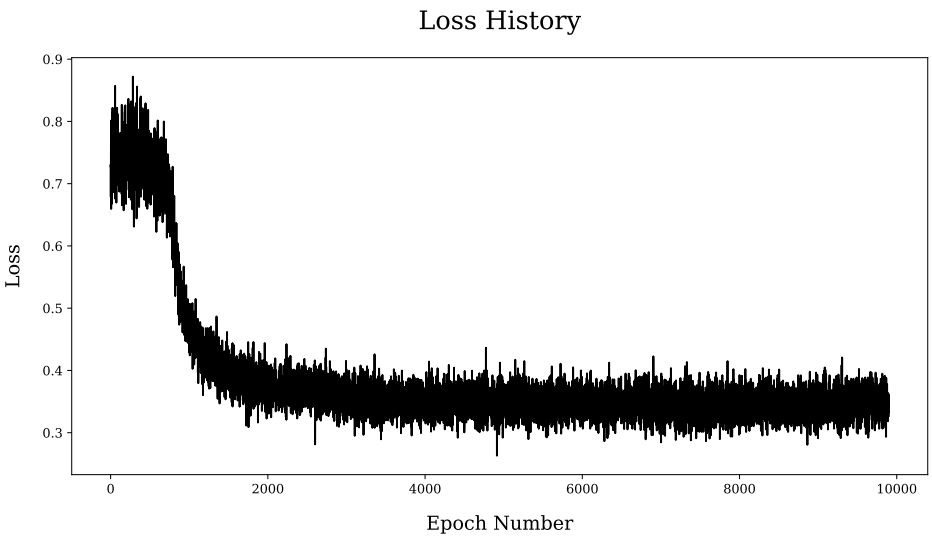

Fig. 4.1 Loss history during PINN training, for de-noising 3D steady flow through a convergent nozzle, at $Re \approx 162$. The PINN uses a residual weighting of 1%, 30 hidden layers, and 225 nodes per layer.

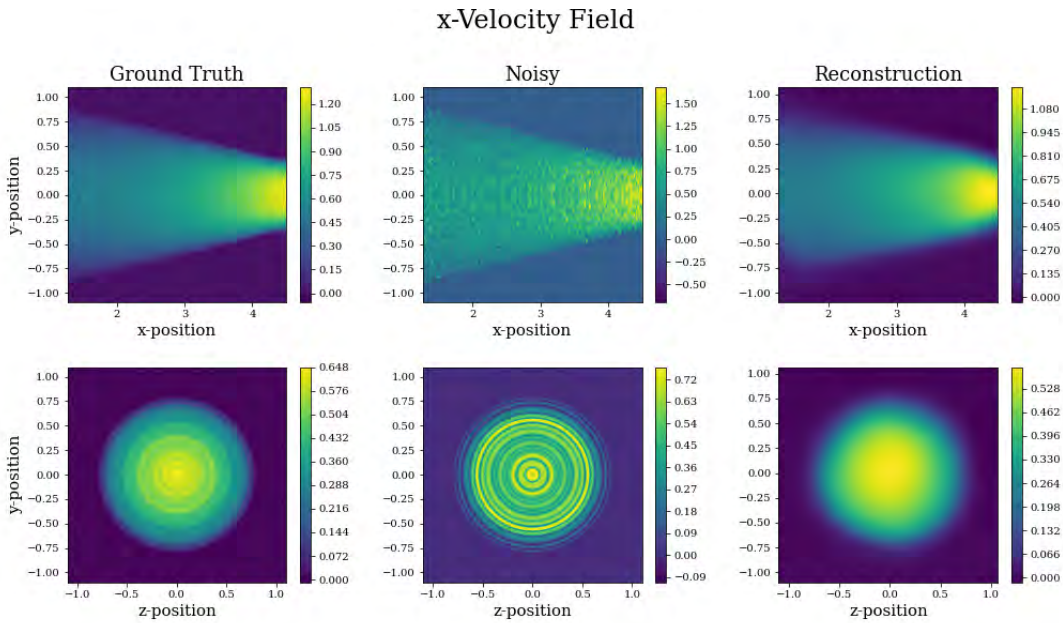

Fig. 4.2 Field of x-velocity for 3D steady flow through a convergent nozzle at $Re \approx 162$. The top row of contours represent velocities at a reference yx plane where $z = 0.44$. The bottom row of contours represent the velocities at a reference yz plane where $x = 2.2$. The left column represents the ground truth u_x^*, obtained experimentally via MR methods (high SNR). The middle column is the noisy MRV field $\hat{u}_x$, also obtained experimentally via MR methods (low SNR). The right column is the PINN reconstruction u_x, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

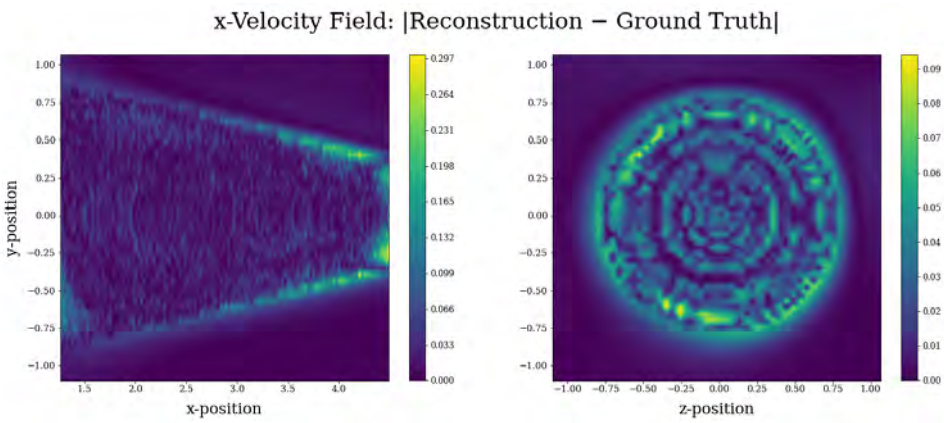

Fig. 4.3 Absolute discrepancy between the reconstruction and ground truth x-velocity fields from Figure 4.2. The left plot represents the discrepancy at a reference yx plane where $z = 0.44$. The right plot represents the discrepancy at a reference yz plane where $x = 2.2$.

