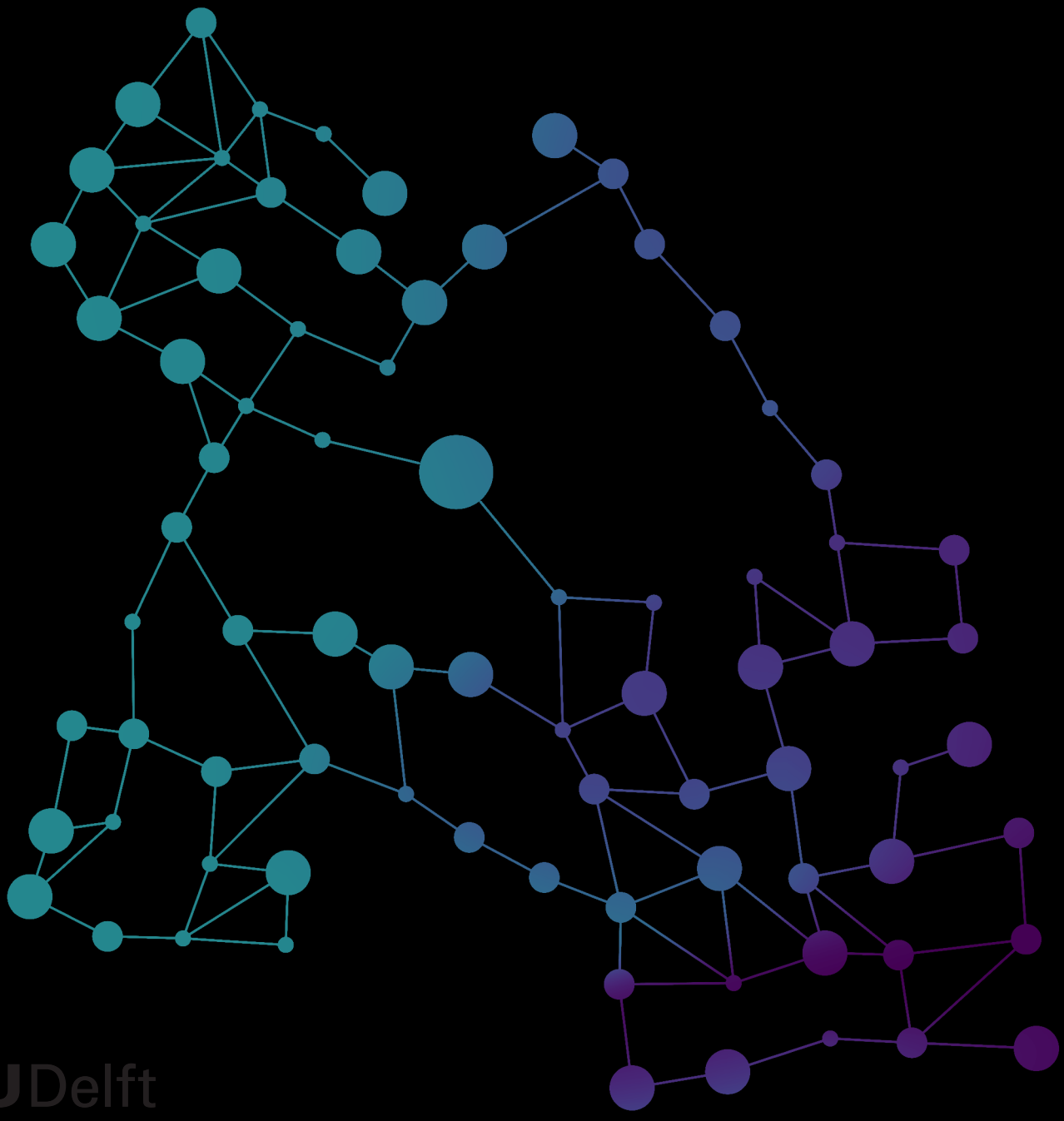
Velocity Reconstruction in Pool Fires using Physics-Informed Machine Learning

Dirk de Boer



TU Delft

Velocity Reconstruction in Pool Fires using Physics-Informed Machine Learning

by

Dirk de Boer

to obtain the degree of Master of Science at the Delft University of Technology, to be defended publicly on Tuesday December 13, 2022 at 14:00.

Student number:	4562771
Project duration:	January - December 2022
Thesis supervisors:	Dr. N.A.K. Doan & Dr. M. P. Sitte
Thesis committee:	Dr.ir. M.I. Gerritsma & Dr. W. Yu
Submission date:	November 29, 2022

TU Delft

Contents

Acknowledgement

I would like to take the opportunity to thank Anh Khoa Doan and Philip Sitte for their valuable support and supervision. In addition, I thank my family and friends for being there for me in the process. I would like to wish you a pleasant time reading this work.

List of Figures

List of Tables

List of Symbols and Abbreviations

Symbols

α	Ratio to tailor the impact of measurement and physics loss separately
α	Thermal diffusivity
ϕ_m	Provided flow states
$\breve{\xi}_P$	Proper Orthogonal Decomposition in terms of basis functions
$\dot{\omega}$	Reaction rate
ϵ_m	Measurement loss
ϵ_m	Physics loss
γ	Learning rate
$\hat{\mathbf{s}}$	Optimal weights for the library elements
$\hat{\mathbf{x}}$	Approximate state vector
$\hat{\mathbf{y}}$	Approximate measurement vector
$\mathbf{a}$	Weights for the basis functions
$\mathbf{b_i}$	Bias vector
$\mathbf{s}$	Weights for the library elements
$\mathbf{W_i}$	Matrix of weights
$\mathbf{x_i}$	Vector of outputs of layer i of neural network
$\mathbf{x}$	State vector
$\mathbf{y}$	Measurement vector
$\mathcal{L}$	Total loss function
$\mathcal{R}$	Residual(s) of the governing equation(s)
μ	Mean of a signal
ϕ_i	Basis functions
Ψ	Matrix containing library elements
Ψ_j	Item j from library elements
ρ	Density
σ	Activation function
τ	Molecular stress tensor
$\tilde{\phi}_m$	Network output for provided states
$\tilde{\Phi}$	All output states

a	Pool fire radius
b	Bias
C	Matrix to transform from state vector to measurement vector
D	Molecular diffusivity
g	Gravitational field strength
h	Enthalpy
N	Total number of data points
N_c	Number of collocation points
N_m	Number of measurement points
P	Amount of basis functions used
P	Signal power
p	Pressure
R	Specific gas constant depending on composition of considered gas
s	Signal
T	Temperature
t	Time
u	First component of the velocity vector
v	Second component of the velocity vector
w_i	Weight
x	First component of space vector
Y	Species mass fraction
y	Second component of space vector
g	Vector direction of steepest descent
p	Parameters in optimization domain (weights, biases)

Abbreviations

BC	Boundary Condition
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy
CNN	Convolutional Neural Network
DE	Differential Equation
GPOD	Gappy Proper Orthogonal Decomposition
HFM	Hidden Fluid Mechanics
IC	Initial Condition
ML	Machine Learning

NN	Neural Network
ODE	Ordinary Differential Equation
OH-PLIF	Hydroxyl Planar Laser Induced Fluorescence
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
PIV	Particle Image Velocimetry
POD	Proper Orthogonal Decomposition
PTV	Particle Tracking Velocimetry
Tomo-BOS	Tomographic Background-Oriented Schlieren

Abstract

In many flow experiments it is complex to measure all flow states of interest, leading to the need for a method to retrieve unmeasured flow states from measured ones. This work focuses on Hidden Fluid Mechanics (HFM), which refers to a Physics-Informed Neural Network (PINN) able to incorporate the Navier-Stokes (NS) equations into the loss function of the Neural Network (NN) for the purpose of reconstructing flow fields.

As HFM is a recent framework which has not been applied to many flows yet, it is unknown if HFM is capable of reconstructing flow fields in chemically reacting flows where strong gradients are present. For this reason, the performance of HFM is investigated here using data sets from Direct Numerical Simulations (DNS) of an axisymmetric pool fire in cylindrical coordinates, for which the velocity fields are purposely removed and attempted to be recovered from the density, pressure and temperature fields. The implementation has been performed making use of TensorFlow in Python, where both steady and unsteady pool fires are considered.

Pool fires are simplified representations of real-life accidental and forest fires, illustrating the importance of their research. If the critical pool diameter is not exceeded, a steady flow occurs. However, if surpassed, the flow exhibits an unsteady flickering behaviour known as puffing.

The main finding is that both in the steady and unsteady pool fires, the PINN is capable of reconstructing the most prominent features of the velocity fields when density, pressure and temperature are provided, even though strong gradients are usually underestimated by the PINN, causing wrongly predicted extreme values. This implies that the method can be used mainly for qualitative purposes, as the exact values are not always captured. It is found that the accuracy is lower in the unsteady flow compared to the steady flow. Furthermore, the axial velocity field is generally reconstructed to a higher accuracy than the radial velocity field.

It is found that the accuracy decreases when increasing the spacing between provided data points, from which implications on possible applications to real-life measured data can be drawn, as not all measurement equipment might reach the required spatial resolution. Investigating the effect of adding artificial white noise to mimic measurement noise, it has been found that the method is robust to noise, but less so for the steady case than for the unsteady flow case. Lastly, the effect of removing the density and/or pressure from the provided flow states is investigated, where it is found to have a detrimental effect on the accuracy except if physical assumptions are given to the PINN (i.e. the ideal gas law). Using the right assumptions, it is found that for the steady case providing only the temperature field is deemed enough to reach a relatively similar accuracy compared to not removing any flow states, whereas for the unsteady case at least the temperature and pressure fields need to be provided. The current research shows the capabilities but also the limitations of the HFM framework in case of steady and unsteady flow with chemical reactions and strong gradients, creating opportunities for further research of HFM in other flow cases and for potentially applying the framework to real experimental data of fires or other flows.

1 Introduction

Measurements in flows are important because they allow for determining forces, thermodynamic states across the flow and many more parameters, which can help answer questions about flow phenomena and the performance of scaled wind tunnel models in certain flows. Examples entail flow velocity visualization to understand the vortical structures appearing behind bluff bodies and the shape optimization of an object to produce the least amount of drag or heat, amongst others. In the most optimal situation one would acquire all flow states everywhere in the flow densely, as from that all forces and other important flow parameters and phenomena could be found and analyzed. However, it is almost impossible and very expensive to measure all flow states. Furthermore, in some cases it is even impossible to measure one or more particular flow states due to inaccessibility because of physical obstruction or due to a too rough physical environment. This motivates the necessity for reconstruction methods, as this would allow to reconstruct the states that could not have been measured directly.

There are many different flow reconstruction methods many of which a lot rely on Proper Orthogonal Decomposition (POD), which is used to generate a library of candidate flow states from which some reconstructions are performed by projecting the limited partial available measurement [1, 2]. The major disadvantage of this and most other reconstruction methods is that at least some measurements need to be done to create this library of candidate full flow states. In some experiments or real-life scenarios, it might not be possible to measure the flow state of interest at all, meaning that methods that require this can not be used. This requirement is not present for the use of a recent reconstruction method based on Machine Learning (ML), denominated Hidden Fluid Mechanics (HFM).

The name HFM refers to the fact that flow states can be recovered despite the inability or incapability to measure them. HFM makes use of a Neural Network (NN) with hidden layers [3–5]. In opposition to the 'black box' methods in ML, HFM implements the governing flow equations in a NN. This is referred to as a Physics-Informed NN (PINN), where its name can be explained by the fact that the network is informed by one or more physical/mathematical laws. HFM is still a recent method and has not been tested in many different types of flows. It is thus important to asses how well this method is able to retrieve flow states in different flow cases, as it adds to the knowledge about when HFM works sufficiently and what its limitations are. Specifically, the focus of this work is on a reacting flow of practical interest: a pool fire. This kind of application constitutes a large challenge for HFM given that it will have to deal with chemical reactions and large density and temperature gradients for which HFM has not been demonstrated to work yet.

A pool fire can be described as a small-scale diffusion flame and is a simplified representation of real-life fire scenarios [6]. It is known that in certain circumstances pool fires behave in an unsteady and unstable way. In this motion, vortices provide fresh air to the flame in a periodic manner, initiating the flickering effect called puffing [7]. Because a pool fire is such a general description for a fire, it is an important part of fire and fire safety research. For this reason, if HFM is capable of reconstructing unmeasured states in flow fields around pool fires, it would possibly aid future fire safety research. This is particularly relevant as measurements in large fire experiments are often very limited.

To be able to understand the rationale through this thesis, background theory about pool fires and ML is given first in chapter 2. After that, the state of the art of reconstruction methods and more detail on HFM is given in chapter 3. Then, the scientific gaps that follow from the logical structure are stated and

the research questions following from this are proposed in chapter 4. After this, the method to tackle the investigation of the research questions is explained in chapter 5. Furthermore, the steady and unsteady pool fire data to which HFM is to be applied are introduced and it is mentioned how this data is obtained. The results and outcomes from the method are shown and discussed in chapter 6, where the reconstruction accuracy of the velocity fields from the density, pressure and temperature fields is assessed in different scenarios. After knowing the results, the main conclusions and the answers to the research questions are stated in chapter 7. Finally, recommendations for future research are given in chapter 8.

2

Background Theory

To be able to understand the concepts in the state of the art chapter and the line of reasoning of this work, it is first necessary to present some background knowledge about NNs, HFM, and pool fires. For this reason, background information about NNs and HFM is given in section 2.1, whereas that on pool fires is given in section 2.2.

2.1. Neural Networks & Hidden Fluid Mechanics

This section consists of two subsections. Firstly, subsection 2.1.1 provides the basics of artificial neurons and NNs. Finally, subsection 2.1.2 elaborates on the background knowledge of the advanced topics, namely PINNs and HFM.

2.1.1. Basics of the Artificial Neuron & Neural Networks

Originally, the concept of AI was to mimic and replace human (intelligent) thinking capabilities: the biological understanding and interpretation of human brains was applied in a computational setting. In Figure 2.1, one can see the biological and computational interpretation of a neuron. Multiple inputs (or dendrites) end at a neuron, where the impulse is carried further as output (to an axon). In brain cells, the output impulse strength is known to be dependent on all the input strengths, where some have stronger effects than others, and some have a positive effect whereas others cause the impulse to decrease in strength [8]. This is a very complex phenomenon, but is represented computationally by multiplications of the inputs with weights w_i and adding a bias b in the end. Usually, all input contributions have to sum up to a value that exceeds a threshold before the neuron starts transmitting an output pulse through the axon. Also, this output pulse has a certain range of strengths which it can have (electrical pulses in axons cannot exceed certain values), meaning that the sum of all inputs must be transformed to meet these two conditions: an activation function is added in the computational model to mimic this. Furthermore, the activation function allows for non-linearity in the mapping function between inputs and outputs of a NN.

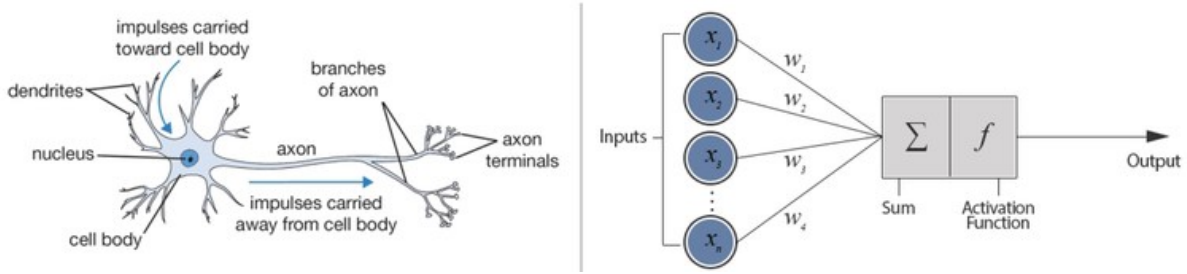


Figure 2.1: Representations of a neuron from the biological and AI perspective [9]

If many neurons are connected in multiple rows and layers, a NN is created. The operations performed between two layers of a NN can be written in a short form seen in Equation 2.1. In this, the bold

faces refer to vectors or matrices, $\mathbf{x_i}$ refers to the output vector of the layer i and $\mathbf{x_{i+1}}$ to the output vector of the next layer, $\mathbf{b_i}$ to the bias vector added, $\mathbf{W_i}$ to the matrix of weights and σ to the activation function which is applied to all the vector terms.

$$\mathbf{x_{i+1}} = \sigma(\mathbf{W_i x_i} + \mathbf{b_i}) \tag{2.1}$$

A visual representation of a NN with one input, one output and three hidden layers with width two can be seen in Figure 2.2. Note that the NN is very small for the purpose of comprehensibility, whereas actual NNs generally contain many more layers and neurons per layer. Every neuron sends its (output) value to the next layer of neurons through the connections. Every connection has a label corresponding to the weight associated with that connection. The biases are incorporated by multiplying unit value neurons (the three smaller neurons on the top) with the correct bias values. At every neuron, the weighted values along all input connections are summed after which the activation function is applied (see Equation 2.1) to get the neuron's output value.

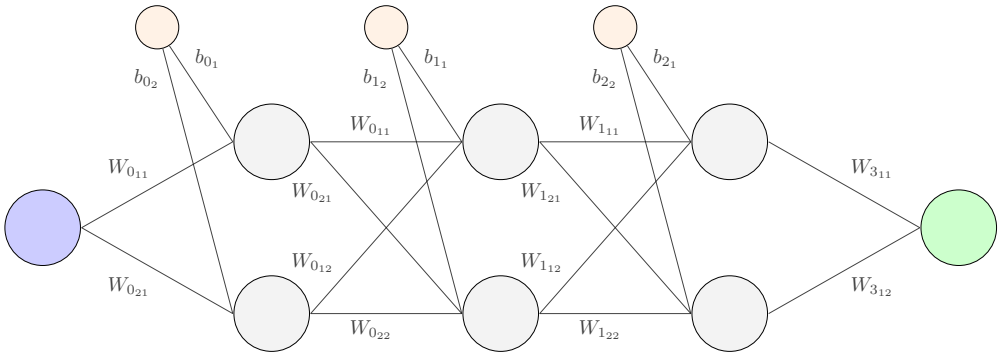


Figure 2.2: Representations of an arbitrary NN with four inputs, two outputs and two hidden layers with width six

During training of a NN, the weights and biases are updated to optimize the network to do a certain task. This is usually done using a loss function which has to be minimized. The loss function is a measure of how erroneous the NN is predicting, being related to the difference between the prediction and the actual outcome it should have. To be able to effectively decrease the loss function, the most common approach with NNs is to use gradient-based optimizers which require to find the direction of the steepest descent of the loss function. For this, the gradients of the loss with respect to the weights and biases have to be known. The process that handles this is called backpropagation and it works similarly to the method of automatic differentiation introduced subsequently. The weights and biases can be updated with a certain learning rate to change the loss function in the direction of the minimum. A very simple representation of an updating procedure can be seen in Equation 2.2, where more elaborate information can be found in literature [10]. In this, $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ refer to the vectors of parameters (weights and biases in the NN) at the current and next iteration, respectively. Furthermore, γ refers to the learning rate and $\mathbf{g}_i$ refers to the vector in the direction of descent.

$$\mathbf{p}_{i+1} = \mathbf{p}_i - \gamma \mathbf{g}_i \tag{2.2}$$

The lower the loss function, the better the network's prediction. There is one disadvantage, namely that the network is trained on the same data over and over, meaning that it can notice patterns that only appear in the training data and that are not a general feature of such a problem. This phenomenon is called overfitting and it refers to the fact that the network evolves a bias and fits too strongly to the training data. A way to check if overfitting occurs is to compare the loss function/error when using the NN on new data and the training data. If the loss is (much) greater for the new data, overfitting occurred.

There are numerous possible activation functions, many of which serve a certain purpose. One of the simplest activation function is Rectifier Linear Units (ReLU), which projects negative inputs to a zero output and leaves positive inputs unaffected. This leads to a discontinuous derivative at zero, possibly causing issues. For this reason, there are alternatives that also project the input to a positive value which do not have this discontinuity in slope. Many times, it is not favourable that certain neurons

output very high or low (negative) values. Thus, there are also activation function that project the output between zero and one, or minus one and one, thereby limiting the output range.