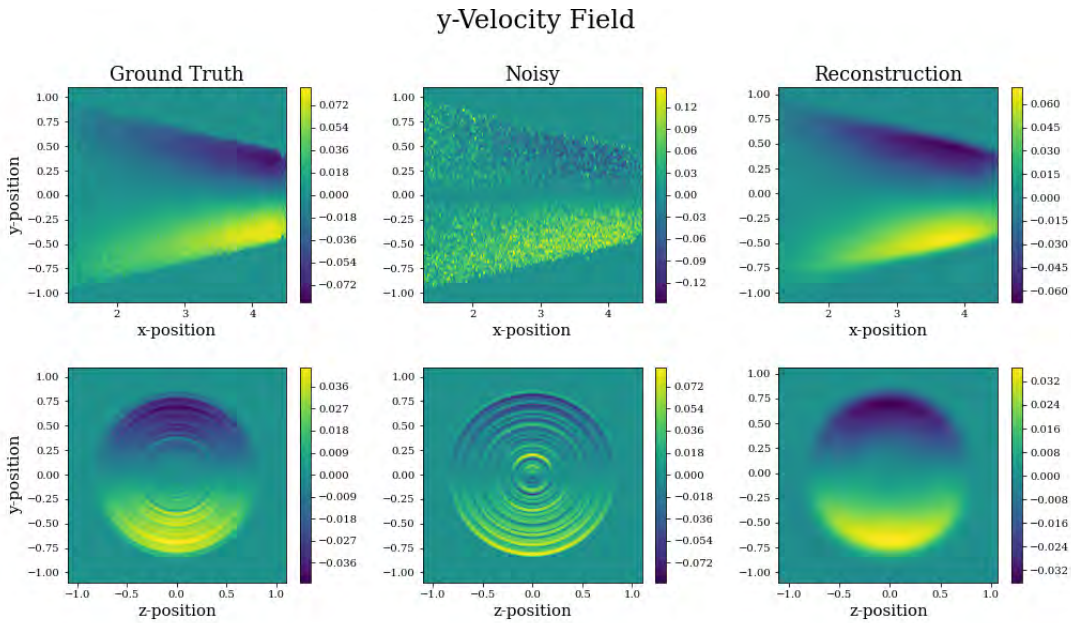

Fig. 4.4 Field of y-velocity for 3D steady flow through a convergent nozzle at $Re \approx 162$. The top row of contours represent velocities at a reference yx plane where $z = 0.44$. The bottom row of contours represent the velocities at a reference yz plane where $x = 2.2$. The left column represents the ground truth u_y^*, obtained experimentally via MR methods (high SNR). The middle column is the noisy MRV field $\hat{u}_y$, also obtained experimentally via MR methods (low SNR). The right column is the PINN reconstruction u_y, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

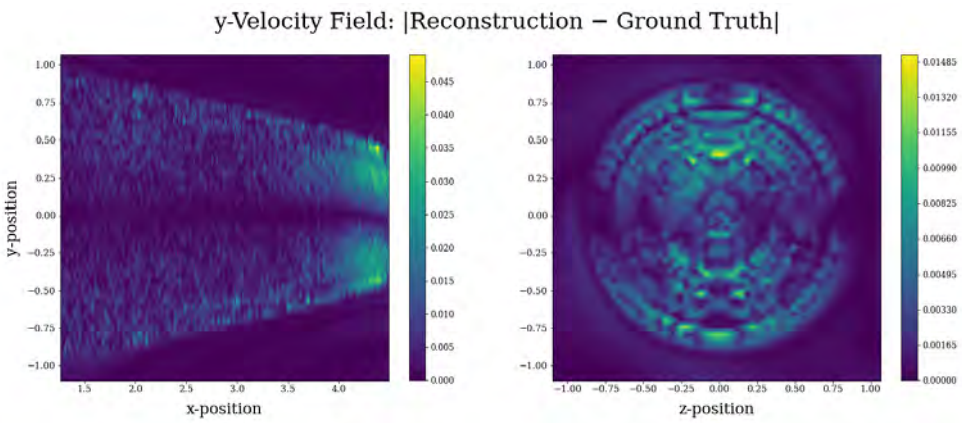

Fig. 4.5 Absolute discrepancy between the reconstruction and ground truth y-velocity fields from Figure 4.4. The left plot represents the discrepancy at a reference yx plane where $z = 0.44$. The right plot represents the discrepancy at a reference yz plane where $x = 2.2$.

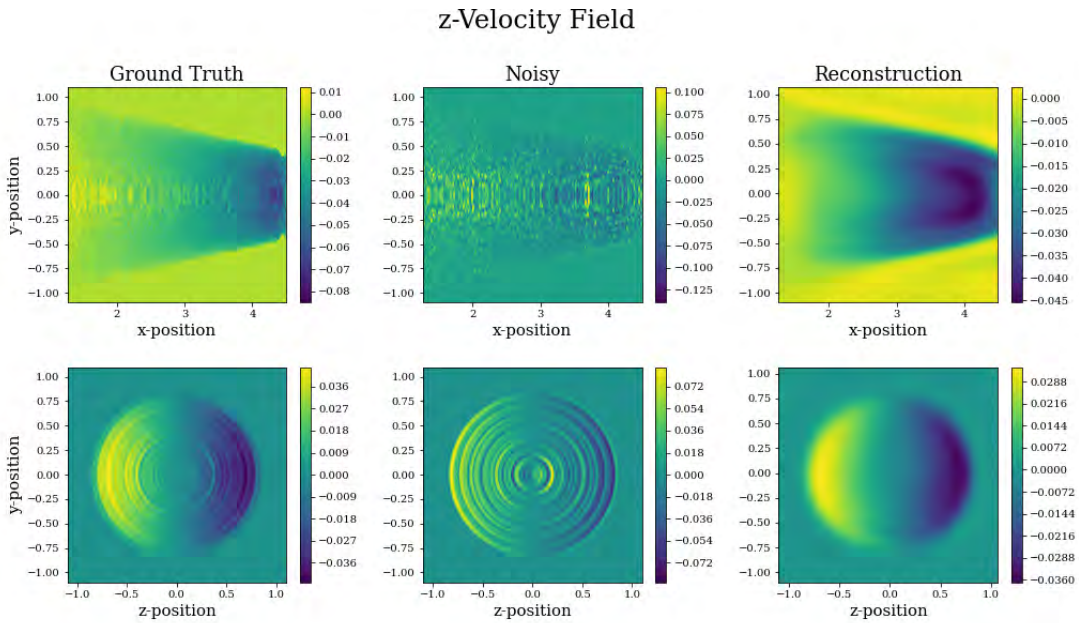

Fig. 4.6 Field of z-velocity for 3D steady flow through a convergent nozzle at $Re \approx 162$. The top row of contours represent velocities at a reference yx plane where $z = 0.44$. The bottom row of contours represent the velocities at a reference yz plane where $x = 2.2$. The left column represents the ground truth u_z^*, obtained experimentally via MR methods (high SNR). The middle column is the noisy MRV field $\hat{u}_z$, also obtained experimentally via MR methods (low SNR). The right column is the PINN reconstruction u_z, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