The application of NNs to problems is reasoned by the universal approximation theorem, stating that a NN with at least one hidden layer can approximate any function from a finite dimensional space to another finite dimensional space up to any desired degree of accuracy [11]. In other words, even a NN with only one layer can represent any function if enough neurons are added in that layer. However, it is not known how many neurons are necessary in this one layer to be able to represent functions of a certain complexity. Also, it does not give any indication about what the most efficient way of setting up a NN is, if the NN should have a higher width (neurons per layer) or depth (layers) in general, or how these parameters affect the total number of neurons needed to represent a function of certain complexity. Many of these questions are yet to be answered and depend on the application. For now, no general 'best' NN shape or activation function is known and the problem of NN architecture partly remains one of trial and error.

Most applications of AI are 'black box' methods: they are based on large data sets for which certain patterns can be discovered during training, after which it is possible to apply the network to new data to cluster, label or for other applications. Black box methods are generally popular as they do not involve the complex procedure of encompassing physical laws into the NN. In many different AI applications, black box methods are used as the prediction does not have to obey a certain physical law or rule. For example, in case of image recognition of cats and dogs, there are no built-in rules or laws and the prediction is fully based on images it has trained on. The disadvantage of black box methods is that they ignore physical laws and are thus fully data based. This also means that these methods can only be applied in cases where enough data is available and only in particular problems. As in fluid flow the solution has to adhere to a set of physical conservation laws, NNs made to solve fluid flow problems should incorporate these conservation laws. These NNs are called Physics-Informed NNs (PINNs), as they encompass the physical laws in the NN. In the following subsection, the concepts and functioning of PINNs and HFM are depicted comprehensively.

2.1.2. Physics-Informed Neural Networks & Hidden Fluid Mechanics

The main concept of a PINN is to embed physical laws in the NN, which is not a simple task. The main idea is to add the residuals of the governing equations into the loss function. One should also note that the concept of this type of NN is different than the ones used in other applications: the NN is not trained on other data before being applied to the actual problem to be solved. The NN inputs and outputs depend on the problem considered, but in fluid flow the inputs would be the temporal and/or spatial coordinates and the outputs would consist of the values of the flow states. The concept is to match the exact solution as well as possible by training the network on the data that is available and the governing equations. This is not straightforward, as the governing equations often contain temporal and spatial derivatives, which are not part of the output of the network. Fortunately, this issue can be dealt with using automatic differentiation, which is a concept that chains all partial derivatives of the neurons together to determine the temporal and spatial derivatives [12]. This is known to be neither numerical nor symbolic differentiation. The working of this is based on the evaluation trace, which is a way of writing a large function into smaller functions, allowing to split the large function the NN represents into smaller sub-functions. As the inputs for a neuron are the outputs of the previous column of neurons, the chain rule can be used to couple all the functions' derivatives. This would be symbolic differentiation and cause an extremely long expression to be evaluated. Instead of doing this, the derivative of every neuron is found symbolically and then evaluated at the current settings (at current inputs), after which all can be chained together to find an approximation of the full derivative. Applying this, the residuals of the physical laws can be found at many different arbitrarily defined points in the space-time domain, called collocation points.

PINNs are applied to many different physics problems, in which the NN usually adheres to a type of conservation law. The most basic examples are cases in which Differential Equations (DEs) are solved using a PINN [13]. For example, PINNs are applied to Ordinary DEs (ODEs) [14], the linear Poisson equation, heat conduction and heat equation problems with corresponding Initial Conditions (ICs) and Boundary Conditions (BCs) [15, 16]. PINNs are even applied to the Navier-Stokes (NS) equations in the case of incompressible cavity flow [14]. All these problems are forward problems, in which the BCs and ICs are known and one wants to find the solution. However, PINNs can also be used in reconstruction problems where only part of the solution is known (for example measured in

an experiment) and the rest of the solution needs to be retrieved from this partial information, without information about BCs and ICs. In these specific inverse problems, which are referred to as HFM, 'hidden' flow states are aimed to be retrieved from other measured ones [3–5]. In fact, these inverse methods are the main use of PINNs in fluid flows, as forward methods (where BCs and ICs are known and the solution is to be found) are more efficiently solved by traditional methods such as CFD [17]. As the reconstruction problem can be ill-posed, it is generally not possible to solve it with traditional methods, as it would require adjusting BCs and ICs until the measured part of the solution is met as close as possible, which means the rest of the solution is close to the exact one, requiring to rerun the forward method (for example CFD) many times. Although PINNs can be more efficient in these situations, the question of uniqueness is still an issue, as there is no proof yet to show that in these cases there is a unique solution.

A typical HFM network would look like the one shown in Figure 2.3. The inputs are the coordinates in space and time and can be seen in blue. The outputs are the velocity, density, pressure and temperature, shown in green. The hidden layers of the NN are grey and the additional part on the right that allows to find the derivatives for determining the residuals is shown in yellow. The additional layers on the right are a simplified version of what happens in reality, namely the automatic differentiation steps. The loss terms are shown in red, together adding up to the total loss function. The first loss, ϵ_m, is due to the difference between the outcome and actual measured values of the known flow states. In the specific network shown, it is the difference between the predicted density, pressure and temperature and the measured ones. The second loss, ϵ_p is due to the nonzero residuals in the governing equations. The governing equations could be the NS equations or a variation of these. The combination of both loss terms gives the total loss, $\mathcal{L}$. Note that the dots in the hidden layers indicate that there are numerous layers and neurons per layer, not shown for the purpose of clarity.

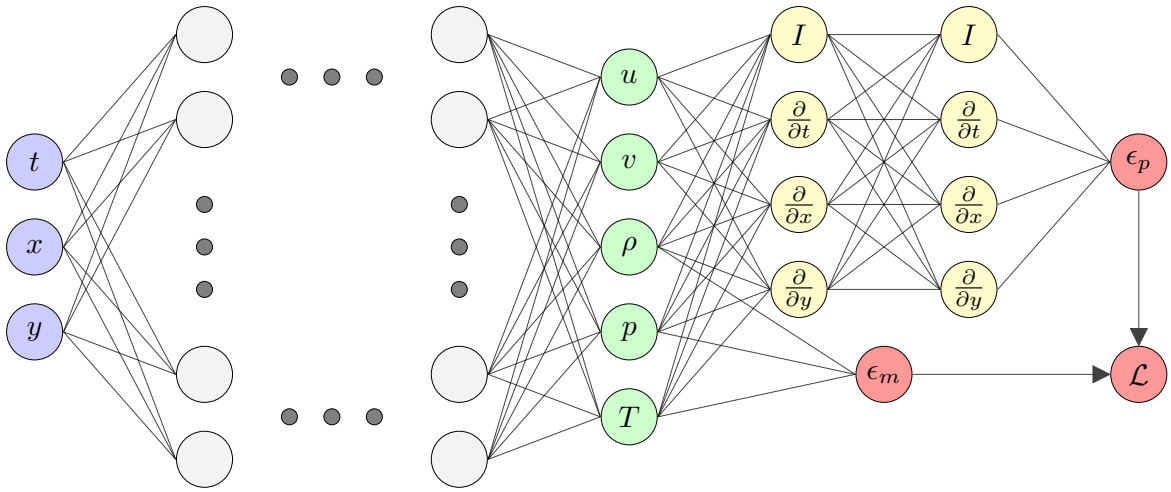


Figure 2.3: Representation of a typical HFM network

The full loss written in mathematical form can be seen in Equation 2.3. In this expression, $\mathcal{L}$ stands for the total loss function, N_m represents the number of measurement points, N_c the number of collocation points, $\tilde{\phi}_m$ stands for the network output states for the parameters that were provided as measurements, whereas ϕ_m are those provided measured values, $\widetilde{\Phi}$ represents all output states from the NN and $\mathcal{R}$ means the residuals from the governing laws. As seen, both the error between predicted states and target states and the nonzero residuals build the total loss function. It should be noted that the differences are squared and normalized by the amount of measurement and collocation points. The α in front of the measurement loss term allows to tailor the impact of the separate loss terms. When the loss reaches below a threshold, the problem is considered to be solved and the prediction $\widetilde{\Phi}$ is to be considered sufficiently close to the exact solution.

$$\mathcal{L} = \alpha \underbrace{\frac{1}{N_m}\sum_{n=1}^{N_m}\left|\phi_m\left(x_m^n, t_m^n\right) - \tilde{\phi}_m\left(x_m^n, t_m^n\right)\right|^2}_{\epsilon_m} + \underbrace{\frac{1}{N_c}\sum_{i=1}^{N_c}\left|\mathcal{R}\left(\tilde{\Phi}\left(x_c^i, t_c^i\right)\right)\right|^2}_{\epsilon_p} \tag{2.3}$$

2.2. Pool Fires

Since pool fires are considered as a reasonable simplification of many fires, such as forest fires or leakage fires [6], much research on pool fires already exists. Pool fires are small-scale diffusion flames which in most conditions show a flickering behaviour called puffing. The dynamics of this mode is determined by the buoyant instability that occurs around the flame. The flickering behaviour of the flame can be seen schematically in Figure 2.4 and 2.5 [7, 18].

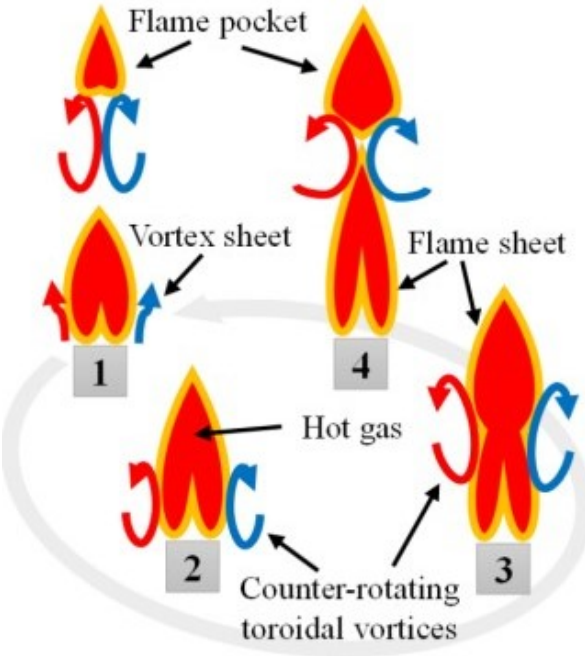


Figure 2.4: Visualization of puffing cycle [18]

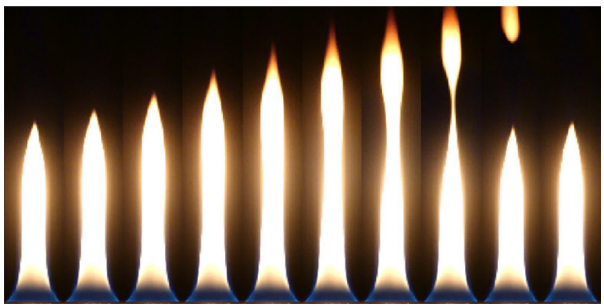


Figure 2.5: Visualization of puffing cycle from experiment [7]

The puffing phenomenon is a cycle, where the flame first elongates (1 - 2), after which a vortex ring is generated (2 - 3). This ring then rolls up (3 - 4), separating part of the flame from the burner and decreasing the size of the original flame (4 - 1), starting the cycle over again. In some cases, it does not happen that the top part of the flame separates. Flame pinch-off, when a small flame pocket separates from the original flame, only occurs under certain conditions. One can explain the growth of the toroidal vortex by the difference in density across the domain. Near the flame, the temperature

is higher, causing the density to decrease, thus causing the air and gas to flow upward. Away from the flame, the temperature and thus also the density are close to the ambient values, meaning there is neither a flow upward nor downward. This means that there must be a velocity gradient from inside the flame to away from the flame, which is by the vorticity equation known to be a source for the growth of vortices. As there is a certain time associated with the cycle, there is a frequency at which the flame flickers, called the puffing frequency. The increasing and decreasing size of the flame can be explained by the fact that the influx of fresh air (thus oxygen) depends on the state (1, 2, 3 or 4) of the vortex ring.

There are certain conditions to be met for the puffing to occur. Using analytical, computational and experimental methods, it has for example been determined that the flickering behaviour appears only above a critical Rayleigh number which depends on the configuration [7]. The Rayleigh number is the ratio between the gravitational and viscous forces, indicating that a higher Rayleigh number means relatively stronger gravitational forces. As a difference in density is at the base of the instability, it is logical that a stronger gravitational pull would start or intensify the puffing behaviour. Furthermore, the occurrence of flame pinch-off also requires certain conditions: only below the critical Froude number, flame pinch-off occurs [19]. As the Froude number is the ratio between inertia and gravitational forces, a higher Froude number means more inertia relative to gravity. It is logical that high Froude numbers would stop flame pinch-off from happening, as the vortices are weaker and the flame is not pushed up as much in that situation.

3

State of the art

In this chapter, the state of the art for different research fields related to this thesis is presented. Firstly, research on data reconstruction methods is discussed in section 3.1. Then, the focus shifts to HFM, because existing reconstruction methods need at least some (other) measurements from the flow state to reconstruct that particular flow state in a new case, which is in some cases not possible. The state of the art on HFM can be found in section 3.2. As HFM is a relatively recent method, it is to be tested on different flow cases to be able to determine how promising it is for future use. Pool fires are important flows for which HFM has not been shown to work yet. As pool fire flows are complex because of ongoing reactions, it is interesting to see if HFM is still able to retrieve unmeasured flow states in this type of flow.

3.1. Data reconstruction

One of the most prominent reconstruction methods relies on Proper Orthogonal Decomposition (POD). POD is a method based on data analysis which aims to find an approximation of a solution field using a linear combination of basis functions [20]. Both the basis functions (eigenmodes) and weights (eigenvalues) are found from measurements performed on flow states. The idea is that with enough data, the most optimal basis functions are known and thus the full (dense) solution can be approximated using these basis functions with proper weights. With this, sparse measurements on for example the velocity can be used to approximate the full velocity field, also in regions where no measurements have been performed. One way of writing the solution in the form of a combination of basis functions can be seen in Equation 3.1 [1].

$$\breve{\xi}_P(\mathbf{x}, t_k) = \sum_{i=1}^{P} a_i^k \phi_i(\mathbf{x}) \tag{3.1}$$

The POD solution is written as time-dependent weights multiplied with spatial basis functions (independent of time). When having data on the flow, finding the modes and the coefficients is an eigenvalue problem: the eigenvectors are the system's modes and the eigenvalues are the energy related to the modes (the coefficients). An advantage of the POD method is that it leads to a system which has only one optimal combination of basis functions and coefficients. One of the disadvantages entails obtaining large matrix systems, for which simplifications have to be made to be able to solve them with an acceptable amount of computational effort. Another issue is the matter in which POD treats gaps in the data as real information, affecting the eigenmodes and eigenvalues found. Gap filling methods such as Gappy-POD (GPOD) are introduced to resolve this issue [1]. These methods are based on guessing an initial value in the domain gaps and updating this based on the POD approximation at every iteration. Lastly, POD and GPOD function solely by converting sparse flow measurements to dense flow information, or through the gap-filling function in case gaps are present in the data. This means that these methods are not capable of reconstructing flow states for which no measurement has been done at all.

Another noteworthy manner for data reconstruction concerns sparse representation [2]. In this method, a library of flow field data is used to recover new full flow solutions from sparse data. In this method, it is assumed that the measurement vector $\mathbf{y}$ can be written linearly as shown in Equation 3.2.

$$\mathbf{y} = C\mathbf{x} \tag{3.2}$$

Furthermore, it is assumed that there exists a vector $\mathbf{s}$ with weights such that the state vector $\mathbf{x}$ can be estimated as a linear combination with the weights in $\mathbf{s}$ of elements Ψ_j found in the library, as stated in Equation 3.3.

$$\hat{\mathbf{x}} = \Psi\mathbf{s} \tag{3.3}$$

Combining the two, one can write the approximation of the measurement vector $\mathbf{y}$ as a function of vector $\mathbf{s}$ with still unknown weights, as seen in Equation 3.4.

$$\hat{\mathbf{y}} = C\Psi\mathbf{s} \tag{3.4}$$

In case the correct weights in $\mathbf{s}$ are found, the full state $\mathbf{x}$ can be recovered using Equation 3.3 and the reconstruction is concluded. This means that the problem is reduced to an optimization problem: it is required to find the best match $\hat{\mathbf{s}}$ that minimizes the difference between the actual measurement $\mathbf{y}$ and its prediction $\hat{\mathbf{y}}$, which is depicted mathematically in Equation 3.5, where the subscript 2 refers to the least squares method (other methods could be chosen as well).

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{y} - C\Psi\mathbf{s}\|_2 \tag{3.5}$$

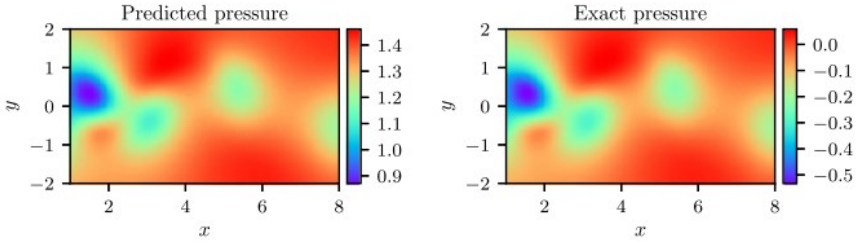
Advantages of this method are that the method does not seem to be prone to noise, and can handle complex flow fields. On the other hand, only the measured state is retrieved, similarly to POD. Furthermore, an extensive library of very similar (complex) flow types must be present for this method to be able to give reasonable results. Applications in current research of these methods, among others, are that of flow over a stalled airfoil, flow inside gas turbine combustors and flows with periodic vortex shedding [1, 2, 20]. The main conclusions are that generally the performance of POD and sparse reconstructions depend strongly on the placement of the measurement equipment over the domain. Furthermore, as mentioned, at least some measurements is needed on the flow state to be reconstructed.

Furthermore, efforts have been done to recover flow states from sparse measurements using a shallow NN [21] or a convolutional NN (CNN) [22]. The main benefit of this is that the relation between the measurements and the flow state can also be nonlinear when using NNs, whereas that is not the case with the previous method. Although this gives promising results for reconstructions in situations with sparse data, still some measurements of the to be reconstructed state are required, as in all the other mentioned methods as well. An attempt has been done to map Hydroxyl Planar Laser Induced Fluorescence (OH-PLIF) images to PIV using a CNN in a reacting flow [23]. However, one of the main conclusions drawn is that the method is incapable of retrieving the correct velocity fields when applying the mapping to OH-PLIF images at different operating conditions than operating conditions the network has been trained on. This means it would still be required to do measurements in similar conditions to be able to apply it.

It is clear that the methods discussed above all require some form of measurement of the state to be retrieved, for example at another time instant or spatial location. However, in some cases it is impossible or very expensive to measure a particular flow state even once. Furthermore, requiring measurements of all flow states increases the complexity and cost of the experimental apparatus. Moreover, some measurement techniques can not be used in parallel, meaning that even in case such a complex and expensive experimental apparatus is available for an experiment, the measurements would be asynchronous. For these reasons, there is still a need to be able to infer unmeasured flow states from measured ones. There is a recent reconstruction method that attempts this, called HFM, which is a PINN focused on reconstructing ‘hidden’ flow states in fluid flows. PINNs work by minimizing a loss function based both on the residuals of the governing equations and the error of the prediction compared to the provided measurements. A more theoretical background about PINNs and HFM was provided in chapter 2. As HFM is capable of retrieving flow states for which no measurement at all have been done, where all other methods are not, this method will be considered for this research.

3.2. Hidden Fluid Mechanics

PINNs have been applied to different physics problems such as problems concerning the NS equations. An example is a method known as 'data-driven discovery', where part of the NS equations contains undefined constants and the idea is to find those by using the velocity field of an existing solution, thereby also finding the pressure distribution [3]. The flow on which this is shown is an incompressible flow past a cylinder. The results are shown in Figure 3.1, where it can be seen that the constants are predicted up to an accuracy of a few percent, even in case of added noise to the data. The predicted pressure field is matching the exact one up to a constant, as there are only derivatives of the pressure in the NS equations, allowing to recover the pressure up to a constant only.



Correct PDE	$u_t + (uu_x + vu_y) = -p_x + 0.01(u_{xx} + u_{yy})$ $v_t + (uv_x + vv_y) = -p_y + 0.01(v_{xx} + v_{yy})$
Identified PDE (clean data)	$u_t + 0.999(uu_x + vu_y) = -p_x + 0.01047(u_{xx} + u_{yy})$ $v_t + 0.999(uv_x + vv_y) = -p_y + 0.01047(v_{xx} + v_{yy})$
Identified PDE (1% noise)	$u_t + 0.998(uu_x + vu_y) = -p_x + 0.01057(u_{xx} + u_{yy})$ $v_t + 0.998(uv_x + vv_y) = -p_y + 0.01057(v_{xx} + v_{yy})$

Figure 3.1: Pressure field and constants in NS equations found using 'data-driven discovery' compared to the exact ones [3]

Another example is that of HFM, where in this case the solution of the concentration of a passive scalar c is used in combination with the governing equations to find the full flow solution [4, 5]. This is done in an incompressible flow for a Newtonian fluid, resulting in the outcomes seen in Figure 3.2. It is promising to see that the predictions match reasonably well with the exact solutions. However, for both these examples the case considered is an incompressible flow with limited Reynolds number (100), purposely avoiding regions where the NS equations become turbulent and chaotic. Another limitation is the use of very clean concentration data, which has been synthetically generated, so not measured with possible measurement errors.

The previously mentioned research motivated others to apply HFM to other problems. An example of this is a work in which the three-dimensional velocity and pressure fields are inferred from the temperature field found from Tomographic Background-Oriented Schlieren (Tomo-BOS) in case of the flow over an espresso cup. The governing equations used are the incompressible NS equations with the Boussinesq approximation [24]. In Figure 3.3, the results from the Tomo-BOS PINN are compared to the experimental method of Particle Image Velocimetry (PIV). The comparison between the two is meant to qualitatively validate the PINN's results. The results can not be compared one to one as the time instances between the Tomo-BOS PINN and PIV are different and the experiments have been performed independently. This has to do with the fact that the Tomo-BOS and PIV experiments could not be performed in parallel. Overall, similar flow patterns are observed, and the maximum and minimum velocity are the same in both methods. The main differences are seen just above the flame, where the PIV shows more flow variation and in the first time instant even flow recirculation, whereas the Tomo-BOS PINN does not capture this. One benefit of the Tomo-BOS PINN is that it can give data in a time continuous manner whereas PIV is limited by its acquisition frequency and by the necessity of setting up the experimental apparatus. Another benefit is that measurement data and the predictions from the PINN are parallel in time, whereas this is not the case when PIV is used.

Similarly, research has been done on generating a dense velocity reconstruction from PIV or Particle Tracking Velocimetry (PTV), thereby also recovering the pressure distribution [25]. The idea of this is that PIV and PTV normally give a qualitative insight, whereas reconstructing the full velocity field would also enable a quantitative insight. This has been tested in a case of Taylor-Green vortices, which are decaying vortices in an incompressible flow for which the exact solution is known. The research

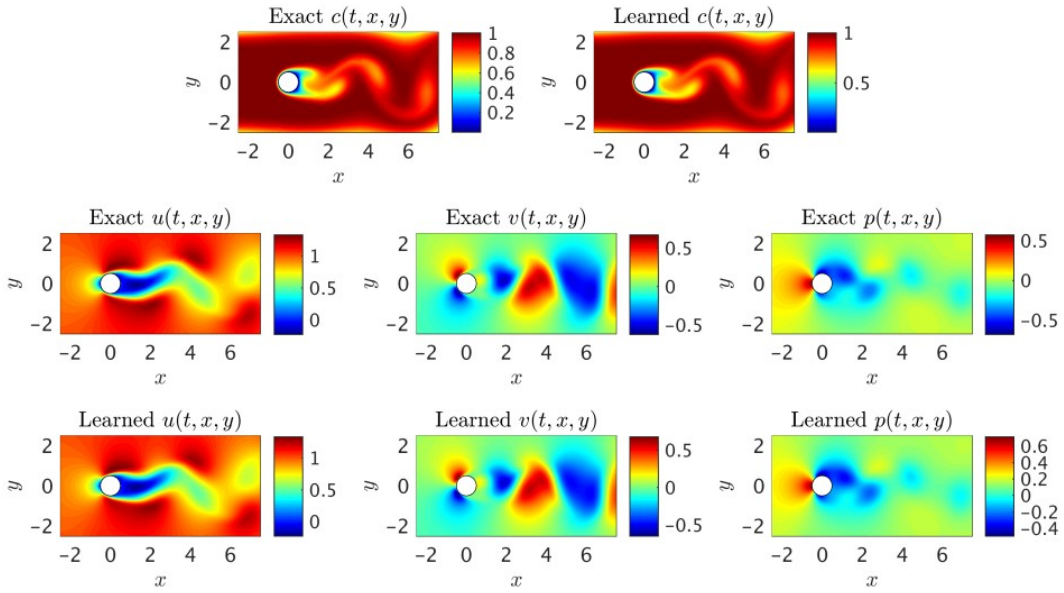


Figure 3.2: Concentration field, velocity components and pressure found using HFM [4]

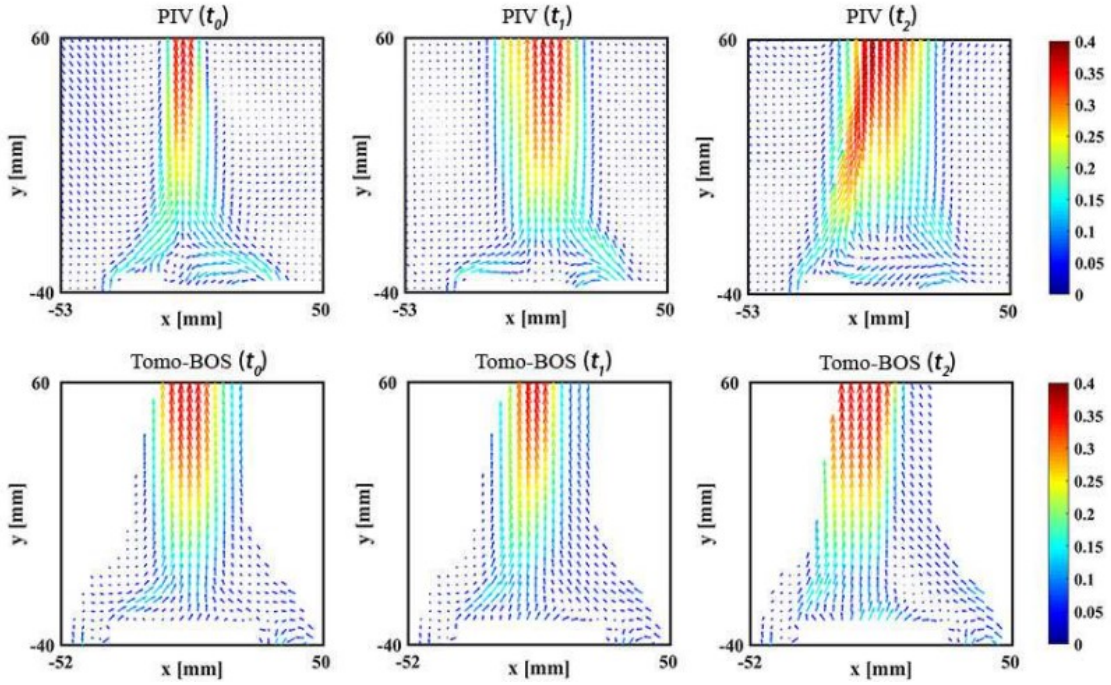
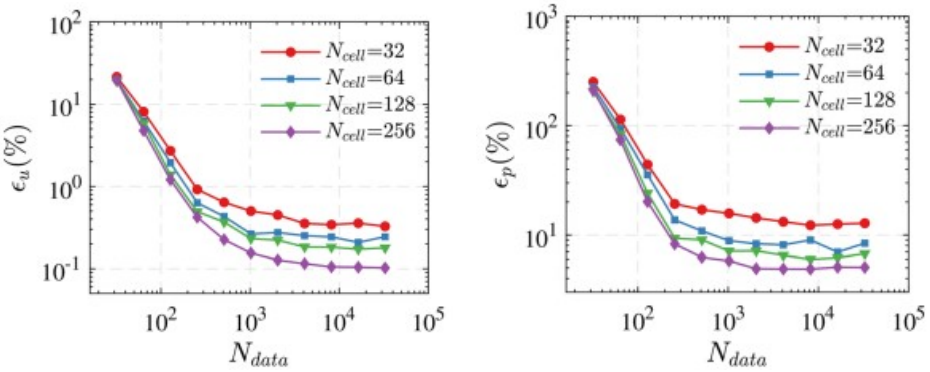


Figure 3.3: Velocity fields from PIV (top) at times $t = 1.0s$, $7.0s$, and $16.0s$ compared to the velocity fields from the Tomo-BOS method (bottom) at $t = 2.0s$, $4.0s$ and $6.0s$ in case of the flow over an espresso cup [24]

focused on the effect of changing the number of provided data points, the number of neurons in each hidden layer and the weighing coefficient (α in the paper) on the errors in pressure and velocity. The weighing coefficient is the factor in front of the physics term in the loss function (Equation 2.3).

In Figure 3.4, the error as a function of the number of points where the data has been provided and the number of neurons in each hidden layer can be seen in case no extra noise is added. The number of layers was set to seven and all other parameters were kept constant for both cases. It can be seen in Figure 3.4a that the error in velocity is approximately one order of magnitude smaller than the error in pressure seen in Figure 3.4b. This can be reasoned by the fact that the pressure has to be retrieved fully, whereas the velocity is fit to measured data. Also, both errors go down if more data

points are added up to a certain point, where adding extra data points has hardly any effect. Adding more neurons in each hidden layer seems to have the effect of shifting the error graph down. This can be reasoned by the fact that adding more neurons in the hidden layers enables the network to represent the function better (universal approximation theorem). Similar results occur in case noise is added with one major difference: there is hardly an effect on the velocity error when adding more neurons in the hidden layers, whereas the effect is still there for the pressure error. Another observation is that the errors are generally higher when noise is added. The latter is understandable, as adding more noise to the measurement makes the network try to match faulty data, meaning that the result will also be of worse quality.

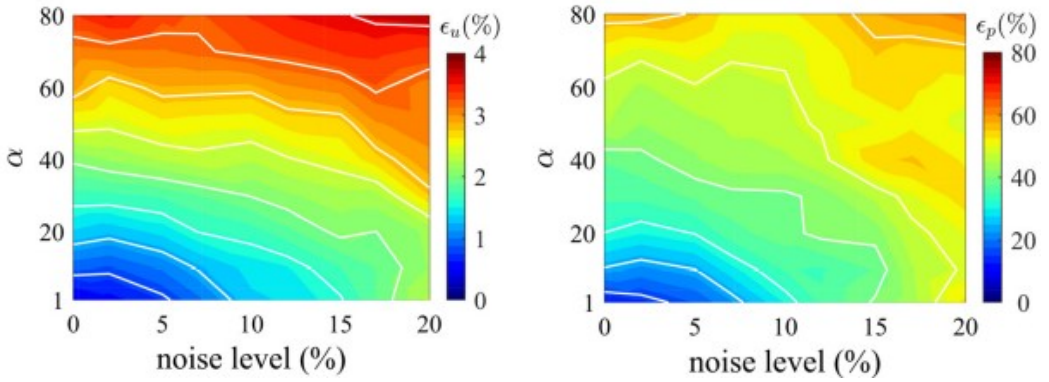


(a) L2 error of the velocity as a function of amount of data points and neurons per hidden layer

(b) L2 error of the pressure as a function of amount of data points and neurons per hidden layer

Figure 3.4: L2 errors as a function of number of data points and number of neurons per hidden layer in case of no added noise, seven hidden layers and weighing coefficient set to one [25]

The effect of the weights in the loss function are then considered and can be seen together with the effect of noise in Figure 3.5. In Figure 3.5a, it is seen that the method's predicted velocity field is closest to the exact one in case the weighing coefficient α is low (one) and with low noise. The same behaviour is seen for the pressure, but the error is an order of magnitude higher. Important conclusions from the research are that the PINN cannot handle very well the spatially correlated noise introduced in cross-correlation in PIV, meaning that PTV data with higher quality and resolution is better suited for PINN applications. Furthermore, in most cases the optimal value of the weighing coefficient has been found to be one, except for high noise levels in the data, which meant the optimal value increased to around two. This is due to the fact that the weight decreases the effect of the noise, as the loss focuses more on the governing equations and less on the measurement with noise.



(a) L2 error of the velocity as a function of weighing coefficient and added noise

(b) L2 error of the pressure as a function of weighing coefficient and added noise

Figure 3.5: L2 errors as a function of weighing coefficient and added noise in case of seven hidden layers [25]

Apart from the methods discussed, there are many new methods that use PINN and combine or add another concept to it. Firstly, the concept of domain decomposition has been tested [26, 27]. The main idea of this is to divide the domain in smaller domains for which the method can be applied, so that it can be run on different cores/servers, requiring the passing on of information at the boundary between the smaller domains. Another concept is that of time decomposition, which is very similar but in the time domain. The reason for using this is different: it is mainly used to increase the accuracy in time for problems with given initial conditions [28]. As the time is just an input to the network like any other (for example spatial coordinates), the network in principle allows for violation of causality as the future values could influence past values. This generally leads to low accuracy for solutions at long time from the initial condition. Using time decomposition might prevent this from happening, as then the causality is not violated anymore (only in the smaller time domains). It is also possible to do both time and domain decomposition [29]. Furthermore, there has been research on what activation functions to use in case of PINNs [30].

From previous examples, it is clear that PINN/HFM has some potential for flow quantity reconstruction, but there are also still several uncertainties: how it would behave in high Reynolds number flow, reacting flow, compressible flow and more. It is thus important for the knowledge about this reconstruction method to investigate its possibilities and capabilities, but also its limitations. It is decided to investigate a particular case of a pool fire, which is a small diffusion flame showing periodic behaviour. Pool fires are interesting because HFM has not been tested on reacting flow yet, meaning it will show the potentials and limitations of using HFM in reacting flows. For more theory about pool fires, one can refer to chapter 2. The use of HFM in case of pool fires could potentially impact fire research, as in case of favourable results it would allow to retrieve the full flow of a real pool fire by just measuring the temperature field or smoke pattern development. Inspiration for potential test cases for the use of HFM in case of pool fires are introduced in the next section.

4

Research Objectives

Although HFM seems to be a promising method, to date it has solely been applied to a few canonical flow cases. There are still many scientific gaps to be explored, one of which being the use of HFM in reacting flows. For this reason, it is regarded relevant to research the potential use of HFM in case of pool fires. Considering the fact that this is a relatively new method applied in a case differing from the common application, several questions are still open. Firstly, one does not know how many flow states are needed to retrieve the full flow solution up to a certain accuracy in case of pool fires. Secondly, it is also not known what the consequences for the accuracy of this method are in case noise is added, or if coarse data is used. For these reasons, the following research question has been formulated:

To what extent can the concept of Hidden Fluid Mechanics be exploited in the problem of a steady and unsteady puffing pool fire, to enable the recovery of the full flow states from few known flow states?

Considering the persisting vagueness regarding the extent of exploitation, sub-questions need to be stated to clarify the previous formulation:

- Can HFM correctly retrieve the flow solution from few provided states in pool fire problems? Do the retrieved states' values match the purposely removed states' values up to a sufficient accuracy and does the total flow solution cause the residuals in the governing equations to be small?
- What is the reconstruction accuracy at which the unknown flow states can be recovered?
- How does the reconstruction accuracy behave when noisy measurement states are provided?
- How does the reconstruction accuracy behave when decreasing the spatial resolution of the provided measurement states?
- How does the reconstruction accuracy behave when fewer flow states are provided?
- How does the reconstruction accuracy behave when applying the method to the unsteady puffing pool fire compared to the steady pool fire?

Considering the first and second sub-questions, the answer will have an impact on the possible applications of HFM. For example, if it is possible to indeed reconstruct the full flow solution from a partial solution and the reconstruction is of high accuracy, it might replace measurement devices as they are very costly. On the other hand, if it is not always possible and/or if the accuracy is lower, the method can be seen as a qualitative method to be able to capture some flow features, but not capturing correctly all details. In the most extreme case, the method might be not suitable for use in reacting flows at all.

Moving to the third and fourth, which have great impact on if applying the method to actual problems is realistic or not. To elaborate, if only extremely low noise levels are accepted, the method can never be applied on real measurements as they always contain a significant level of noise. Furthermore, if an extremely high spatial resolution is required, the method can not be used with real measurements, as there is a limit on the spatial resolution which is achievable using current experimental methods.

The fifth sub-question has an impact on the amount of flow states still required to be measured in an experiment: if the full flow can be recovered from for example only the temperature field, measuring the temperature field in an experiment would be sufficient. On the other hand, if all except the velocity fields are to be provided, these fields all need to be measured, which adds costs and might not even be practically possible. Note that uniqueness of the reconstruction problem is not proved even in case all states except the velocity are provided.

The research question related to reducing the provided flow states might give an indication on when the prediction with close to zero full loss is close to the actual one (this implies the solution might be unique, or at least that the minimum in the optimization domain could be found) or far away from the actual one, meaning the uniqueness is probably not present, as apparently there is another solution to the reconstruction problem with very low loss which is not the actual solution. This can be seen as a different minimum value in the optimization domain than the one corresponding to the actual solution.

Lastly, the last sub-question impacts the possible applications of HFM on unsteady flow: for example, if the results in case of steady flow are of much higher accuracy than those of unsteady flow, one might consider only applying HFM in case of steady pool fire flow.

5

Methodology

As the research questions are clear, the method to answer those can be discussed in more detail. Firstly, in section 5.1, the flow cases the framework is applied to, including their geometry and relevant occurring physical phenomena are set out. After this, the reconstruction problem and its application in these flow cases are described in section 5.2. Finally, the architectural and computational settings related to the PINN are outlined in section 5.3.

5.1. Flow cases

In this section, the flow cases to which the method is applied are discussed. Firstly, the DNS and its settings to obtain the data used in this work are elaborated upon in subsection 5.1.1. To evaluate the robustness of the PINN in different cases, both a steady and unsteady flame are considered. Firstly, a pool fire with a pool diameter well below the critical diameter is considered in subsection 5.1.2, leading to a steady flame. Secondly, an unsteady flame exhibiting puffing behaviour is treated in subsection 5.1.3, generated by selecting a pool diameter above the critical diameter.

5.1.1. DNS

The data used in this thesis comes from a DNS performed using fireFoam, which is a solver in OpenFoam being able to incorporate combustion reactions. The governing equations are to be discussed subsequently. Firstly, no mass is added or removed, meaning that the mass is conserved, as stated in Equation 5.1.

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \tag{5.1}$$

Secondly, there should be a balance between the change in momentum and the forces caused by the pressure gradient, stresses, and gravity. This leads to the momentum equation seen in Equation 5.2, in which τ is the molecular stress tensor, g_i is the gravity field strength in direction i and the letters i and j refer to the indices of the vectors/tensors.

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i \tag{5.2}$$

The third equation states that the enthalpy can only changes due to diffusion, as written in Equation 5.3. This means that steep gradients of temperature are flattened out over time to distribute the energy more evenly over the domain. In this, α is the thermal diffusivity.

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\rho \alpha \frac{\partial h}{\partial x_i}\right] \tag{5.3}$$

Finally, as combustion occurs in the flow, the separate species mass fractions are governed by a conservation law as well. Similarly to the enthalpy, particles diffuse. Furthermore, certain species are increasing in mass fraction whereas others are decreasing, depending on the chemical reaction and its

rate. Thus, the species mass fraction equation for species k can be written as Equation 5.4, in which $\dot{\omega}_k$ refers to the reaction rate.