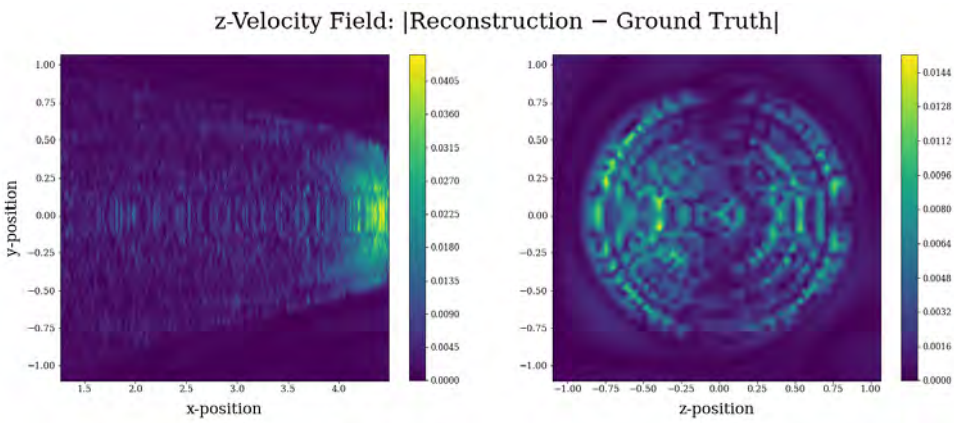

Fig. 4.7 Absolute discrepancy between the reconstruction and ground truth z-velocity fields from Figure 4.6. The left plot represents the discrepancy at a reference yx plane where $z = 0.44$. The right plot represents the discrepancy at a reference yz plane where $x = 2.2$.

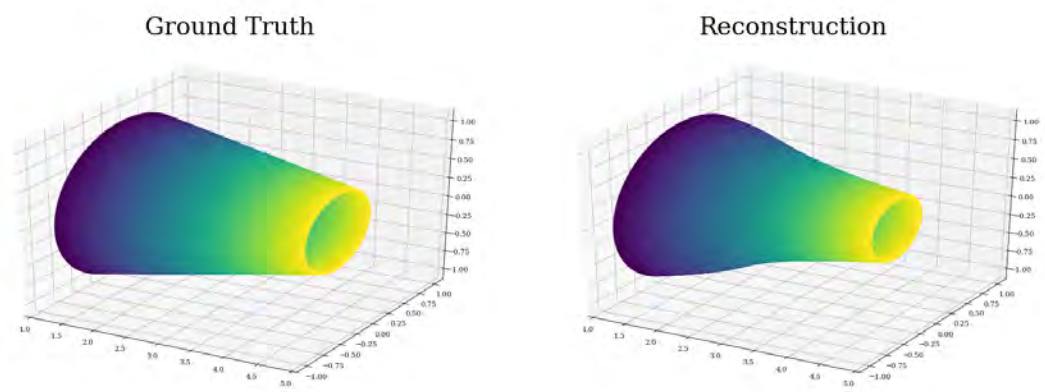

Fig. 4.8 A comparison of the ground truth flow shape and the reconstructed flow shape, for 3D steady flow through a convergent nozzle ($Re \approx 162$). The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field. The colour-map represents x-position.

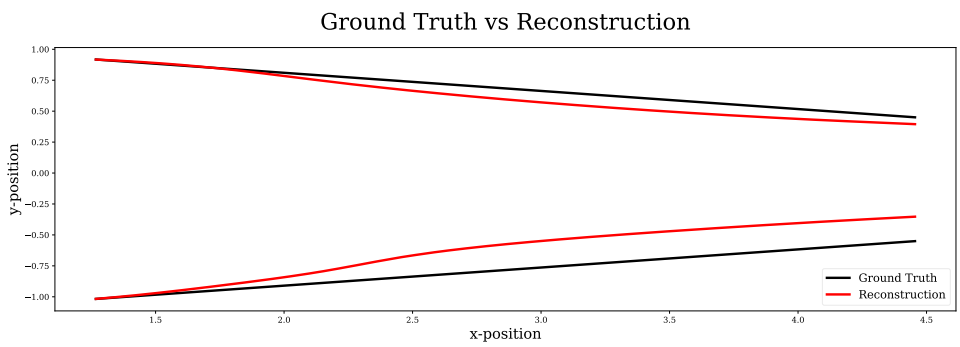

Fig. 4.9 A comparison of the ground truth flow shape and the reconstructed flow shape, for 3D steady flow through a convergent nozzle ($Re \approx 162$). The shapes are plotted on the xy plane where $z = 0$. The reconstructed flow shape is obtained by traversing streamlines across the reconstructed flow field.

Table 4.1 Effect of residual weight r on reconstruction accuracies of the flow field and flow shape, for 3D steady flow through a convergent nozzle ($Re \approx 162$). For all cases, the number of hidden layers is 30, and the number of nodes per layer is 225. A residual weight of 0% means the neural network is no longer physics-informed. 'Exp Variation' refers to variation of residual weight as per Equation 4.1.

Residual Weight (%)	ACC_x (%)	ACC_y (%)	ACC_z (%)	ACC_{Shape} (%)
0	88.15	74.85	75.05	96.21
1	88.25	74.87	75.06	96.20
3	88.16	74.87	75.06	96.17
5	87.95	74.87	75.02	96.14
Exp Variation	87.98	74.90	75.10	96.15

4.4 Discussion

Figures 4.2, 4.4, and 4.6 indicate reasonable reconstructions of the flow fields. Again, gradient smearing is observed in the reconstructions at the vicinity of the boundary due to the use of Swish activation functions that encourage smoothing on the PINN output (Figure 2.4). Figures 4.3, 4.5, and 4.7 show high discrepancy between ground truths and reconstructions near the boundary and the outlet, which are most likely due to no incorporation of BC residuals and due to over-smearing. Looking from the yz plane the discrepancy is also high at several concentric rings within the flow domain, which is due to the strange ring artefacts present in the ground truth and noisy velocity fields. The artefacts are a result of rotating 2D-axisymmetric data to generate a 3D velocity field, in which any 2D noise is rotated as well (see Subsection 2.4.2 on details of how experimental MRV training data is obtained and pre-processed).