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho u_i Y_k}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\rho D \frac{\partial Y_k}{\partial x_i}\right] + \rho \dot{\omega}_k \tag{5.4}$$

These governing laws are applied in an two-dimensional axisymmetric pool fire using n-heptane fuel for which the irreversible chemical reaction can be seen in Equation 5.5, for which the reaction rate is found from the Arrhenius law. The ambient pressure and temperature in the far field are set to $p_0 = 100kPa$ and $T_0 = 300K$, respectively. Furthermore, the far field oxygen fraction is set to $Y_{O_2} = 0.233$, whereas the far field nitrogen fraction is set to $Y_{N_2} = 0.767$. These values are chosen as they are similar to real-life values at sea level.

$$C_7H_{16} + 11O_2 \rightarrow 7CO_2 + 8H_2O \tag{5.5}$$

As the governing equations used and the ambient BCs are discussed, a schematic of the domain can be introduced, as seen in Figure 5.1. It is seen that the ambient boundary at the top of the cylinder is at a height of 50 times the pool radius, whereas the ambient boundary on the side is at a radius 5 times the pool radius. These values are chosen to ensure that the boundaries have no effect on the flame and its dynamics. The floor away from the pool is set to a constant temperature (isotherm).

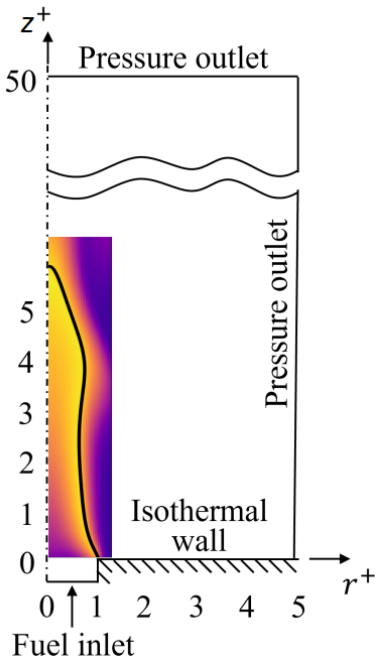


Figure 5.1: Illustration of the DNS domain. z^+ and r^+ refer to the axisymmetric coordinates normalized by the set pool radius a with unit in mm

The only BC left to discuss is that of the pool itself, which is assumed to be at the boiling temperature of n-heptane ($T_B = 371.5K$). The mass flux of the fuel at this surface is determined from the fuel evaporation rate. Other assumptions made are the ideal gas law to determine the density and the Sutherland law to find the viscosity.

As the domain has been discussed, some numerical settings can be introduced. Firstly, the mesh used is a structured grid of 600 by 130 nodes, which is finer near the flame and coarser further away. The finest region has a resolution with an order of magnitude of approximately $0.1mm$. Furthermore, the numerical schemes that were used were first and second order in time and space respectively. The time step was set to a constant of 10^{-5} in order for the Courant–Friedrichs–Lewy (CFL) number to stay below 0.1.

This numerical setting is used for two cases: flow with a small pool diameter, resulting in a steady flame, and flow with a larger one which results in a puffing flame, as mentioned previously. As the

whole domain is parameterized (normalized) with respect to the pool diameter, it is a matter of scaling the whole domain. For the steady case, the simulation is run until a converged steady solution is obtained. For the unsteady case, $400ms$ are simulated, and the evolution over time is saved in 400 time snapshots separated by $1ms$.

5.1.2. Steady Flame

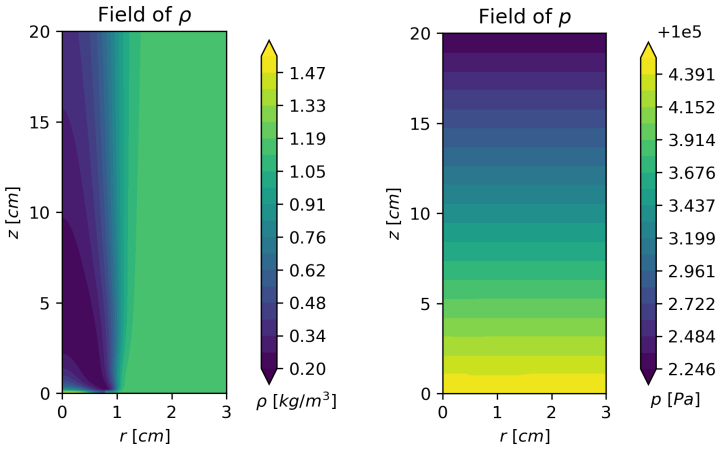
When setting the pool diameter to $15.9\ mm$, the resulting flame from DNS shows a steady behaviour, as expected from theory as the critical pool diameter is not exceeded. The corresponding flow fields can be seen in Figure 5.2 (note the scales for r and z are not the same and that the domain is cropped at radius and height 1.89 and 12.58 times the pool diameter respectively ($r = 3cm$ and $z = 20cm$) for plotting and readability purposes, which is always done from now on), where half of the cylindrical domain is plotted (the flow case is assumed to be axisymmetric).

The density field seen in Figure 5.2a shows constant density in the far field as ambient conditions are present there. At the fuel inlet, there is very high density due to the dense fuel. After the chemical reactions, the density in and above the flame drops, causing the air to rise. The pressure field seen in Figure 5.2b shows almost perfectly linearly decreasing pressure when increasing the axial location (height), meaning the pressure gradient with respect to height is approximately constant. This occurs naturally in vertical domains with nonzero gravity as the air closer to the bottom feels more air pushing from the top. The vertical pressure gradient in a situation with fixed air ignoring any other effects can be described using the hydrostatic equation, shown in Equation 5.6.

$$\frac{dp}{dh} = -\rho g \tag{5.6}$$

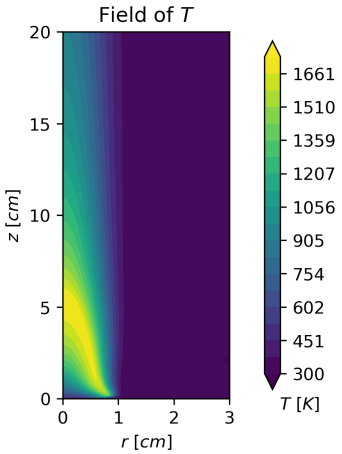
Interestingly, the pressure field from the steady flame seems to adhere almost perfectly to this, having a nearly constant vertical pressure gradient everywhere in the flow, and showing no other effects even though the flame is present. This linear pressure gradient is more clearly depicted when plotting the pressure over the height z at a certain radius r, as seen in Figure 5.3. The pressure distribution over height z is almost exactly equal to a linear line, with just very tiny disturbances. This observation is at the base of an assumption made later on in subsection 6.1.4. The pressure at the top of the domain (not depicted as the domain is cropped for illustration purposes) equals the ambient pressure of $10^5\ Pa$. In Figure 5.2c, the temperature field can be seen. It shows ambient conditions far away from the flame, as the flame is too far away to have an effect there. At the fuel inlet, the temperature is still ambient, as the pure fuel needs air (oxygen) to be able to burn. Just above the fuel inlet, the temperature increases strongly, caused by the heat generated by the chemical reactions. The lower temperatures further above are caused by the air slowly cooling down as it rises.

The axial velocity field can be seen in Figure 5.2d. As one would expect, the air above the flame rises, indicated by a positive axial velocity. Furthermore, the closer to the center of the flame, the faster the air rises. It is also seen that the radius of affected air increases for increasing height z. Interestingly, the maximum axial velocity does not occur in the region of maximum temperature, but well above that. This can be reasoned by the inertia of the air: it takes some time for the air to accelerate, in the process already cooling down, meaning that the maximum axial velocity is located at increased height z compared to the location of maximum temperature. In Figure 5.2e the radial velocity field is shown. It can be seen that the maximum radial velocity is an order of magnitude smaller than the axial velocity. Furthermore, in the majority of the flow, the radial velocity is nearly zero. The radial velocity has one major peak near the fuel inlet outermost radius, having a negative radial velocity indicating the flow is towards the center. The chemical reactions occurring slightly above the fuel outlet cause a decrease in density, leading to new air being sucked in, thus causing the negative radial velocity peak.

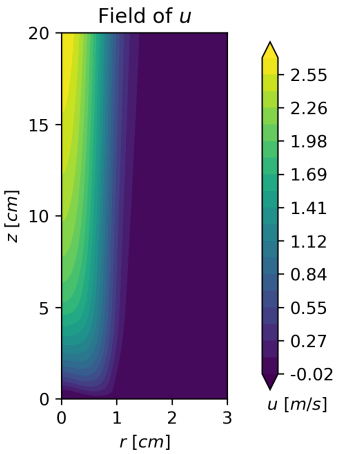


(a) Field of ρ from DNS

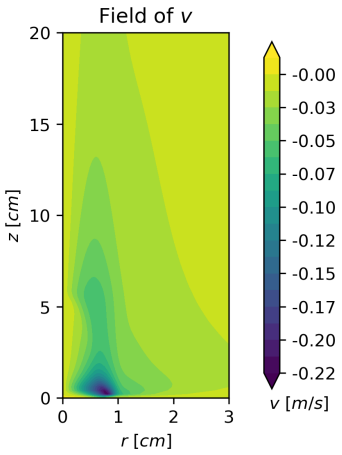
(b) Field of p from DNS



(c) Field of T from DNS



(d) Field of u from DNS



(e) Field of v from DNS

Figure 5.2: Flow fields of steady flame from DNS

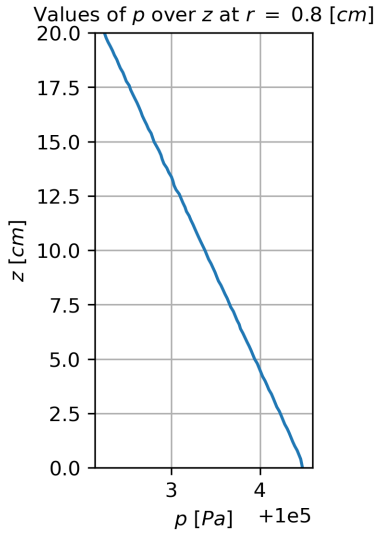


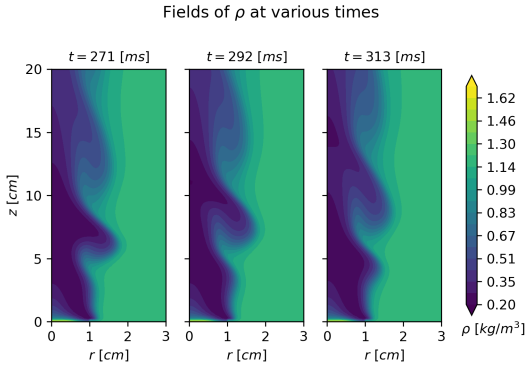
Figure 5.3: Pressure distribution over height z at radius 0.50 times the pool diameter($r = 0.8cm$) of steady flame from DNS

5.1.3. Puffing Flame

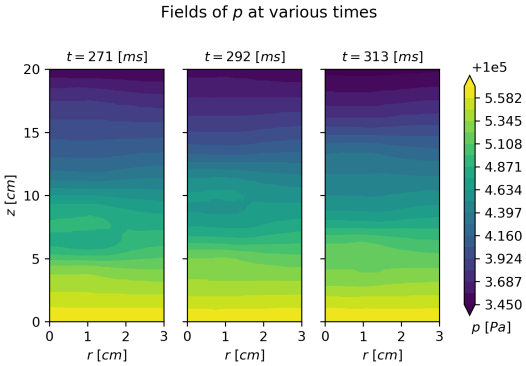
When increasing the pool diameter to $19.1\ mm$, the resulting flame from DNS shows unsteady puffing behaviour, expected from theory as the critical pool diameter is exceeded. As this leads to unsteady flow, the flow fields at multiple time snaps are shown. The flow fields at times $t = 271ms$, $t = 292ms$ and $t = 313ms$ can be seen in Figure 5.4 (note the scales for r and z are not the same). The decision for these time snaps is based on the period of the unsteady motion, which is elaborated upon subsequently.

The density fields seen in Figure 5.4a show similarities with the field of the steady case, where the main difference is that here pockets of high density are rolling up, rising up and dissipating. This means that there are vortices being generated, as is known to be the case in the puffing flame, caused by the strong velocity gradients, explained in chapter 2. The pressure fields over time can be seen in Figure 5.4b, where two differences with the steady case can be found. Firstly, the values have increased in general, but this is solely caused by the expanded domain. To elaborate, the increase in pool diameter increased the domain size (as it is parameterized), meaning the ambient pressure of 10^5 Pa is at increased height z causing the pressure to be slightly higher at the bottom of the domain. The other main difference is that the pressure gradient is not fully vertical and constant anymore, caused by the mixing effect of the vortices. This can be more clearly seen when looking at the pressure plot over height at a certain radius, seen in Figure 5.5, where the graph is curved with fluctuations approximately around the linear curve that has been seen for the steady case. In Figure 5.4c, the temperature fields can be seen for the three different time snaps. The main difference with the steady temperature field is that there are regions of high temperatures rolling up exactly at the locations where this happens for the density.

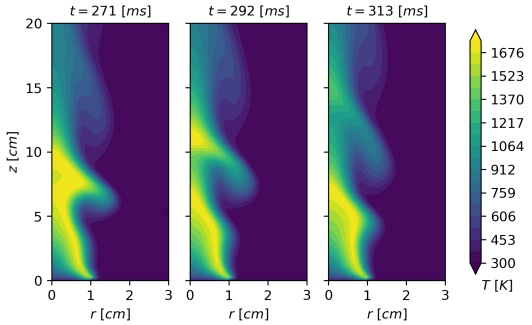
For the axial velocity fields seen in Figure 5.4d, the main differences are that the location where the maximum axial velocity occurs has decreased significantly in height and that there are regions of lower axial velocity moving up in the domain and dissipating over time. The first can be explained by the effect of the vortices, accelerating the flow more than usual before moving it radially. The second effect can be reasoned by the flow being deflected in radial direction (rolling up as a vortex), meaning it loses its axial velocity. Lastly, the unsteady effects of the radial velocity fields can be seen in Figure 5.4e. The highly negative radial velocity near the pool showing up in the steady flow is still there, and the radial velocity is still zero in most of the field. However, pockets of negative and positive radial velocity appear above the flame and seem to make an alternating pattern. This is caused by the vortices: a vortex induces both flow away from the plume (above its center) and towards the plume (below its center). As the vortex rises and dissipates, the effect becomes weaker. New vortices are generated and previous vortices rise, causing an alternating pattern to appear.



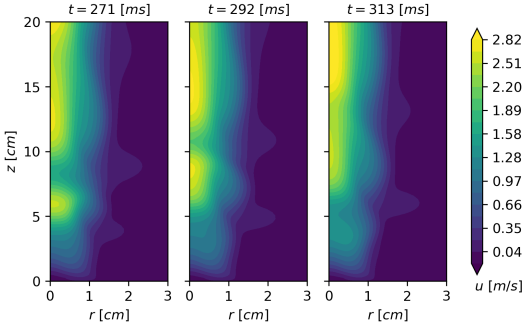
(a) Fields of ρ from DNS at various times



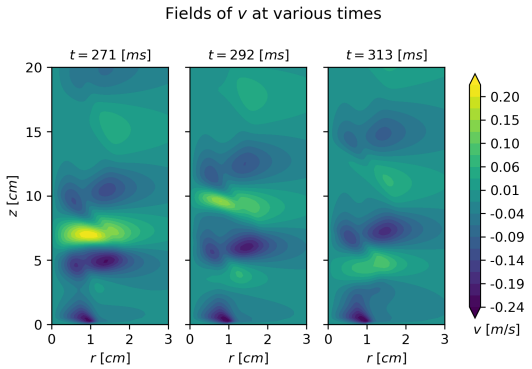
(b) Field of p from DNS at various times



(c) Field of T from DNS at various times



(d) Field of u from DNS at various times



(e) Field of v from DNS at various times

Figure 5.4: Flow fields of unsteady flame from DNS at various times

The alternating pattern generated by the vortices implies periodic vortex shedding. Furthermore, it is known from theory that the puffing behaviour of the flame shows periodic behaviour. Taking a point in the domain (radius and height 0.63 and 2.98 times the pool diameter, meaning $r = 1.2cm$, $z = 5.7cm$) and plotting the flow states over the full time domain, one obtains the graphs clearly showing periodicity plotted in Figure 5.6, where only the density and axial velocity are plotted for conciseness. Using the Fourier transform, one can find that the period is approximately $84ms$ for both, meaning that Figure 5.4 shows the fields at the start of a period, at a quarter of a period and at a halfway point. These three time snaps are also used to show the performance of the PINN when predicting the unsteady fields, as other time snaps will just be repetitions or interpolations of these, which will become more clear in chapter 6. It should be noted that even though the (main) period of the fluctuations of the flow states is equal, their phase and amplitude is very different, and that those also heavily depend on the location coordinates selected. Furthermore, the axial velocity (and the radial velocity as well) seems to show a local maximum just before the global maximum in every period, interestingly. This occurs due to the fluctuations not being a perfect sinusoidal function, but rather a more complex build-up function due to

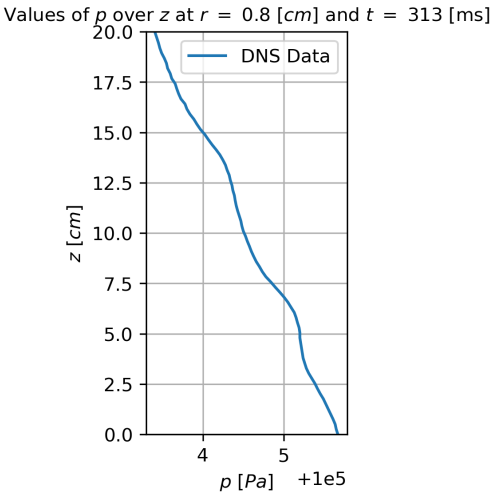
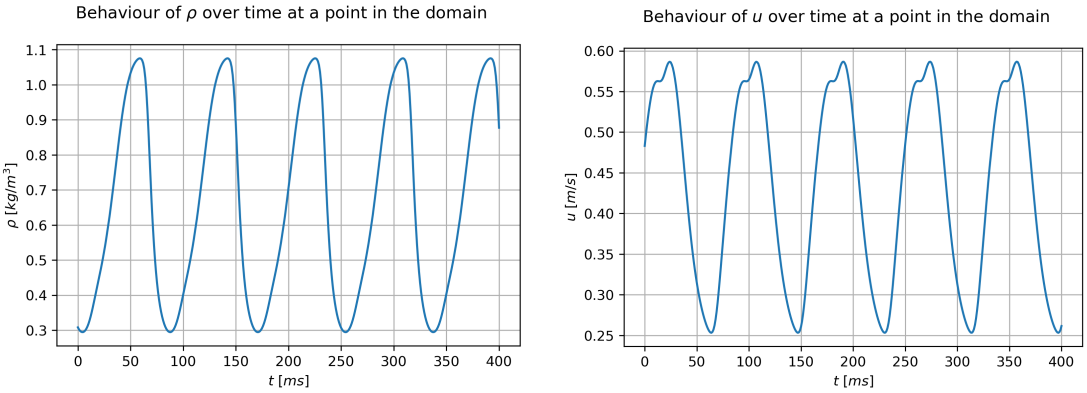


Figure 5.5: Pressure distribution over height z at radius 0.42 times the pool diameter ($r = 0.8cm$) at $t = 313\ ms$ of unsteady flame from DNS

the effects caused by the vortices and their interactions.



(a) Fields of ρ from DNS at various times

(b) Field of u from DNS at various times

Figure 5.6: Flow fields of unsteady flame from DNS at various times

5.2. Reconstruction Problems

As the flow cases have now been introduced in more detail, the specific reconstruction problems to be solved can now be introduced, as depicted in Figure 5.7. The general problem boils down to retrieving the axial and radial velocity fields using the provided density, pressure and temperature fields. This means the velocity fields are purposely removed from the DNS results, after which the velocity fields are predicted by the PINN and finally compared to the previously removed velocity fields coming from the DNS. All computations and predictions are on the points coming from the DNS mesh, being fine near the flame and coarser in the far field. The following cases are considered, coming from the research questions introduced in chapter 4:

- Standard case
- Coarsening data
- Adding artificial noise
- Removing more flow states, reconstruction problem needs to be adjusted

Firstly, the standard case has been introduced above, where the data is not changed and the density, pressure and temperature are used to find the purposely removed velocity fields.