Figures 4.8 and 4.9 show the similarity between the ground truth shape and the reconstructed shape, although it is observed that the convergence of the nozzle in the reconstruction is not linear and results in a smaller diameter at the outlet.

Table 4.1 indicates flow field reconstruction accuracies to be greater than 85% for x-velocity and around 75% for y- and z-velocity. Shape reconstruction accuracies are greater than 95% despite the shape mismatch noted earlier. The dip in performance of the PINN method in this chapter compared to the previous chapter is due to experimental errors of obtaining MRV data. These experimental errors have direct implications on the performance of the PINN

that makes use of the data. Another factor are the ring artefacts mentioned earlier. The PINN algorithm would perform better if it uses 3D data rather than 2D-axisymmetric data rotated into 3D, but raw 3D data is unavailable.

Nevertheless, it is observed from Table 4.1 that a PINN with residual weighting of 1% outperforms a normal neural network (zero residual weighting) in terms of reconstructing flow fields, although shape reconstruction is still most accurate when residual weighting is zero.

In the next chapter, the PINN model is used to de-noise synthetically generated MRV data that represents 2D unsteady flow past a cylinder.

Chapter 5

2D Unsteady Flow Past a Cylinder

In this chapter the PINN method of flow reconstruction is tested by reconstructing a synthetic noisy MRV field that represents 2D unsteady (periodic) flow past a cylinder. The effect of the residual weight parameter r (introduced in Subsection 2.1.1), the number of hidden layers, and the number of nodes per layer, on the flow reconstruction accuracies and reconstruction time, is investigated[1].

5.1 Training Hyperparameters

All the PINNs in this chapter are trained over 15000 epochs using the SM3 algorithm, with a non-adapted learning rate $\eta = 0.001$. The Swish activation function is used in these PINNs, and the weights are He-initialised. At every training epoch, 2048 points are randomly sampled from the input space to evaluate $\mathscr{L}_{Data}$, and 512 points are randomly sampled to evaluate $\mathscr{L}_{Residual}$. Keeping the aforementioned hyperparameters constant, the residual weight r, number of hidden layers, and number of nodes per layer are varied in this chapter. We explore two schemes for the residual weight - one in which it activates according to a step change and stays constant throughout training, and one in which it varies exponentially every 100th epoch:

$$r = 1 - exp\left[\frac{-(Epoch\,Number)}{300000}\right] \tag{5.1}$$

[1] Since the flow in this chapter is external flow, there are no shape inferences conducted.

which varies the residual weight from 0% to 5% over 15000 epochs. Section 2.3 outlines more detail on PINN training, as well as justification of the hyperparameter choices.

5.2 Flow Assumptions

The flow in this chapter is assumed 2D, unsteady-periodic, and at $Re \approx 100$. This means the input to the PINN is $(x,\ y,\ t)$, and the output is $(u_x,\ u_y)$, which is a slight modification to Figure 2.2. The ground truth velocity field is also generated via FEM according to these assumptions, and this ground truth is corrupted to yield a field with an SNR of 3 (see Subsection 2.4.1 for more details of how the training data is generated). Additionally, Equation 2.18 for calculating the PDE residual at a given point in the flow field during PINN training reduces to:

$$RES_q = \sum_{i=1}^{2} \left[\frac{\partial u_i}{\partial x_i}\right]^2 + \sum_{i=1}^{2} \left[\frac{\partial \omega_i}{\partial t} + \sum_{j=1}^{2} \left(u_j \cdot \frac{\partial u_i}{\partial x_j} - \omega_j \cdot \frac{\partial u_i}{\partial x_j} - \nu \cdot \frac{\partial^2 \omega_i}{\partial x_j \partial x_j}\right)\right]^2 \tag{5.2}$$

where the kinematic viscosity ν is set at 0.01, implying that:

$$Re = \frac{u \cdot l}{\nu} \approx 100 \tag{5.3}$$

5.3 Results

Figure 5.1 plots the training loss history. Figures 5.2 and 5.4 show the ground truth, noisy, and reconstructed velocity fields for the x-velocity component at half and full period, respectively. Figures 5.6 and 5.8 show the ground truth, noisy, and reconstructed velocity fields for the y-velocity component at half and full period, respectively. Figures 5.3, 5.5, 5.7 and 5.9 show the corresponding discrepancies between the ground truth and reconstructed velocity fields. Table 5.1 gives an idea of how flow field reconstruction accuracies vary with residual weighting r. Tables 5.2, and 5.3 give an idea of how reconstruction accuracies of flow field, as well as PINN training and convergence times, vary with number of hidden layers and number of nodes per layer, respectively[2].

[2]Training time refers to the time taken for the PINN to train over 15000 epochs, and convergence time is the time taken for the loss to converge to a minimum.

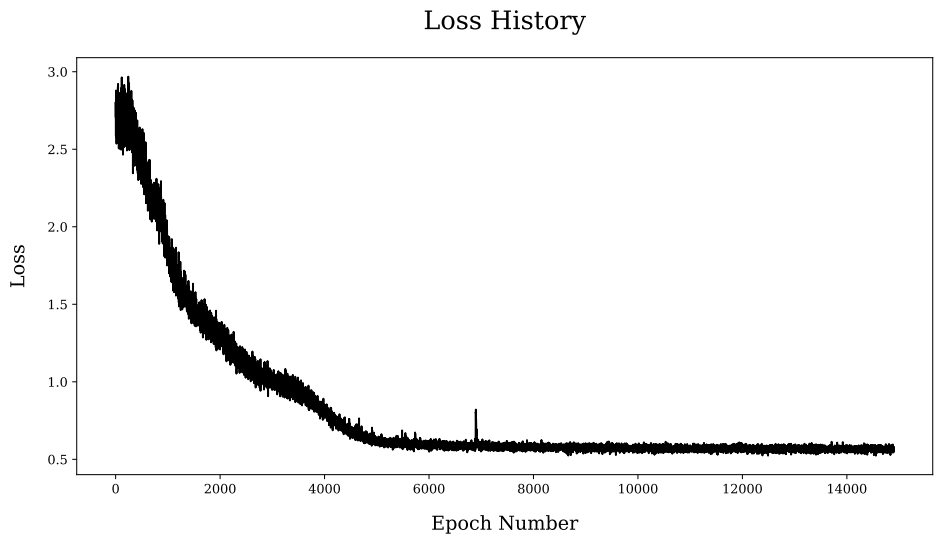

Fig. 5.1 Loss history during PINN training, for de-noising 2D unsteady flow past a cylinder, at $Re \approx 100$. The PINN uses a residual weighting of 1%, 30 hidden layers, and 225 nodes per layer.