Secondly, in case of coarsening the only difference is that the data is provided at fewer points throughout the domain. For simplicity, the coarsening is handled by removing every other point in both z and r directions everywhere in the domain to get a double coordinate spacing everywhere, or repeating this multiple times to get the spatial resolution divided by a power of two everywhere. This means that the spatial distribution of points remains similar despite the coarsening, hence there are still more points close to the flame than in the far field. One could also consider providing equally spaced data, which would be expected if data comes from an actual experiment as measurement equipment generally has a uniform spatial resolution. This, however, is not an arbitrary task as it removes the implicit weights related to the importance of the flame and the relative unimportant far field (as in the data from the DNS, more points are in the near field, which is not the case in a uniform grid). Providing data in a uniform grid removes the weights implicitly thus requires assigning higher weights to the data in the flame than to the data in the far field, which if not done correctly lead the PINN to a (wrong) constant solution. This is discussed in more detail in chapter 6, where the data is provided in a structured grid for the sake of reproducing similar results and thus showing that it is possible.

Thirdly, when noise is present, the only difference is that the provided density, pressure and temperature fields have an artificially added noise on top of their actual values. It is decided to artificially generate white noise in space and time, with no bias and with various Signal to Noise Ratios (SNRs), as this is the most straightforward and simple method to generate noisy data. To be able to add noise with a certain SNR to the flow states, the signal power needs to be found first using Equation 5.7. In this $s_{i_{state}}$ is the signal for a particular flow state at a certain spatial and temporal coordinate, μ_{state} is the mean value over the full domain of this flow state and P_{state} is the power of the signal of the particular flow state. Furthemore, N refers to the total amount of space-time coordinates in the data. The mean needs to be subtracted before squaring the signal to after which it is summed and the mean is taken to obtain the power as that gives information about how much the signal actually changes from its mean on average.

$$P_{state} = \frac{\sum_{i=1}^{N} (s_{i_{state}} - \mu_{state})^2}{N} \tag{5.7}$$

Using Equation 5.8, the noise power P_{noise} can be calculated from the SNR value and the signal power P_{state} of the particular flow state described above. Note that if the SNR value is given in dB this would first have to be converted. Finally, then noise with the desired noise power P_{noise} is generated using a Gaussian distribution with zero mean. The latter implies that the noise has no bias that would decrease or increase the mean value over the domain.

$$P_{noise} = \frac{SNR}{P_{state}} \tag{5.8}$$

Lastly, removing more flow states than just the velocity fields actually changes the reconstruction problem, as it requires the investigation of the performance of the PINN in case less flow information is known. This means that either only the pressure and temperature are provided, or just the temperature is provided, from which the other fields need to be found.

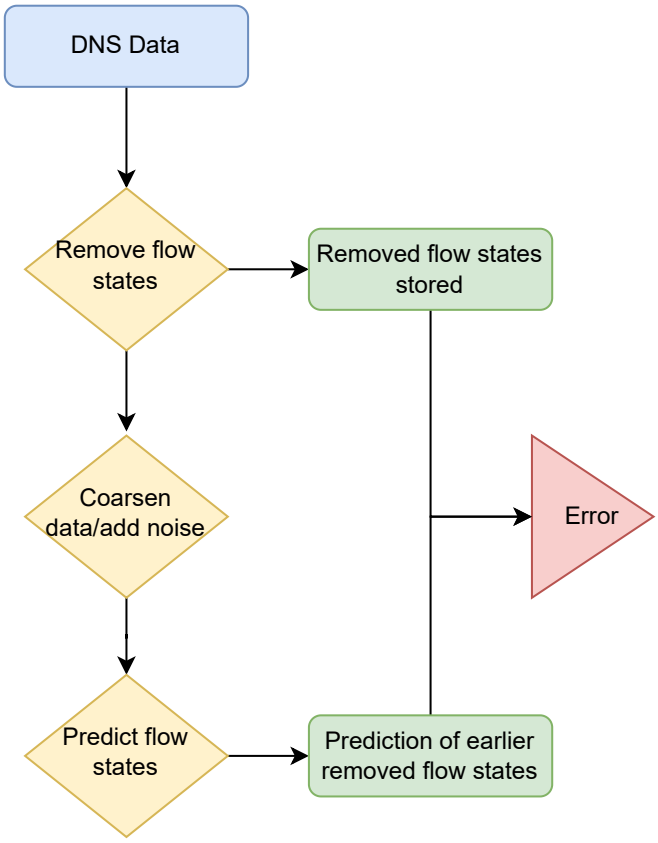


Figure 5.7: Representation of the reconstruction problem

5.3. NN Architecture & Settings

As the cases and the reconstruction problems are now clearly set out, the computational and architectural settings related to the PINN can be described. As the steady and unsteady flow case have major differences both physically and considering the data, the common settings are first set out after which the specifics are discussed separately in subsection 5.3.1 and 5.3.2.

Before discussing the PINN architecture and its settings, one can refer back to Figure 2.3 for the general representation of a PINN. In the two flow cases considered, the inputs provided to the PINN are different. The outputs are five tensors: density, pressure, temperature, axial velocity and radial velocity. From these output tensors, the loss can be computed. The total loss function is build up as written in Equation 2.3, where the α in front of the measurement loss has been assigned a value of 100, meaning the loss caused by the difference between the predicted fields and the provided fields is multiplied by a factor of hundred, whereas the physics loss due to the non-zero residuals in the governing equations is multiplied by one. These, and other computational or architectural settings related to the PINN have been tuned to be able to find optimal conditions for the problem considered, but it should be noted that this has not been done extensively as this research does not focus on finding optimal NN architectures, rather investigating the use of HFM in case of a pool fire.

First, the NN is only trained on the measurement loss, ignoring the physics loss. This decision is made as it is computationally cheaper and more efficient, because the network first adapts quickly (no expensive gradient calculations have to be done in this step) to the provided data, after which it is trained on the full loss and only needs to adapt slightly to reduce the physics loss based on the residuals. The second phase is only entered when the predictions for the provided data have converged, meaning the measurement loss has flattened out. Finally, if the full loss flattened out, the PINN is considered to be fully converged to its most optimal weights and biases, meaning it gives the best prediction on the points, minimizing the full loss.

The loss function to be minimized can be explained in more detail. Firstly, the measurement loss is known to consist of the mean square difference between the predicted fields and the provided fields.

In case the density, pressure and temperature are provided, it thus consists of the mean square differences of the predictions of those fields and their values from the data from the DNS. In case less fields are provided, the measurement loss reduces to only being the mean square difference between the predicted temperature fields and provided temperature fields, or both predicted temperature and pressure fields and provided temperature and pressure fields, depending on the case to be considered. Moving on to the physics loss, the residuals of the governing equations are to be considered, which are separately introduced for the steady case and the unsteady case in subsection 5.3.1 and 5.3.2.

Other important settings related to the NN and the optimization are discussed now. Firstly, an important setting related to the NN to consider is the activation function used. It is decided to use a hyperbolic tangent, which is a common activation function which maps the output of a neuron to values in the range from negative one to one. Furthermore, an Adam optimizer has been used with a learning rate of 10^{-3}, which determines the step taken at every iteration in the direction of the descending gradient [10]. Decreasing the learning rate to a very low number would mean the convergence goes extremely slow, whereas picking a too high number would mean the optimizer does not converge or even diverges. Then, the batch size is set to 10^4, meaning that at every iteration, it determines the loss depending on 10^4 randomly selected points from the domain. Decreasing the batch size would cause the optimizer to go faster through iterations, but with less accurate information of the actual loss (the loss is based on only part of the points in the domain), whereas increasing this number would make iterations costly in time but make the loss that the optimizer minimizes more representative. Then, the collocation points are the points where the residuals are computed, which can be any set of points throughout the domain. For simplicity, it is decided to set the collocation points at the same locations as the measurement points, although in chapter 6, there is a short discussion on using a different set of collocation points.

As a last computational note, it is important to realize that NNs are most efficient if the input and output parameters are in the same order of magnitude and have similar variance. This is not the case automatically, as both the input and output parameters have enormous variations between them, where some are orders of magnitudes higher than others. In fact, normalization is a process which is often done when making use of NNs. For this reason, the inputs are normalized before being fed to the PINN. Furthermore, the outputs are trained to be predicted in normalized manner, meaning the predictions are compared to the normalized data from DNS.

The normalization of parameters is performed by subtracting the mean and dividing by the standard deviation. This means that when using the PINN to predict the final results, one needs to normalize the inputs using the same normalization, and then denormalize the outputs with the same steps as the DNS data has been normalized with. Using this, an obstacle appears: the governing equations use quantities which are not normalized. This issue is tackled by first denormalizing the flow states, before applying the cylindrical NS equations to find the correct residuals. In this process, the velocities are also denormalized, as in this way their predictions coming from the PINN directly are also normalized and thus of the same order of magnitude as the other predicted flow states. The mean and standard deviation of the velocities are unknown upfront, as these flow states are to be retrieved. However, it is possible to make very rough estimates of the mean and standard deviation of those and use that in this step: these estimates do not have to be very accurate, as long as they are of similar order of magnitudes as the actual mean and standard deviation (otherwise, the velocities coming out of the PINN are of different order of magnitude than the other flow states, which is harder to handle for the PINN). The normalization procedure can be schematically seen in Figure 5.8.

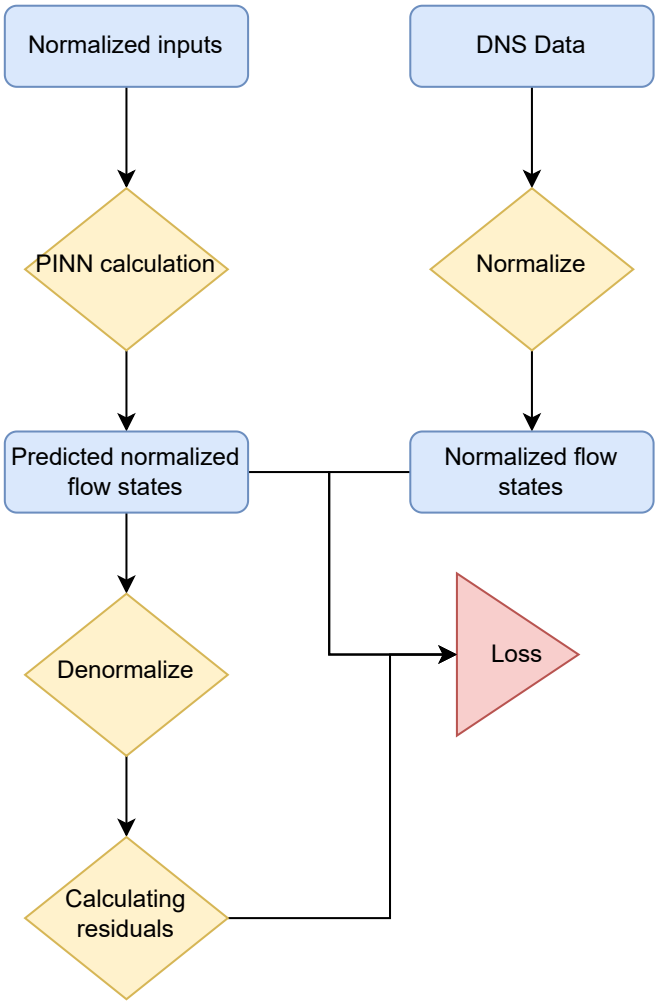


Figure 5.8: Representation of the normalization and denormalization steps taken

5.3.1. Steady Case

In this subsection the specific settings used for the steady flow are elaborated upon. Firstly, the time disappears as an input of the NN, leaving only the two tensors containing the spatial coordinates r and z as inputs. Furthermore, the width and depth of the NN are decided to be 75 and 15 respectively, as these are reaching the smallest values which still give similar results to a large, computationally more expensive network in the steady flow case.

The governing laws also differ for the steady and unsteady flow cases. In the steady flow case, the governing laws for which the residuals are to be minimized by the PINN are the steady mass and momentum conservation in cylindrical coordinates due to the flow being steady and in two-dimensional cylindrical coordinates. The conservation laws are written in Equation 5.9, 5.10 and 5.11, where the ϵ indicates the residual which should be as close to zero as possible. Note that these equations do not take into account chemical reaction due to the burning of fuel, but rather are the conservation laws for the full mixture of chemical elements present, not of separate species. This has been done as these conservation laws should also hold for the mix of species, not only for separate species. Furthermore, measuring chemical compositions of the flow everywhere in the field would be difficult if not impossible, meaning it is more straight-forward and achievable to apply the mass conservation and momentum equations without taking reactions into consideration. The residuals are calculated by finding the gradients using automatic differentiation, introduced in chapter 2. Of these residuals, the mean square is used to add to the loss function, as is done for the measurement error as well. The method is performed using the TensorFlow package in Python[31], as it allows for easy implementations of NNs, automatic differentiation and one can thus easily define the physics-based loss.

$$\epsilon_1 = \frac{1}{r}\frac{\partial\left(\rho r u_r\right)}{\partial r} + \frac{\partial\left(\rho u_z\right)}{\partial z} \tag{5.9}$$

$$\epsilon_2 = -\rho\left(u_r\frac{\partial u_z}{\partial r} + u_z\frac{\partial u_z}{\partial z}\right) + \rho g_z - \frac{\partial p}{\partial z} + \frac{\partial \tau_{11}}{\partial z} + \frac{\partial \tau_{12}}{\partial r} + \frac{\mu}{r}\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right) \tag{5.10}$$

$$\epsilon_3 = -\rho\left(u_r\frac{\partial u_r}{\partial r} + u_z\frac{\partial u_r}{\partial z}\right) + \rho g_r - \frac{\partial p}{\partial r} + \frac{\partial \tau_{22}}{\partial r} + \frac{\partial \tau_{12}}{\partial z} + \frac{2\mu}{r}\left(\frac{\partial u_r}{\partial r} - \frac{u_r}{r}\right) \tag{5.11}$$

In these relations, τ_{11}, τ_{22} and τ_{12} are defined in Equation 5.12.

$$\begin{aligned}
\tau_{11} &= \mu\left(-\frac{2}{3}\nabla\cdot\bar{V} + 2\frac{\partial u_z}{\partial z}\right)\\
\tau_{22} &= \mu\left(-\frac{2}{3}\nabla\cdot\bar{V} + 2\frac{\partial u_r}{\partial z}\right)\\
\tau_{12} &= \mu\left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z}\right)\\
\nabla\cdot\bar{V} &= \frac{1}{r}\frac{\partial}{\partial r}\left(r u_r\right) + \frac{\partial u_z}{\partial z}
\end{aligned} \tag{5.12}$$

Some settings related to the architecture of the NN have already been discussed. A summary of all settings related to NN architecture or the computational optimization domain, their meaning and their set values in the steady case are summarized in Table 5.1. The parameters that have not been introduced yet are the measurement and physics training time. The training times in the steady flow case are $1h$ and $6h$ for the measurement and physics training respectively. This is decided as these allow the loss functions to flatten out over these times. Note that these settings were found by trying out different combinations and as it is costly in time, and the focus is not on excessively improving the PINN's settings, these settings might not be perfect.

Table 5.1: Important architectural parameters and other settings for steady case

Parameter	Value/Setting	Meaning
Width	75	Amount of neurons per layer
Depth	15	Amount of neuron layers
α	100	Multiplication factor for measurement loss
Activation function	Hyperbolic tangent	Activation functions applied to the neurons
Optimization algorithm	Adam optimizer	Loss minimization algorithm used
Learning rate	10^{-3}	Learning rate used in optimization algorithm
Batch size	10^4	Amount of points given per iteration
Collocation points	Measurement points	Points where the residuals are found
Measurement training time	$1h$	Time spent training only on provided data
Physics training time	$6h$	Time spent training on full loss

5.3.2. Unsteady Case

As the specific computational and architectural settings have been discussed for the steady flow case, they are now discussed for the unsteady flow case as well. One of the differences is that an extra input tensor containing the time is now added as an input to the input layer. Furthermore, the problem becomes more computationally expensive: previously only one time snap was required, as the flow was steady, whereas here time snaps every one millisecond are provided, resulting in 400 times as much data. As the problem contains more data and is more complex than the steady flow, the PINN size is adjusted to 400 by 20 to be able to capture the flow physics.

Another difference is that, due to the flow being unsteady, the unsteady cylindrical mass and momentum equations govern the flow, causing time derivatives to appear. The residuals can then be written as in Equation 5.13, 5.14 and 5.15, where the ϵ indicates the residual which should be as close to zero as possible. τ_{11}, τ_{22} and τ_{12} are defined as before in Equation 5.12.

$$\epsilon_1 = \frac{\partial \rho}{\partial t} + \frac{1}{r}\frac{\partial\left(\rho r u_r\right)}{\partial r} + \frac{\partial\left(\rho u_z\right)}{\partial z} \tag{5.13}$$

$$\epsilon_2 = -\rho\left(\frac{\partial u_z}{\partial t} + u_r\frac{\partial u_z}{\partial r} + u_z\frac{\partial u_z}{\partial z}\right) + \rho g_z - \frac{\partial p}{\partial z} + \frac{\partial \tau_{11}}{\partial z} + \frac{\partial \tau_{12}}{\partial r} + \frac{\mu}{r}\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right) \tag{5.14}$$

$$\epsilon_3 = -\rho\left(\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + u_z\frac{\partial u_r}{\partial z}\right) + \rho g_r - \frac{\partial p}{\partial r} + \frac{\partial \tau_{22}}{\partial r} + \frac{\partial \tau_{12}}{\partial z} + \frac{2\mu}{r}\left(\frac{\partial u_r}{\partial r} - \frac{u_r}{r}\right) \tag{5.15}$$

Most architectural settings specific to the unsteady case have now been mentioned of which a summary can be seen in Table 5.1. The only parameters still to mention are the measurement and physics training times, which are set to $8h$ and $16h$ respectively. This is needed as the computational size of the problem increased drastically, meaning fewer iterations can be performed per unit of time, requiring more time to let the loss flatten out.

Table 5.2: Important architectural parameters and other settings for unsteady case

Parameter	Value/Setting	Meaning
Width	400	Amount of neurons per layer
Depth	20	Amount of neuron layers
α	100	Multiplication factor for measurement loss
Activation function	Hyperbolic tangent	Activation functions applied to the neurons
Optimization algorithm	Adam optimizer	Loss minimization algorithm used
Learning rate	10^{-3}	Learning rate used in optimization algorithm
Batch size	10^4	Amount of points given per iteration
Collocation points	Measurement points	Points where the residuals are found
Measurement training time	$8h$	Time spent training only on provided data
Physics training time	$16h$	Time spent training on full loss

6

Results

In this chapter, the results coming from the PINN are shown and discussed. Firstly, the ability to reconstruct flow quantities is investigated in the case of a steady flame in section 6.1. Afterwards, the focus is on the more complex unsteady puffing flame in section 6.2.

6.1. Steady Flame

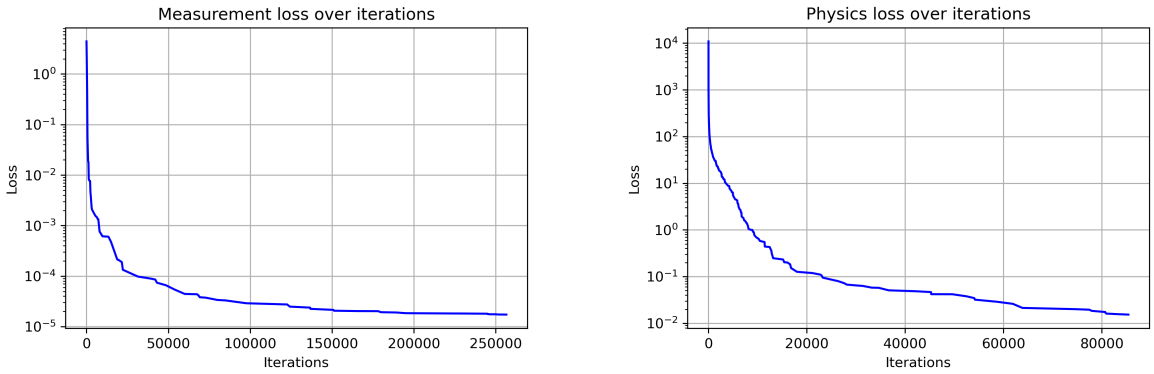
In this section, the performance of a PINN with 15 layers of 75 neurons in retrieving the velocity fields in the case of a steady flame is investigated. Firstly, this is done by using the full data set of the density, pressure and temperature from the DNS in subsection 6.1.1. Afterwards, the effect of coarsening the data in space on the performance is considered in subsection 6.1.2. Then, the robustness to noise is researched in subsection 6.1.3. After this, the effect of reducing the number of provided flow states is inspected in subsection 6.1.4. Finally, a short remark is made about the possibility to provide data in an equally spaced grid in subsection 6.1.5.