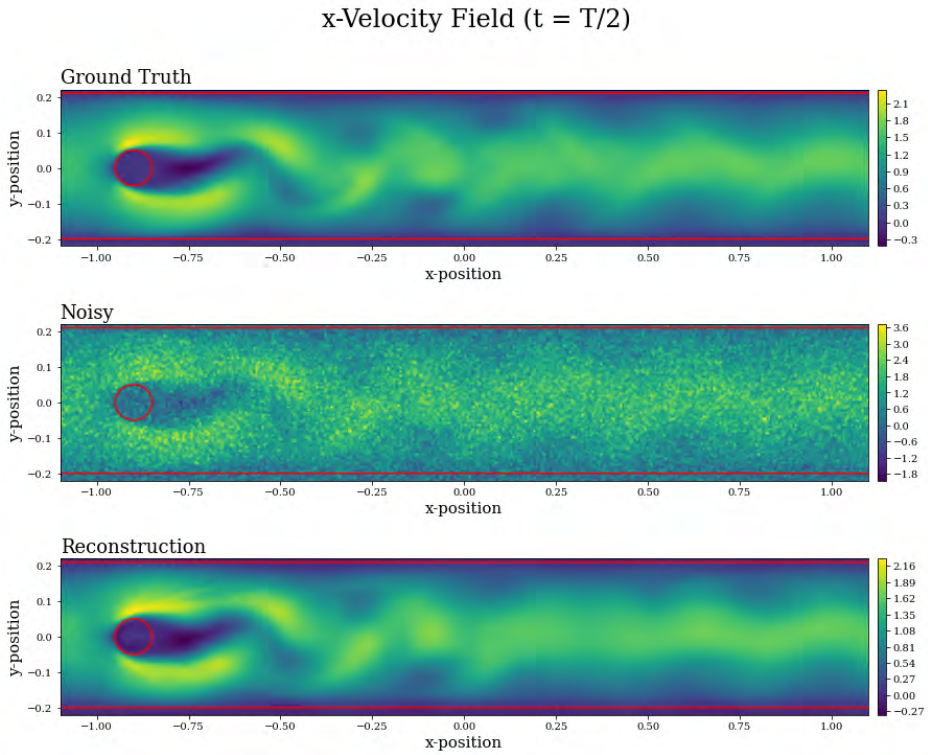

Fig. 5.2 Field of x-velocity for 2D unsteady flow past a cylinder at $t = T/2$, $Re \approx 100$. The top row represents the ground truth u_x^*, that is obtained synthetically via FEM. The middle row is the noisy MRV field $\hat{u}_x$, obtained by corrupting the ground truth with noise. The bottom row is the PINN reconstruction u_x, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

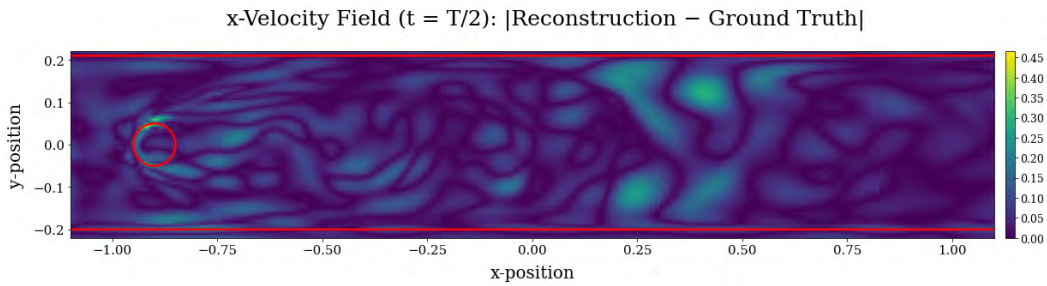

Fig. 5.3 Absolute discrepancy between the reconstruction and ground truth x-velocity fields from Figure 5.2 ($t = T/2$).

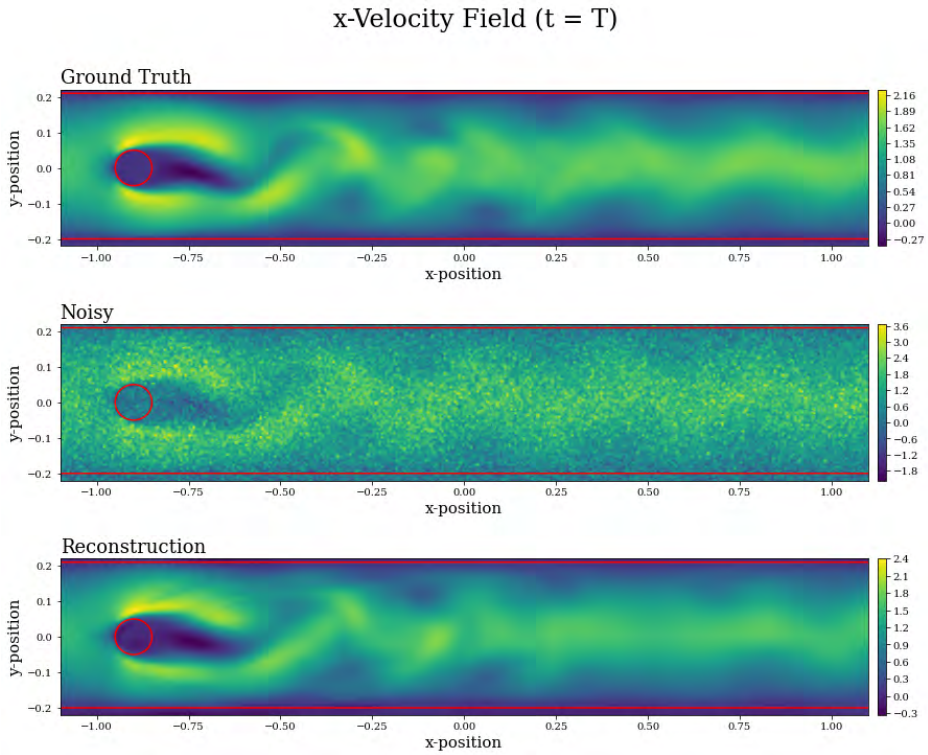

Fig. 5.4 Field of x-velocity for 2D unsteady flow past a cylinder at $t = T$, $Re \approx 100$. The top row represents the ground truth u_x^*, obtained synthetically via FEM. The middle row is the noisy MRV field $\hat{u}_x$, obtained by corrupting the ground truth with noise. The bottom row is the PINN reconstruction u_x, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

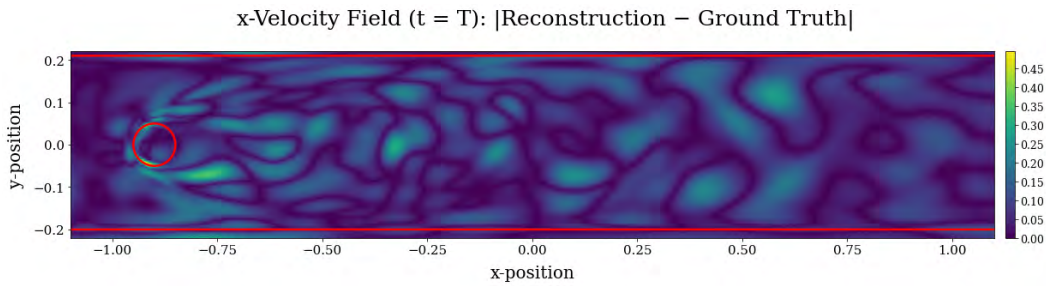

Fig. 5.5 Absolute discrepancy between the reconstruction and ground truth x-velocity fields from Figure 5.4 ($t = T$).

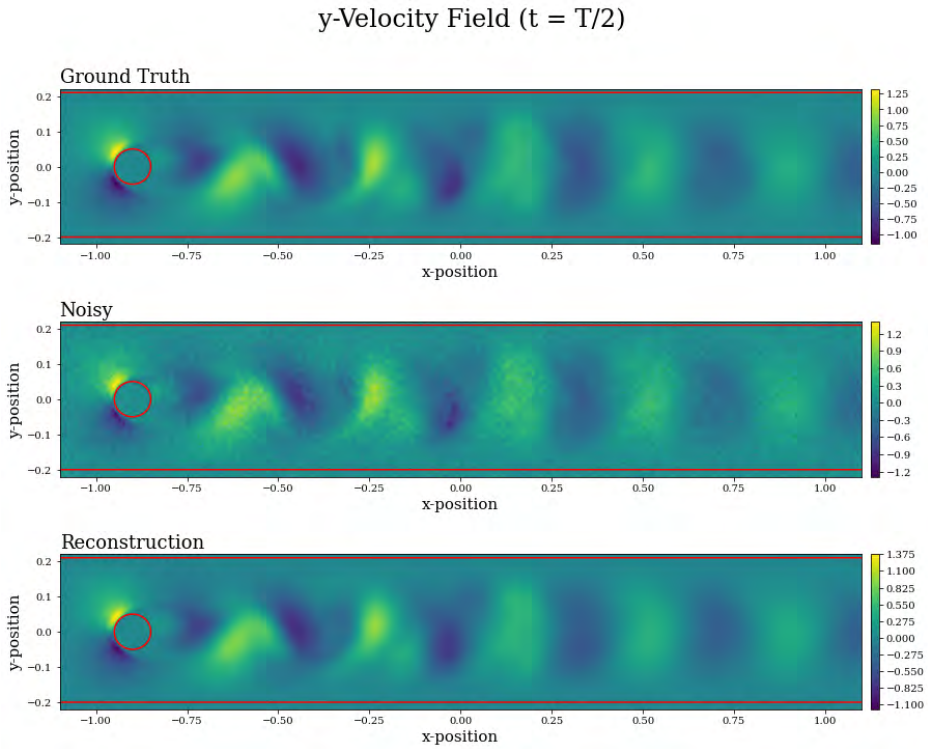

Fig. 5.6 Field of y-velocity for 2D unsteady flow past a cylinder at $t = T/2$, $Re \approx 100$. The top row represents the ground truth u_y^*, obtained synthetically via FEM. The middle row is the noisy MRV field $\hat{u}_y$, obtained by corrupting the ground truth with noise. The bottom row is the PINN reconstruction u_y, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

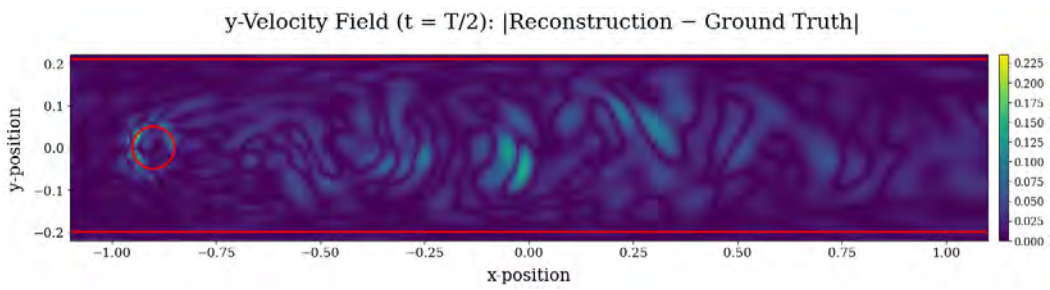

Fig. 5.7 Absolute discrepancy between the reconstruction and ground truth y-velocity fields from Figure 5.6 ($t = T/2$).