6.1.1. Standard

Before looking at predictions, the behaviour of the loss over iterations for both training phases is plotted in Figure 6.1. The plot is generated by storing the loss and iteration number every time the loss is below the previously found minimum, which has been done for readability, as the loss fluctuates heavily due to different batches of points being provided and the network being updated. For the measurement training (Figure 6.1a), it can be seen that the loss flattens out at values orders of magnitudes lower than during physics training (Figure 6.1b). One should take into account that the measurement loss is also multiplied with a factor $\alpha = 100$ in the physics phase, but the major reason for the increase in loss is due to the residuals of the governing equations. At first, the residuals are completely wrong, after which the network is updated until the residuals are closer to zero and the loss flattens out, meaning the network converges to a global or local minimum loss. For other cases, both in steady and unsteady flow, the loss graphs look similar, meaning they are only shown here.

Moving on to the predictions, in Figure 6.2 the PINN's predicted flow fields are seen next to the DNS data, where the absolute difference is shown on the right. It can be seen that the prediction of the density, pressure and temperature are in general good agreement with the actual fields found from the DNS, with some small regions with fairly higher error close to the pool (Figure 6.2a, 6.2b and 6.2c). These fields should be accurate because these are based on the provided flow states, as the PINN is trained on these. The reason for the increased error in some regions, especially for the density and temperature, can be explained by the fact that the network is trying to both minimize the loss caused by the difference between the prediction and the provided data and the loss coming from the residuals of the governing equations (during training of only the measurements, the predictions of these flow states had a much lower error).

However, the more interesting results are that of the retrieved flow states that were not provided to the PINN: the axial and radial velocity u and v, for which the contours can be seen in Figure 6.2d and 6.2e. The axial velocity is predicted well in general, except in the region of high fuel density, just above the pool, where the highest error occurs. Furthermore, there is a region of increased error at greater



(a) Measurement loss over iterations

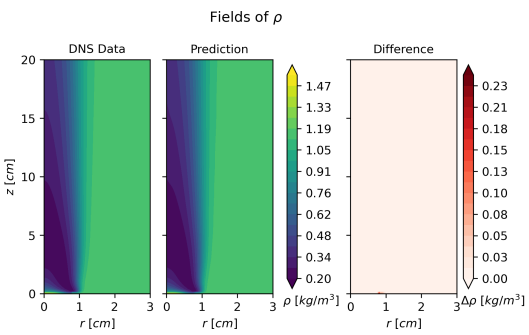
(b) Physics loss over iterations

Figure 6.1: Measurement and physics loss over iterations for the two training phases

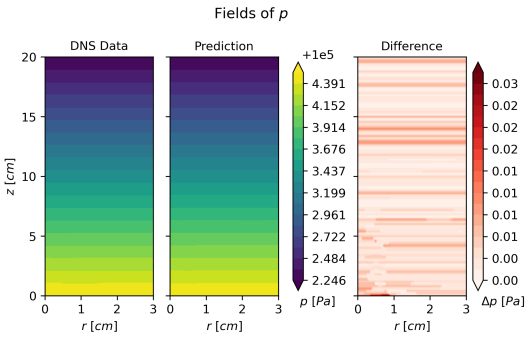
radius and low height. The maximum error of the prediction is low compared to the order of magnitude of the axial velocity. Furthermore, the main features are captured and the maximum axial velocity predicted is correct. For the radial velocity, although the prediction has similar features compared to the data from the DNS, it is clear that the feature shapes are not captured in full detail. Furthermore, the maximum (absolute) radial velocity is not captured correctly, being the region with highest error. The error is of the same order of magnitude as the maximum radial velocity.

The PINN has difficulties with capturing the steep gradients and strong peaks surrounded by nearly zero gradients and values. This can be clearly seen in case of a line plot, as seen in Figure 6.3a and 6.3b, where the axial and radial velocity are plotted over the height z at a radius of 0.50 times the pool diameter ($r = 0.8cm$). The axial velocity is almost completely correctly captured, only showing a slight difference near the pool, being in the region with maximum error in Figure 6.2d referred to earlier. The radial velocity line plot is captured fairly well, except close to the pool where the prediction has an extreme value only being a fraction of the actual extreme value in the flow radial velocity. This region of high error is the spot with maximum error referred to before in Figure 6.2e. Not being able to capture strong gradients is rather inherent to a NN, as the NN can only capture functions up to a certain level of complexity. In case the function to be captured is more complex than the NN can replicate, a trade-off to keep the loss function as low as possible is made, meaning a prediction with high error in a small region in case the majority of the rest of the predicted field has an extremely low error is preferred over a prediction with fairly low error over the full domain. This occurs as the latter would tend to have a higher MSE, which the PINN aims to minimize. A clear visual representation of this can be seen in Figure 6.4, where the sum of the residuals of the mass and momentum equations is plotted over the domain, indicating only one clear spot with extremely high residuals, whereas the rest of the domain has residuals at least one order of magnitude lower. The region with high residuals coincides with the region with high error in Figure 6.2d and 6.2e as one would expect.

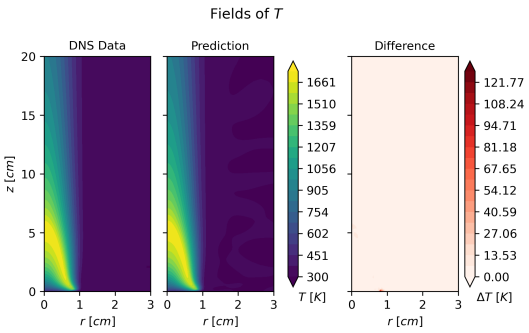
A short note should be made about the collocation points. These were decided to be at the same coordinates as the measurement points, as mentioned in chapter 5. However, the author decided to investigate what would happen if collocation points with a smaller distance between them would be chosen, for example collocation points at measurement points and halfway between them. The main reasoning behind this is that residuals need to be zero everywhere, not just at the measurement points, and one can force this more strongly by adding more collocation points. It was thought that this might lead to peaks and strong gradients being captured slightly better, but it became clear that it hardly influenced the results at all, being disregarded and not included in this research.



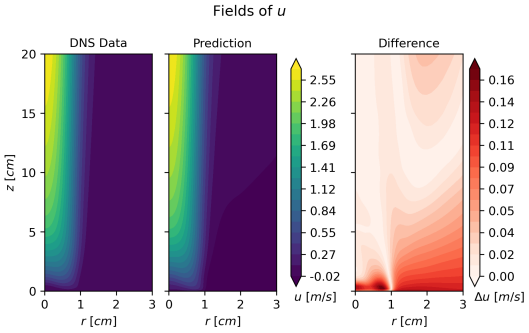
(a) Accuracy of prediction of field of ρ



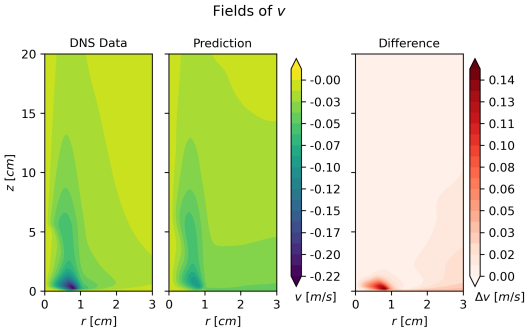
(b) Accuracy of prediction of field of p



(c) Accuracy of prediction of field of T

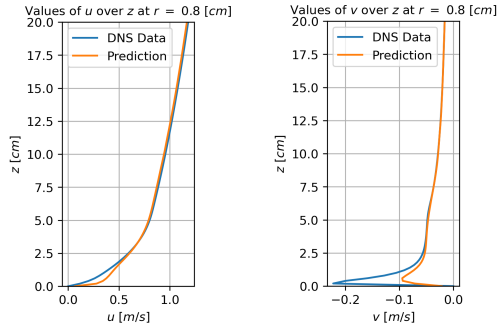


(d) Accuracy of prediction of field of u



(e) Accuracy of prediction of field of v

Figure 6.2: Performance of PINN in predicting flow fields in case of a steady flame



(a) Axial velocity (b) Radial velocity

Figure 6.3: Predicted velocity distribution over height z at radius 0.50 times the pool diameter ($r = 0.8\ cm$) of steady flame compared to DNS data, and sum of residuals of the mass and momentum equations

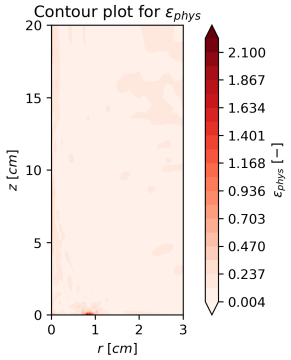


Figure 6.4: Combined residuals of the mass and momentum equations over the domain in case of the standard case in steady flow

6.1.2. Effect of Coarsening

In this subsection, the effect of coarsening the data coming from DNS on the performance of the PINN is considered. The purpose of this is to investigate the required spatial resolution to be able to use the HFM framework.

In Table 6.1, the MSEs of the cases with different coarsened data can be seen, where the coarsening factor refers to the factor with which the spacing between points has been multiplied everywhere in the domain. It can be seen that when coarsened data is provided, the predictions generally have an increased MSE. The MSE of the axial velocity increases a lot whereas the one for the radial velocity does not increase that much, as the magnitudes of the radial velocity are generally low, meaning even a prediction of zero radial velocity everywhere would have a MSE of only slightly higher order of magnitude. In general, it holds that the MSE increases when coarsening the data, as you are removing information, making it harder for the PINN to predict the right fields. However, changing the provided data also changes the optimization domain and could potentially change the performance slightly as well. Furthermore, the optimization process also has quasi-random behaviour, as the data the PINN sees is given in a different order every time, meaning that it can reach different local minima for the loss when rerunning with the same provided data. This could be the cause of the decrease in MSE of the axial velocity in the last row. It could be that just by coincidence a local minimum which has weights and biases that predict the axial velocity better is found for the last coarsening factor compared to the row above. This would mean that the MSE of the axial velocity would indeed be lower for the last case. Looking at the MSE of the radial velocity, it is seen that it increases when comparing the case with coarsening factor of 2^2 to the case with coarsening factor of 2^3, as expected. It could be that this increase in MSE for the radial velocity still caused the total loss function of the last row to be higher than that of the row above, meaning that the full physics loss would still be higher for the last row as one would expect when using data with stronger coarsening. It should be noted that this could be possible as the loss function is not directly related to the errors in u and v, but rather to the residuals of the mass and momentum equations, together with the difference between the predicted fields and provided fields of the density, pressure and temperature).

As the case with no coarsening has already been shown, and showing intermediate cases is not necessary to see the effect of coarsening, it is decided to show the predictions with the highest coarsening still having flow fields which can be recognized from the DNS data, being that with coarsening 2^2.

Table 6.1: Comparison of MSE of u and v in case of various levels of coarsened data

Case	MSE u $[(m/s)^2]$	MSE v $[(m/s)^2]$
Coarsening 2^0	0.0025	0.00033
Coarsening 2^1	0.0065	0.00033
Coarsening 2^2	0.0221	0.00056
Coarsening 2^3	0.0121	0.00065

In Figure 6.5, the performance of the PINN is investigated in case of providing coarsened data with

spacing 2^2 larger than in the original DNS, meaning the amount of points decreased by a factor of 16, and the lowest spacing between points found in the domain is approximately $0.5mm$. It is seen that the prediction of the density, pressure and temperature show similar behaviour as without coarsening, namely showing small regions with slight errors, as seen in Figure 6.5a, 6.5b and 6.5c. This is for the reason mentioned before: the PINN tries to optimize for the measurement loss and the residuals of the governing equations at the same time, meaning that it accepts an increase in measurement loss if it would lead to a stronger decrease in residuals of the governing equations used. During the training of only the measurement data, the fields with higher coarsening have higher error than when no coarsening has been used, as expected.

When looking at the velocity fields in Figure 6.5d and Figure 6.5e, one can see the main features being captured. For the axial velocity, the error that was present in the standard case also appears here: near the fuel inlet there is a high error. However, there is an even higher error caused by the axial velocity gradient in the r direction in the prediction which is not there in the DNS data. This causes an increasing error for increasing r. The reason for this radial behaviour of the prediction is not known, but a reason might be that the prediction reached a local minimum where it could fit the data quite well already and it would not move away from this minimum anymore. This could be argued as there are quite some radial gradients present in the domain, especially for low r and high z, meaning it could cause a local minimum. Furthermore, as the DNS data has less points in the far field already, and the data is coarsened, the number of points in the far field is very low. This could cause that the far field hardly affects the loss function, meaning high residuals in the far field are not punished as heavily as residuals close to the flame. For the radial velocity component, the plot looks very much like the prediction of the standard case, with the peak being captured slightly less accurately (lower peak, higher error). This could be reasoned by the decreased amount of points, causing it to be harder to estimate the gradients for the PINN, meaning the residuals are inaccurate, causing the results to contain higher error.

6.1.3. Effect of Noise

In this subsection, the effect of adding artificial noise to the data provided to the PINN on the performance of the PINN is considered. This has the purpose of investigating how prone the framework is to noise. The noise has been artificially generated using a Gaussian distribution with zero mean and noise power depending on the SNR set. For more background information about the construction of noise, one can refer to section 5.2.

In Table 6.2, the MSEs of the cases with artificially added noise with various SNRs to the provided data can be seen. It is found that the predictions generally get worse when more noise is added (lower SNR). The top row refers to the case when no noise is added. Interestingly, the MSE of the radial velocity is hardly effected by the added noise, whereas the MSE of the axial velocity strongly deteriorates when the SNR decreases. Interestingly, the case with no noise has a higher MSE for the axial velocity than the case of SNR $= 60dB$. The reason for this is unclear and might have to do with the fact that the optimization domain changed or can be caused by the fact that a different, better, local minimum has been found by coincidence (the data is provided in quasi-random order, meaning that every time you rerun the code, a different local minimum could be found). As the case with no noise has already been shown, and showing intermediate cases is not necessary to see the effect of noise, it is decided to show the predictions with the highest noise still having flow fields which can be recognized from the DNS data, being that with SNR $= 50dB$.

Table 6.2: Comparison of MSE of u and v in case of various levels of noise

Case	MSE u $[(m/s)^2]$	MSE v $[(m/s)^2]$
SNR: -	0.0025	0.00033
SNR $= 60dB$	0.0020	0.00038
SNR $= 50dB$	0.0143	0.00032
SNR $= 40dB$	0.0569	0.00035

In Figure 6.6, predicted fields coming from the PINN are shown in case of artificially adding noise with a SNR of $50dB$ to the provided data. It is clear that the predictions of density, pressure and temperature seen in Figure 6.6a, 6.6b and 6.6 have hardly changed. This is because the noise power is very low,

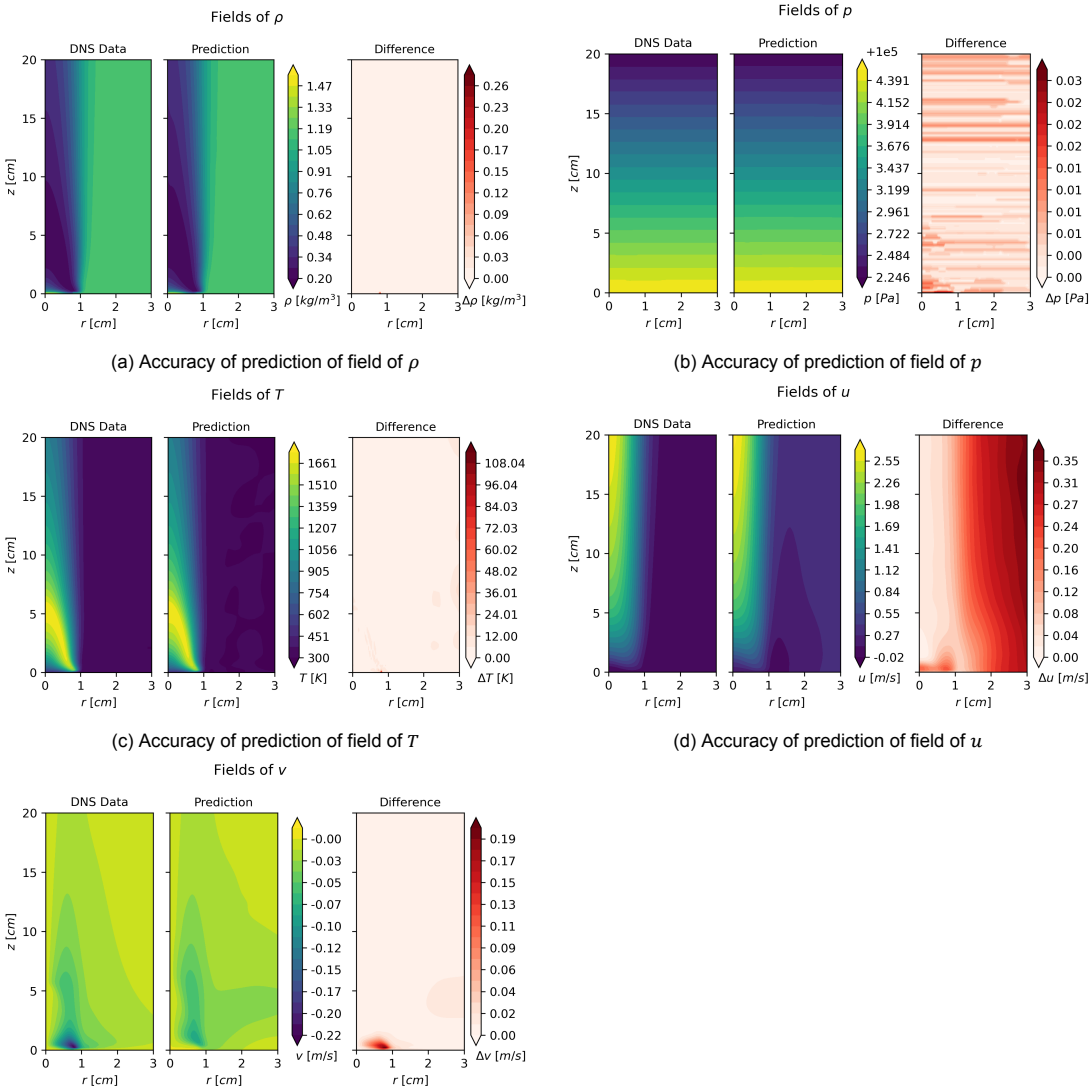
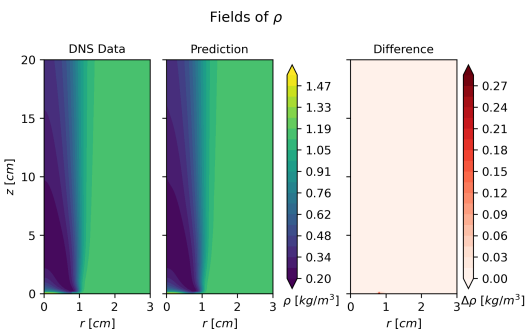


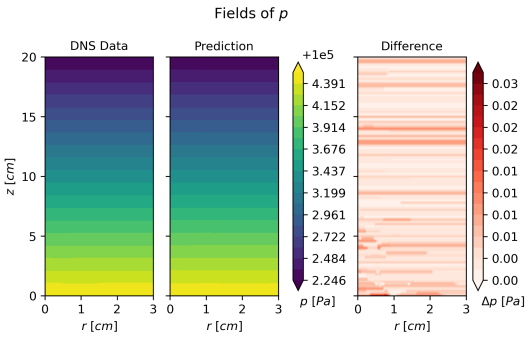
Figure 6.5: Performance of PINN in predicting flow fields in case of a steady flame with coarsened data by factor 2^2

making the effect of the noise in these results practically invisible. Similarly to when coarsening the data, the predictions of these provided flow states did deteriorate during the time it only trained on these provided states, but the effect of optimizing also for the residuals of the governing equations has a stronger deteriorating effect making the previous effect not visible.