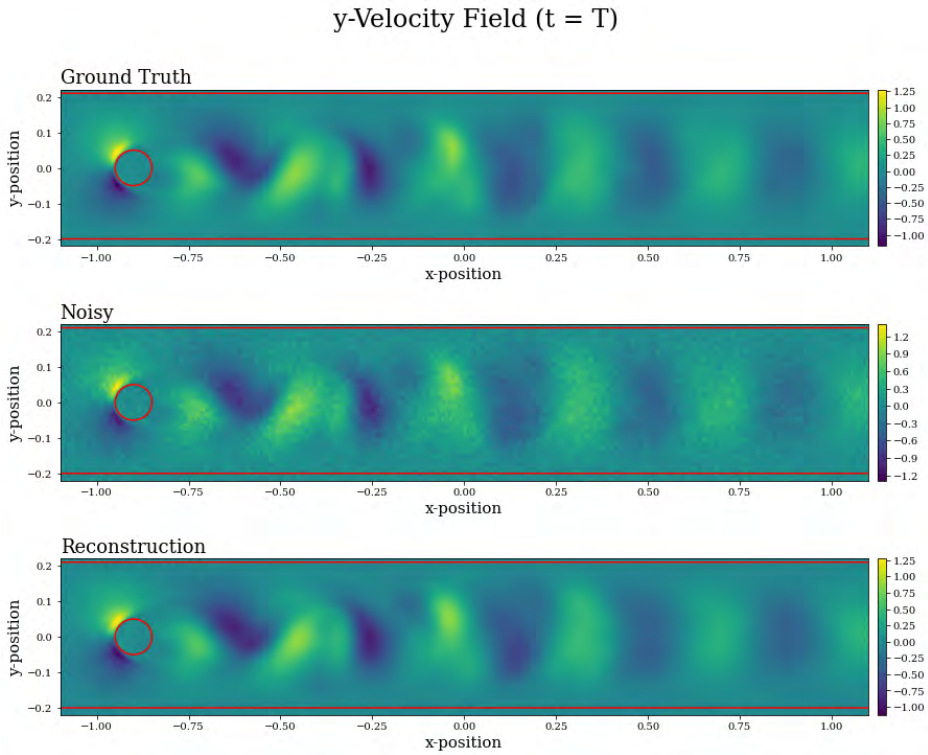

Fig. 5.8 Field of y-velocity for 2D unsteady flow past a cylinder at $t = T$, $Re \approx 100$. The top row represents the ground truth u_y^*, obtained synthetically via FEM. The middle row is the noisy MRV field $\hat{u}_y$, obtained by corrupting the ground truth with noise. The bottom row is the PINN reconstruction u_y, where the PINN has 30 hidden layers, 225 nodes per layer, and residual weighting of 1%.

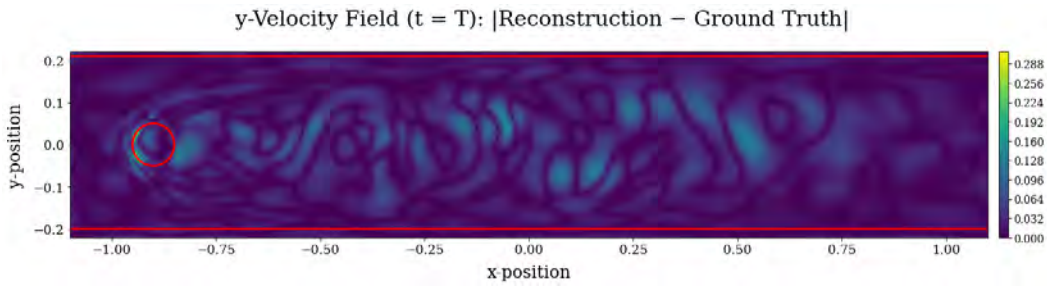

Fig. 5.9 Absolute discrepancy between the reconstruction and ground truth y-velocity fields from Figure 5.8 ($t = T$).

Table 5.1 Effect of residual weight r on reconstruction accuracies of the flow field, for 2D unsteady flow past a cylinder ($Re \approx 100$). For all cases, the number of hidden layers is 30, and the number of nodes per layer is 225. A residual weight of 0% means the neural network is no longer physics-informed. 'Exp Variation' refers to variation of residual weight as per Equation 5.1.

Residual Weight (%)	ACC_x (%)	ACC_y (%)
0	93.43	90.50
1	93.83	90.59
3	92.61	90.19
5	91.53	88.40
Exp Variation	90.93	88.32

Table 5.2 Effect of number of hidden layers on reconstruction accuracies of the flow field, as well as PINN training and convergence times, for 2D unsteady flow past a cylinder ($Re \approx 100$). For all cases, the number of nodes per layer is 225, and the residual weight is set to 1%.

Hidden Layers	ACC_x (%)	ACC_y (%)	$Time_{Train}$ (Min)	$Time_{Converge}$ (Min)
30	93.83	90.59	88.0	35.2
36	94.01	91.29	109.3	14.6
39	92.67	88.65	105.1	42.0
45	94.85	93.13	140.9	18.8

Table 5.3 Effect of number of nodes per layer on reconstruction accuracies of the flow field, as well as PINN training and convergence times, for 2D unsteady flow past a cylinder ($Re \approx 100$). For all cases, the number of hidden layers is 30, and the residual weight is set to 1%.

Nodes per Layer	ACC_x (%)	ACC_y (%)	$Time_{Train}$ (Min)	$Time_{Converge}$ (Min)
225	93.83	90.59	88.0	35.2
250	94.24	92.09	89.5	53.7
275	94.61	92.84	92.8	12.4
300	91.22	87.87	89.8	53.9
325	95.16	93.01	104.8	14.0

5.4 Discussion

Figures 5.2, 5.4, 5.6, and 5.8 indicate faithful flow field reconstructions when the unsteady flow data is assimilated over one period. The detail of the vortex street appears to be generally retained when comparing the reconstructions with the ground truths, especially in the vicinity of the cylinder. However, Figures 5.3, 5.5, 5.7, and 5.9 indicate that discrepancies between ground truth and reconstruction generally lie along the vortex street and at the cylinder boundary. Including BC residuals for the cylinder in the loss may reduce the discrepancy. Nevertheless, Tables 5.1, 5.2, and 5.3 indicate high reconstruction accuracies ranging from 88% to 95%.

Table 5.1 indicates that a PINN with a residual weighing of 1% is superior to using a normal neural network with zero residual weighting. The incorporation of physics clearly helps prevent the neural network from violating mass and momentum conservation.

Tables 5.2 and 5.3 indicate higher reconstruction accuracies with higher number of hidden layers and nodes per layer, respectively. The PINN total training time (i.e. time to train over 15000 epochs) is naturally correlated with the PINN size, being as high as 141 minutes and as low as 88 minutes. The PINN convergence time, on the other hand, seems to vary randomly and with no correlation to PINN size. This is possibly due to different initialisations of PINNs with different number of layers or nodes per layer. Despite this, the highest convergence time observed is 53.7 minutes, and the lowest is 14 minutes[3]. This demonstrates that the PINN approach, which makes use of a single noisy MRV image, is as quick and in some cases even quicker than the traditional method of taking thousands of MRV scans and averaging them.

[3]The recorded times are obtained by running the PINNs on Google Colab Pro, which gives access to T4, P100, and/or K80 Graphics Processing Units (GPUs) during any given session, with 24 GB RAM availability.