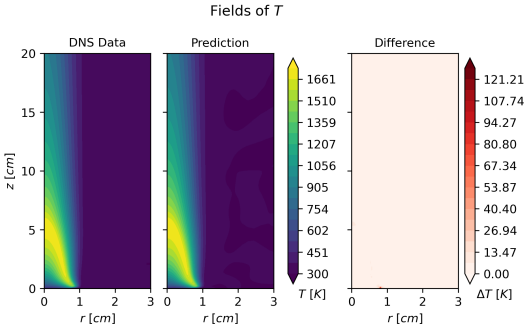
Moving to the axial velocity prediction seen in Figure 6.6d, it is found that the error has increased significantly. Similar to the behaviour of coarsening, the axial velocity shows a gradient in radial direction which is not present in the DNS data again, causing the highest error to be in the far field. The high error spot near the pool still appears as before. In Figure 6.6e, the effect of noise on the prediction of the radial velocity can be seen. The prediction is almost identical to the one in case of no noise, which could also already be seen in Table 6.2, as the MSE of the radial velocity hardly changed for increasing noise levels. The provided noisy fields with SNR $= 50dB$ can be seen in Figure 6.7, where the difference between the actual fields and the noisy fields are also shown. It is seen that the noise level is a major number of orders of magnitudes smaller than the actual fields.



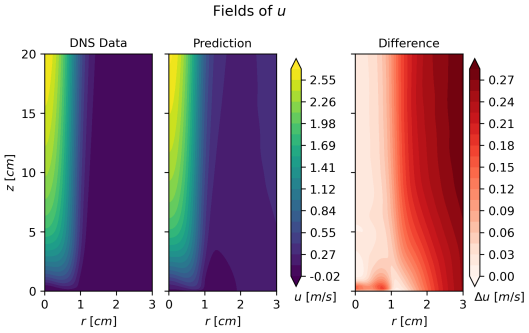
(a) Accuracy of prediction of field of ρ



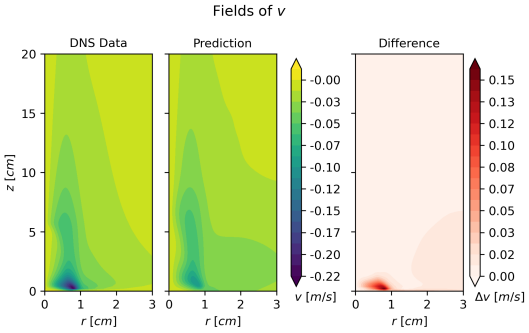
(b) Accuracy of prediction of field of p



(c) Accuracy of prediction of field of T



(d) Accuracy of prediction of field of u



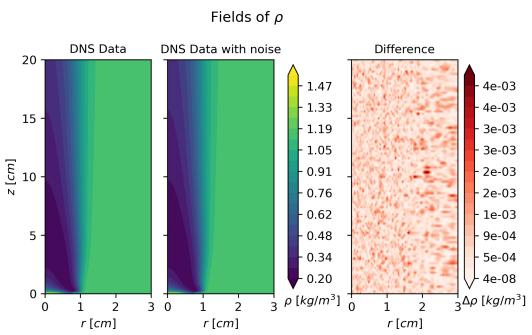
(e) Accuracy of prediction of field of v

Figure 6.6: Performance of PINN in predicting flow fields in case of a steady flame with artificially added noise with SNR = $50dB$

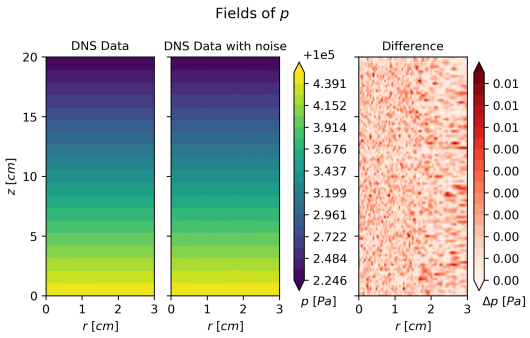
6.1.4. Effect of Reducing Provided Flow States

In this subsection, the effect of providing less flow states to the PINN on its performance is considered. This is significant as it can give an indication on what the minimum number of required flow states to be measured has to be in particular cases. From an experimental point of view, it would be most beneficial if only a few flow states have to be provided.

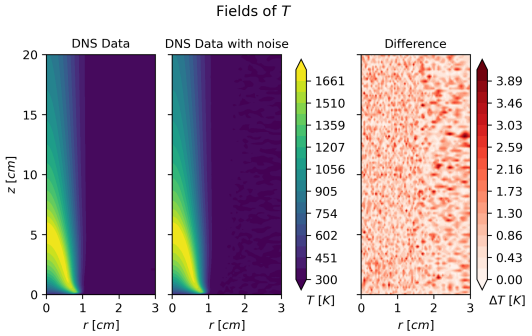
In Table 6.3, the MSEs of the predictions when removing provided flow states can be seen. To start, the MSEs for the standard case with provided density, pressure and temperature is written down, after which in the next row the density is removed. It is seen that the MSE of both the axial and radial velocity increase excessively, meaning the PINN is not robust to providing less flow states in this particular flow. The predicted flow fields when providing only the pressure and temperature look nothing like the actual flow fields, even though the loss function converged to a significantly low number. Remember that the uniqueness of this reconstruction problem has not been proved even in case of providing the density, pressure and temperature all-together. This means that when removing even more information (in this case the density field), it is possible the optimizer converges to a minimum of the loss which is not the actual solution of the flow: in that case, there are multiple options for the loss to converge to, meaning that there is not a unique solution satisfying near zero residuals and near zero measurement loss. The



(a) Noisy field of ρ



(b) Noisy field of p



(c) Noisy field of T

Figure 6.7: Noisy fields provided to the PINN in case of SNR = 50

PINN in its current form is thus not robust to removing one or more thermodynamic flow states from the provided data, as the MSE of both the axial and radial velocity increased disproportionately when doing so, which might be caused by the solution not being unique anymore.

However, there are some physical assumptions one could make to improve the results in case the density or even both density and pressure are removed from the provided data. When only removing one state, one should note that the ideal gas law might be a good assumptions to start with, stated in Equation 6.1. In this, p is the pressure, ρ the density, T the temperature and R the specific gas constant depending on the composition of the gas.

$$p = \rho R T \tag{6.1}$$

Other than the already known limitations of the ideal gas law (that it does not take into account the molecular size and attractions between molecules), it uses a specific gas constant R, which is based on the composition of the gas. In a flame, the reactions cause the composition of the gas to change in space. This would mean that an estimate of the compositions of the gas at all locations is needed to find the exact R, which seems unrealistic in case the density can not even be found. For this reason, the specific gas constant is assumed to be a constant over the whole space domain, accepting the errors related to this. Then, the difference between predicted ρ and the guess for ρ can be added to the loss function to be minimized as seen in Equation 6.2, where p_{data} and T_{data} are the pressure and temperature from the provided data. This points the PINN in the right direction, albeit with a rough assumption especially close to the flame.

$$\Delta\rho = \rho_{pred} - \frac{p_{data}}{R T_{data}} \tag{6.2}$$

Using this method, one achieves the MSEs shown in the third row of Table 6.3, which is a great improvement compared to just taking the pressure and temperature, and only a minor deterioration compared to having the density available. In case both the pressure and the density are removed, an extra assumption is required as without this, similar MSE values to the case of only having pressure and temperature would be obtained. As mentioned in chapter 5, the pressure field is almost purely governed by a constant pressure gradient in the z direction. This means one can assume a constant

pressure gradient from Equation 5.6. The problem is that this gives a differential relation, meaning that it can only be solved up to a constant. Furthermore, the density occurs in the formula for the gradient, which is also not known from the data. These issues are solved in the following manner. First, a constant ambient pressure is assumed to be able to calculate ρ using the ideal gas law. After this, the far field value of ρ from the ideal gas law is used in Equation 6.3 to obtain the pressure in the whole domain. This assumes that the value of the pressure at the top of the domain equals the ambient pressure, which is a given in the experiment.

The reason for these order of steps is that the pressure change is relatively low compared to the pressure value, meaning that the ρ from the ideal gas law would hardly change due to this low change in pressure. Then, the far field density is used as this is the density which would cause the gradient in pressure over altitude in the case of stagnant air. The extra loss term to be added to the total loss is written in Equation 6.4. Using this method, the results of the last row of Table 6.3 are found, where it is seen that the MSE of both axial and radial velocity have increased, which is reasonable as many assumptions had to be made to come to the result. As the main flow features are still even recognizable in the predicted fields of the last case, these are to be shown. The case where only the density is removed and assumed from the ideal gas law is not shown because it shows very similar, slightly better results. As removing the density (or pressure) and not helping the PINN by using any assumptions leads to extremely high MSEs and extremely wrong predictions in which no physical structures can be recognized anymore, that case is also not shown.

$$p = p_{ambient} - \rho g(z_{max} - z) \tag{6.3}$$

$$\Delta p = p_{pred} - (p_{ambient} - \rho g(z_{max} - z)) \tag{6.4}$$

Table 6.3: Comparison of MSE of u and v in case of decreasing provided flow states

Case	MSE u $[(m/s)^2]$	MSE v $[(m/s)^2]$
ρ, p, T	0.0025	0.00033
p, T	1.2418	0.00160
ρ_{guess}, p, T	0.0068	0.00038
$\rho_{guess}, p_{guess}, T$	0.0113	0.00047

In Figure 6.8, the performance of the PINN is shown in case of only providing the temperature field but assuming the pressure and density field from the method briefly mentioned before. For the density and pressure it is seen in Figure 6.8a and 6.8b that the main flow features are almost fully correctly captured. However, it is generally seen that the error increased compared to the case where these flow states were provided. This is explained by the fact that these flow states are based on the assumptions of ideal gas and constant pressure gradient rather than the actual measurements. Therefore, an increase in error when using these assumptions is to be expected. Furthermore, the region of high error is near the pool, where the assumption made on a constant specific gas constant, namely the one of air, is inaccurate as there the flow mainly consists of fuel, having a different specific gas constant than the one of air which is used. For the temperature, it is seen that the prediction has lower error than when providing all flow states, as can be seen in Figure 6.8c. This might be explained by the fact that the functions for the density and pressure the NN has to learn are simpler than when using actual data, meaning that the NN can predict the temperature field up to a slightly higher complexity, thus better.

The axial velocity field prediction can be seen in Figure 6.8d, where the highest error occurs near the pool, mainly caused by the fact that the density and pressure are also wrongly predicted in this region. Furthermore, the radial gradient in axial velocity discussed before occurs here as well. For the radial velocity, the field is very similar to the standard case, where the highest error is just above the pool. Interestingly, the prediction of the radial velocity (Figure 6.8e) is hardly affected near the origin, as the error did not increase drastically there, which it did for the axial velocity because of the error in density and pressure related to the wrongly assumed gas constant.

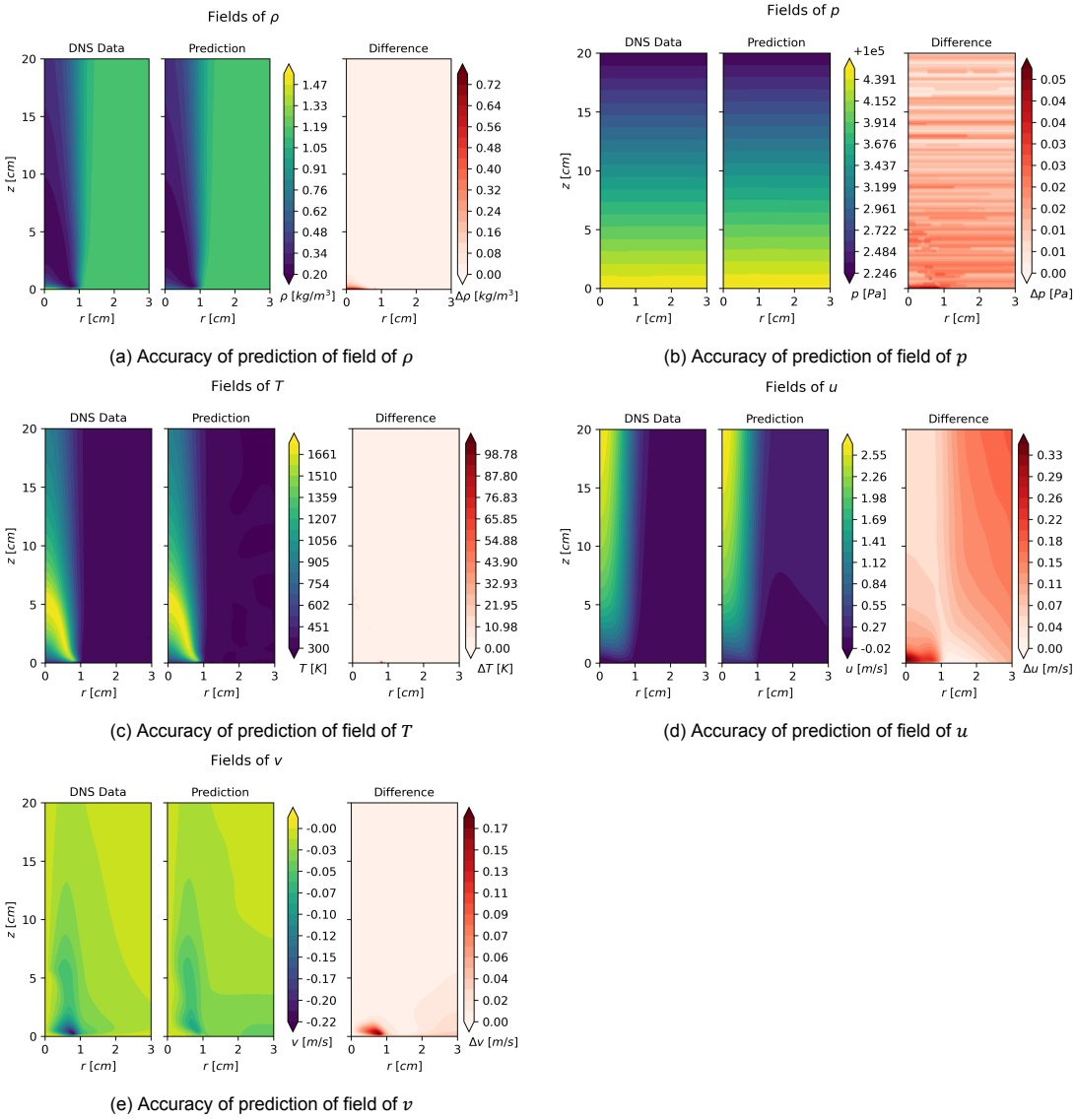


Figure 6.8: Performance of PINN in predicting steady flow fields in case of providing only the temperature field and assuming pressure and density from the hydrostatic equation and the ideal gas law respectively

6.1.5. Predictions on equally spaced grid

As the previous results are all generated using the points from the DNS results, the provided data has always been fine near the flame and coarser far away. However, in real measurements, the spatial resolution is the same throughout the measurement window. For real-life applications, one would thus be training the PINN using an equally spaced grid. Therefore, this subsection discusses the effect of using an equally spaced grid instead of the grid coming directly from the DNS on the performance of the PINN. Note that using an equally spaced grid also means the residuals are computed at these equally spaced points, as it has been decided to set the collocations points equal to the measurement points.

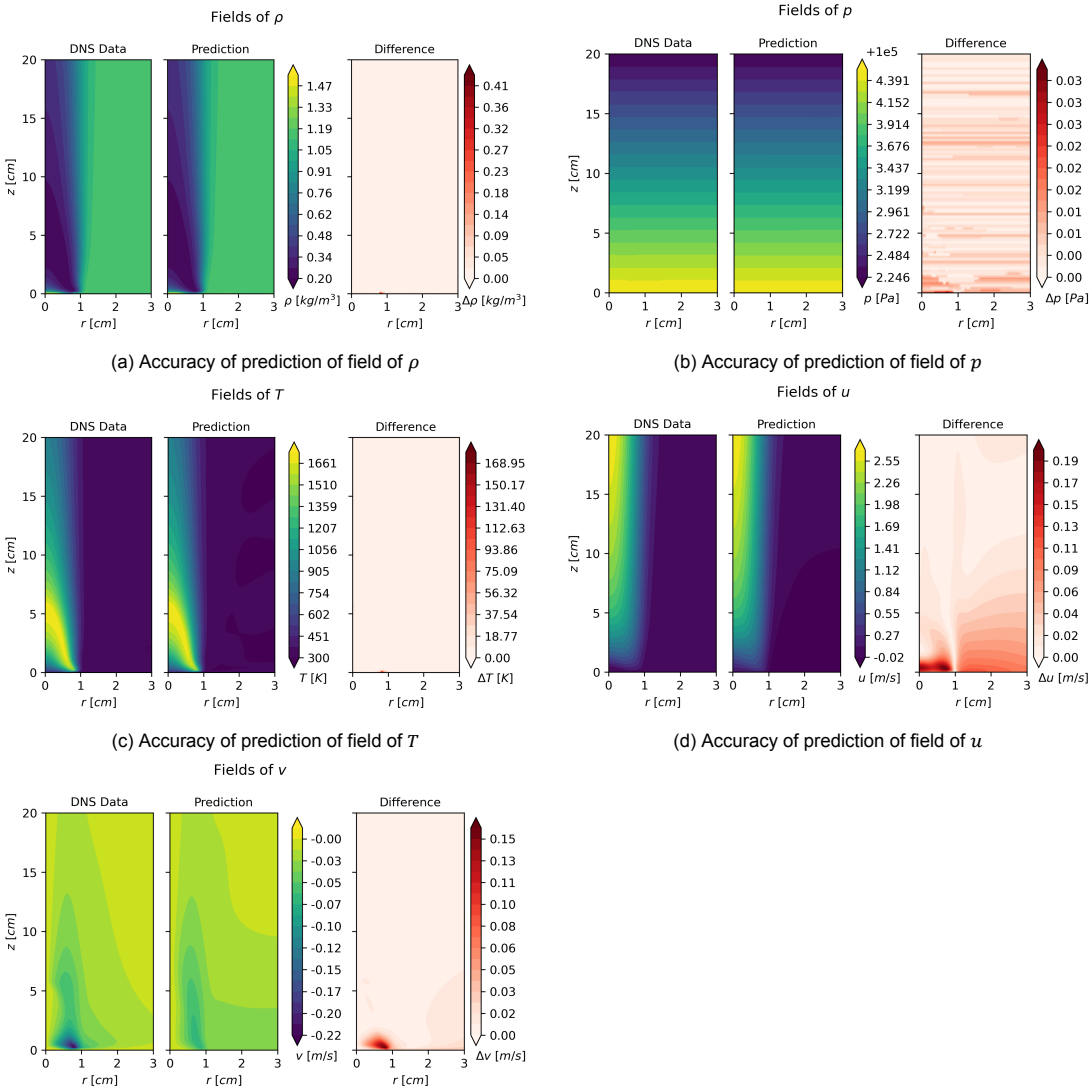
When using the equally spaced grid, the proportion of points being in the (constant) far field increases greatly. This makes the PINN more prone to converge to an erroneous constant solution, which is a substantial disadvantage of the use of HFM. To understand why this occurs, one needs to realize that a constant field has zero or almost zero residuals everywhere, and the measurement error is zero in the far field (if the correct constant is chosen) and high only in the flame. In case the same spacing is used in the far field as in the near field, and knowing the loss is based on the mean square error, the measurement loss is going to be very low. This is due to there being only a small region with high error, and an extremely high area with low error, causing a very low measurement error. This in

combination with the close to zero residuals of the governing equations everywhere in the domain in case constant fields are predicted, makes the total loss to be extremely low. Without adjusting something, the PINN can thus reach an extremely low loss by just predicting constant fields. This means the PINN fails to predict the correct fields or even get close to these.