Chapter 6

Conclusions

In this project we explore the use of a PINN for de-noising MRV images, whereby the neural network becomes physics-informed by accounting for the residuals of fluid dynamic PDEs when computing the loss function. The shape of the flow is then inferred by computing streamlines. Chapter 1 introduces the project and outlines possible contributions. Chapter 2 then explains in more detail the research methodology: the derivation of the physics-informed loss function, the justification of various aspects of PINN training, the way of generating training data, and the success metrics. Thereafter, Chapters 3, 4, and 5 examine the approach defined in Chapter 2 on various flow cases ranging in physical complexity.

Over the course of Chapters 3, 4, and 5, it is demonstrated that reasonably high flow field reconstruction accuracies can be obtained via PINNs that only make use of a single noisy MRV image. It is demonstrated as well that this can be done without prior knowledge of flow geometry as in previous studies, and that instead this geometry can be inferred from the flow field at reasonably high accuracies. The PINN method is applied on flows ranging from 2D to 3D, steady to unsteady, viscous to inertial, within a small project duration, which indicates the ease of scalability of the method. The speed of PINN de-noising in terms of convergence time is also shown to rival that of traditional MRV imaging techniques.

Even though the incorporation of physical residuals into the PINN loss function results in improvements in reconstruction accuracies compared to a standard neural network, it cannot be said that the improvements are observed consistently. For example, improvements are not seen simultaneously in all of ACC_x, ACC_y, ACC_z, ACC_{Shape}, and any improvements are relatively small. Reconstruction accuracies also notably degrade going from synthetic data to experimental data, and discrepancies are notably high at boundaries and outlets. Also, the

method of shape inference using streamlines does not appear to be an optimal one for high *Re* flows.

The work conducted in this project does nevertheless serve as a promising starting point for further research into the use of PINNs for MRV de-noising, and, as such, potential improvements to the work in this project can be outlined. For example, in future research into MRV de-noising via PINNs, it is sensible to explore a neural network architecture that allows de-noising of velocity field and inference of flow boundary shape simultaneously. It is also useful to find a method of estimating PINN uncertainty, such as using Laplace approximation [Ritter et al., 2018] or anchored neural network ensembling [Pearce et al., 2020]. Moreover, a way of automating neural network hyperparameter tuning, via random grid search or perhaps Bayesian optimisation [Snoek et al., 2012] should also be implemented. IC and BC residuals should also be accounted for when computing the loss.

Long term, this PINN approach should be applied on more complicated flow regimes, such as 3D turbulent flow, or flows where the boundaries are compliant.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Anderson, J. D. and Wendt, J. (1995). *Computational fluid dynamics*, volume 206. Springer.

Anil, R., Gupta, V., Koren, T., and Singer, Y. (2019). Memory-efficient adaptive optimization. *arXiv preprint arXiv:1901.11150*.

Baker, N., Alexander, F., Bremer, T., Hagberg, A., Kevrekidis, Y., Najm, H., Parashar, M., Patra, A., Sethian, J., Wild, S., et al. (2019). Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. Technical report, USDOE Office of Science (SC), Washington, DC (United States).

Batchelor, C. K. and Batchelor, G. (2000). *An introduction to fluid dynamics*. Cambridge university press.

Blythe, T., Sederman, A., Mitchell, J., Stitt, E., York, A., and Gladden, L. (2015). Characterising the rheology of non-newtonian fluids using pfg-nmr and cumulant analysis. *Journal of Magnetic Resonance*, 255:122–131.

Brennen, C. E. (2006). An internet book on fluid dynamics. *INDEX for INTERNET BOOK ON FLUID DYNAMICS*.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Ferziger, J. H., Perić, M., and Street, R. L. (2002). *Computational methods for fluid dynamics*, volume 3. Springer.

Fukushima, E. (1999). Nuclear magnetic resonance as a tool to study flow. *Annual review of fluid mechanics*, 31(1):95–123.

Funke, S. W., Nordaas, M., Evju, Ø., Alnæs, M. S., and Mardal, K. A. (2019). Variational data assimilation for transient blood flow simulations: Cerebral aneurysms as an illustrative example. *International journal for numerical methods in biomedical engineering*, 35(1):e3152.

Gladden, L. F. and Sederman, A. J. (2017). Magnetic resonance imaging and velocity mapping in chemical engineering applications. *Annual review of chemical and biomolecular engineering*, 8:227–247.

Glorot, X. and Bengio, Y. (2010a). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Glorot, X. and Bengio, Y. (2010b). Xavier initialization. *Journal of Machine Learning Research, 2010b. ISSN*, 15324435.

Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., and Knoll, F. (2018). Learning a variational network for reconstruction of accelerated mri data. *Magnetic resonance in medicine*, 79(6):3055–3071.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koltukluoğlu, T. S. (2019). Fourier spectral dynamic data assimilation: interlacing cfd with 4d flow mri. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 741–749. Springer.

Koltukluoğlu, T. S. and Blanco, P. J. (2018). Boundary control in computational haemodynamics. *Journal of Fluid Mechanics*, 847:329–364.

Kontogiannis, A. and Juniper, M. (2020). Inverse problems in magnetic resonance velocimetry: shape, velocity and boundary condition inference. In *APS Division of Fluid Dynamics Meeting Abstracts*, pages T07–012.

Kumar, S. K. (2017). On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021). Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228.

Lustig, M., Donoho, D., and Pauly, J. M. (2007). Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195.

Markl, M., Frydrychowicz, A., Kozerke, S., Hope, M., and Wieben, O. (2012). 4d flow mri. *Journal of Magnetic Resonance Imaging*, 36(5):1015–1036.

Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Pearce, T., Leibfried, F., and Brintrup, A. (2020). Uncertainty in neural networks: Approximately bayesian ensembling. In *International conference on artificial intelligence and statistics*, pages 234–244. PMLR.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7:1.

Ritter, H., Botev, A., and Barber, D. (2018). A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

van de Meent, J.-W., Sederman, A. J., Gladden, L. F., and Goldstein, R. E. (2009). Measurement of cytoplasmic streaming in chara corallina by magnetic resonance velocimetry. *arXiv preprint arXiv:0904.2707*.

Vishnevskiy, V., Walheim, J., and Kozerke, S. (2020). Deep variational network for rapid 4d flow mri reconstruction. *Nature Machine Intelligence*, 2(4):228–235.

Weishaupt, D., Köchli, V. D., and Marincek, B. (2003). Factors affecting the signal-to-noise ratio. In *How does MRI work?*, pages 31–42. Springer.

Zhang, F. (2006). *The Schur complement and its applications*, volume 4. Springer Science & Business Media.