One way to prevent this is to remove a lot of points out of the equally spaced grid in the far field. This would increase the inherent weight of the near field due to the amount of points compared to the amount of points in the far field. This then would lead to a higher loss value when the prediction is a constant, as the small region of high measurement error would then have a high weight, due to it being located in the near field. This means the PINN will not converge to a constant anymore. Another way to solve this issue is by adding artificial weighting to the mean square error between the predicted and provided fields, meaning that regions in the far field get assigned a low weight and regions in the near field a high weight. The latter method is thought to be easier to implement and more natural, thus this has been used to still be able to use the PINN with the equally spaced provided data. The weights are generated using an exponential function being high near the origin and low further away, and the parameters that can be altered to change the strength and shape of the exponential function are tuned for optimal performance.

As mentioned, the reason for investigating the effect of using equally spaced data on the performance of the PINN is that real-life measurements usually have an equal spacing. As for real-life measurements the spatial resolution required is of high importance, it is valuable to check if the spatial resolution required using the equally spaced data is similar to when the points from the DNS are used. For this reason, the data is coarsened in the same manner as in subsection 6.1.2. The grid that is considered to be not coarsened is that with the same equal spacing everywhere which is equal to the lowest spacing in the grid from the DNS.

Predicted fields when coarsening factor 2^3 is used are shown in Figure 6.9. It is clearly seen that the results are very similar (compared to Figure 6.5), and even slightly better as in this case a factor of 2^3 has been used, meaning a spacing between points of $1mm$, whereas it was $0.5mm$ before. The main reason for this could be that the far field before had a way higher spacing: the lowest spacing was $0.5mm$, but the points were not equally spread, meaning the far field had extremely coarse points, whereas now the spacing is equal everywhere, meaning the far field does not have such extreme coarse data. The MSE of axial and radial velocity fields are 0.0024 and 0.00044 respectively. This shows that an equally spaced grid can indeed be used as well.



(a) Accuracy of prediction of field of ρ

(b) Accuracy of prediction of field of p

(c) Accuracy of prediction of field of T

(d) Accuracy of prediction of field of u

(e) Accuracy of prediction of field of v

Figure 6.9: Performance of PINN in predicting flow fields in case of a steady flame with equally spaced grid and coarsened data by factor 2^3

6.2. Unsteady Flame

In this section, the performance of a PINN of 20 layers of 400 neurons in retrieving the velocity fields in case of an unsteady flame is investigated. Firstly, this is done by using the full data set of the density, pressure and temperature from the DNS in subsection 6.2.1. Afterwards, the effect of coarsening the data in space on the performance is considered in subsection 6.2.2. Then, the robustness to noise is researched in subsection 6.2.3. After this, the effect of reducing the number of provided flow states is inspected in subsection 6.2.4. Finally, a short remark is made about the possibility to provide data in an equally spaced grid in subsection 6.2.5.

6.2.1. Standard

As the data consists of 400 time snaps, it is impractical to look at the performance at all these time snaps. It is known that the flow has periodic effects, so one would be interested to know if the MSE also shows periodic behaviour.

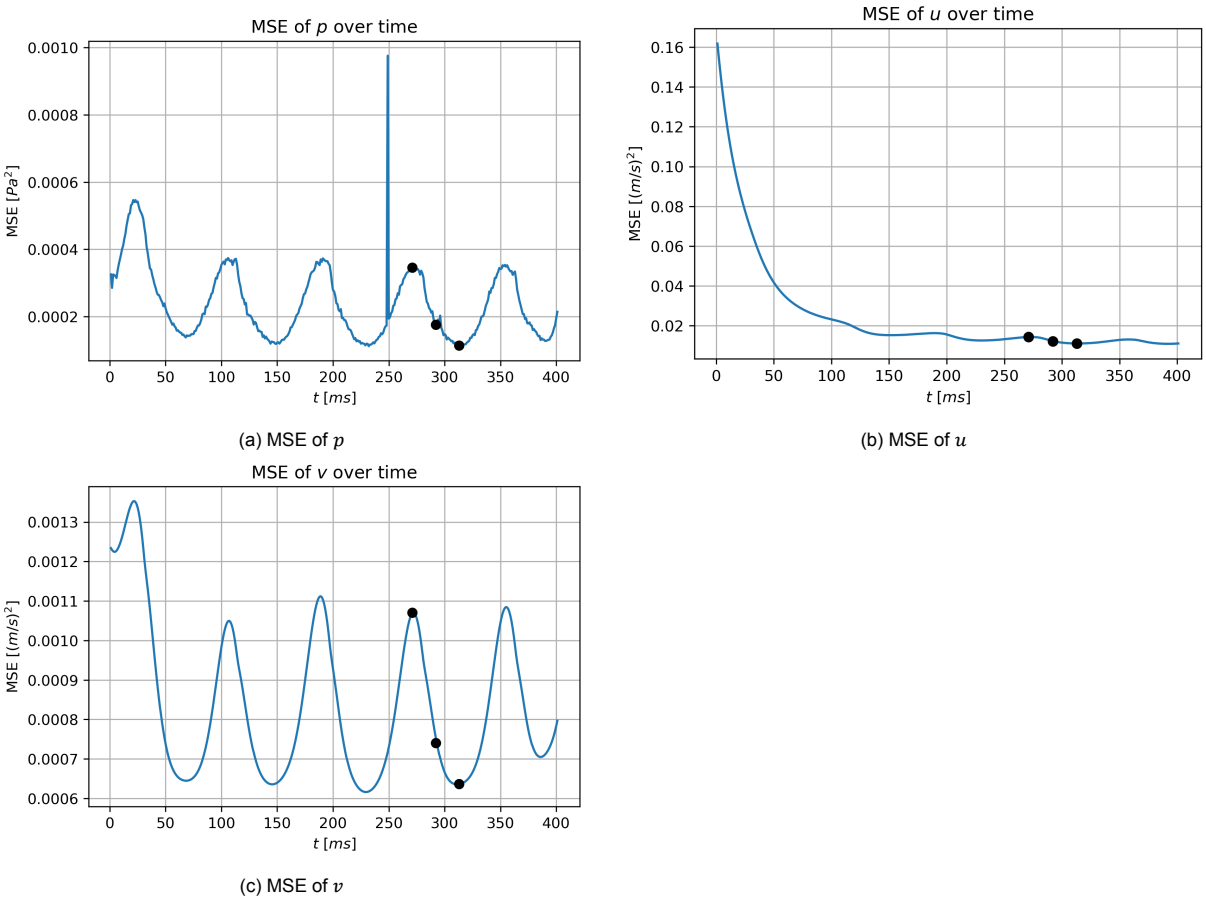
When looking at Figure 6.10, it is indeed true that the MSEs of the pressure, axial velocity and radial velocity show periodic behaviour, having the exact same period as present in the flow. The periodicity in the MSEs can be explained by the fact that the flow is unsteady and contains periodic effects (puffing):

the periodicity is not captured 100% by the PINN's prediction, meaning that a periodic error occurs. The pressure has an extremely low MSE compared to the order of magnitude of the pressure values in the domain. This is because the pressure is provided as a measurement. Interestingly, the pressure shows periodic behaviour with the same frequency as the puffing of the flame in the flow, where the density and temperature do not show this behaviour (not shown for this reason). It could be that the density and temperature have an unsteady component in the MSE, but that the amplitude is so small that it is not visible in the plots. Furthermore, the pressure predictions after only training on the measurement also do not show this behaviour. There seems to be a strong coupling present between the pressure and the velocities, meaning that the periodic error in the velocities would go together with a periodic error in the pressure as well.

In general, the axial velocity field has a relatively low MSE compared to the order of magnitude of the axial velocity value. It is seen that for low time values, the MSE is high. This effect can mainly be explained by the fact that the PINN is not able to accurately compute the residuals of the governing equations as the time derivatives can not be computed accurately (there is no information for $t < 0ms$). After approximately $200ms$, this effect almost fully disappeared. The radial velocity field generally has a relatively high MSE compared to its order of magnitude. The effects discussed for the axial velocity component can also all be seen here. Firstly, the high MSE for low times can be seen, albeit almost hidden as the periodic effect is far more prominent. The periodic effect is so much more visible as the radial velocity is almost fully periodic: most features that the PINN can or cannot capture are periodic, meaning that the error shows strong periodic behaviour. In contrast, the axial velocity component can be decomposed in a mean and fluctuating component, meaning that the MSE shows a weaker periodic behaviour (a large portion of the MSE is not periodic, meaning the amplitude of the periodicity is much smaller than the mean MSE value). The strong unsteadiness of the radial velocity can also be used as an argument to why the MSE of the radial velocity is relatively higher than the one for the axial velocity, even though this is also partly explained by the steep gradients being present in the radial velocity fields. Lastly, the three dots indicate the maximum and minimum MSE, and a value in between, corresponding to $t = 271ms$, $t = 313ms$ and $t = 292ms$, respectively. As the MSE shows periodic behaviour, it is decided to only show the predictions at these times snaps, because those already represent the behaviour of the prediction over the full time domain well as the predictions at other time snaps will be repetitions or interpolations of these.

In Figure 6.11, the predicted flow fields can be seen next to the fields from the DNS data, and the absolute difference is plotted at the time snap of $t = 271ms$ (highest MSE). Looking at Figure 6.11a, Figure 6.11b and Figure 6.11c, it is clear that the density, pressure and temperature are predicted accurately, with relatively small errors. For the density and temperature the error is highest close to the pool, as was also seen in the steady case discussed in section 6.1. However, for the pressure fields, an alternating pattern of increased error occurs which was not present in the steady flow. This can be attributed to the strong unsteady effects caused by vortices in these regions. The PINN partially attempts to minimize the residuals of the NS equations, where the pressure (gradient) is strongly coupled with the velocities, causing the predicted pressure to change and thus have higher error.

For the axial velocity fields seen in Figure 6.11d, the prediction matches the field from DNS data well overall, taking into account that no information at all has been provided about the velocity. However, a number of differences can be observed in the details. Firstly, it can be seen that the local maximum and minimum around radius and height 0.00 and 3.14 times the pool diameter ($r = 0cm$, $z = 6cm$) and radius and height 0.00 and 4.19 times the pool diameter ($r = 0cm$, $z = 8cm$) are not captured to the full extent, where the prediction seems to capture the peaks as less extreme values (lower maximum and higher minimum). In fact it holds generally that small regions in the domain with steep gradients are not fully captured. This is even more clear in case of the radial velocity component subsequently. It might be explained by the following reasoning. The current NN architecture (mainly size) might not allow for functions being complex to the level of having very steep, high peaks, without altering the results elsewhere in the domain. In principle, when increasing the NN size (amount of neurons) towards infinity, the NN would be able to represent such, more complex functions. As the NN is not able to represent the correct mapping function for the velocity field everywhere, the optimizer accepts the higher error in this small region, as it leads to a minimum in MSE which, as its name implies, is a mean over all square differences. As a small region would have few points, it is weighted less and the MSE increases only slightly if such a small region would have a high error. It should also be noted that the region with the local maximum and minimum not being fully captured shows highly unsteady behaviour, making



(a) MSE of p

(b) MSE of u

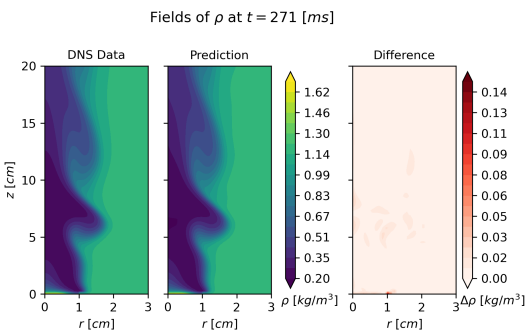
(c) MSE of v

Figure 6.10: MSE of pressure, axial velocity and radial velocity over time for the standard case

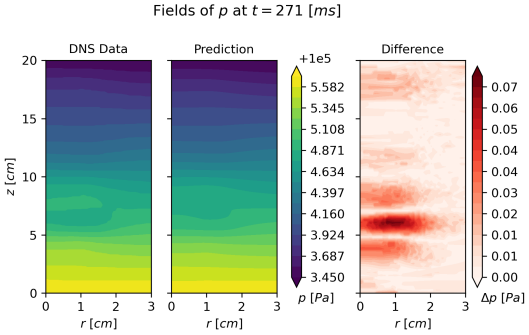
it harder to predict as well. The last main difference between the prediction and field from DNS data of the axial velocity is that the prediction overestimates the velocity at the bottom of the flame, around radius and height 0.52 and 0.00 times the pool diameter ($r = 1cm$, $z = 0cm$). The regions mentioned can all be seen in the absolute difference, where the highest difference occurs at the not fully captured local minimum in velocity, causing a local error of approximately 10% of the maximum value of the velocity.

In Figure 6.11e, the results for the radial velocity component can be seen at time instant $t = 271ms$. It is clear that the predictions in terms of the radial velocity are not nearly as accurate as the ones for the axial velocity: the maximum error is the same order of magnitude as the fluctuations. However, when looking at the main features in the flow field, they are mostly captured: even though the values in these regions are not correct, the patches of highly positive/negative values are similarly shaped in the predicted field when comparing to the field from DNS data. The two main regions of high error are near the highest velocity around radius and height 0.52 and 3.66 times the pool diameter ($r = 1cm$, $z = 7cm$) and at the bottom of the flame around radius and height 0.52 and 0.52 times the pool diameter ($r = 1cm$, $z = 1cm$). These are again regions with very high extreme values with associated steep gradients and highly unsteady effects compared to the rest of the field, which the PINN struggles to predict correctly for the reason described before. These regions of high error are strongly connected to the regions of high error in the axial velocity field and pressure field, as these errors are all related to the high extremes caused by the puffing phenomenon and thus occur at the same frequency (in time, thus also in space as they rise).

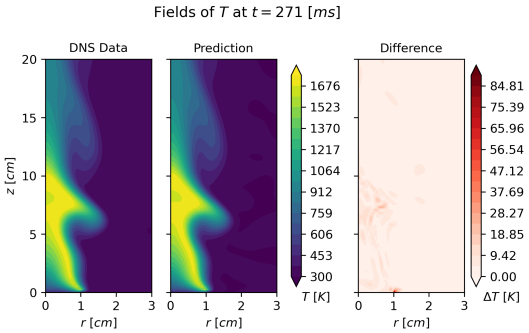
Many statements made for Figure 6.11 can also be made for Figure 6.12, which shows the behaviour of the predictions at time instant $t = 292ms$ (time snap with intermediate MSE). The most important differences are caused by the vortices rising and dissipating over time. This means the pockets of high error corresponding to these vortices have now moved up and become lower in value. This can



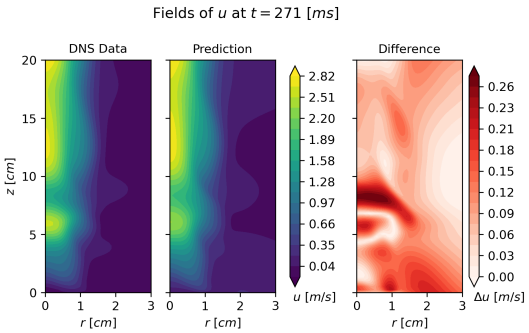
(a) Accuracy of prediction of field of ρ



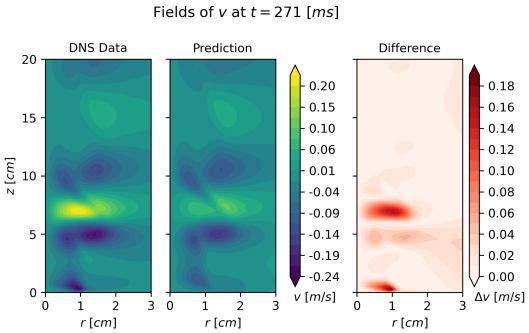
(b) Accuracy of prediction of field of p



(c) Accuracy of prediction of field of T



(d) Accuracy of prediction of field of u

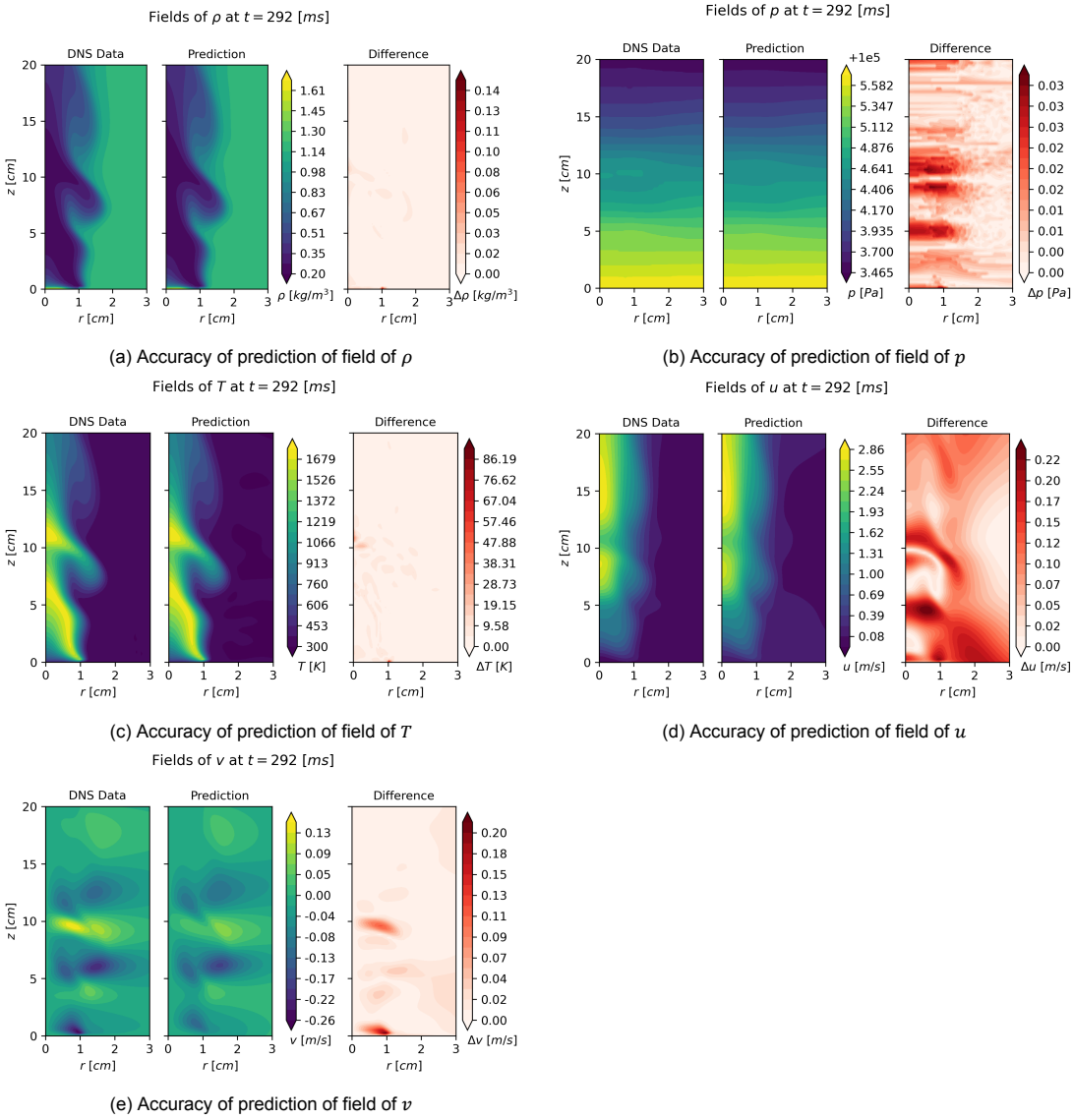


(e) Accuracy of prediction of field of v

Figure 6.11: Performance of PINN in predicting flow fields in case of an unsteady flame at $t = 271ms$

be explained by the fact that the peaks with steep gradients have dissipated, meaning the fields are easier to capture by the PINN. Furthermore, a new spot of high error appeared at radius and height 0.31 and 2.09 times the pool diameter ($r = 0.6cm$, $z = 4cm$) in the pressure and axial velocity fields seen in Figure 6.12b and 6.12d, which is caused by a new local minimum axial velocity originating in the DNS data. For the radial velocity field prediction shown in Figure 6.12e, it is seen that the maximum error increased. This is caused by the highly negative radial velocity spot near the bottom of the flame around radius and height 0.52 and 0.52 times the pool diameter ($r = 1cm$, $z = 1cm$) becoming more negative and increasing in size. This means the spot becomes harder to capture, thus being the cause of an increase in error. It should be noted that the dissipation of the top pocket has a larger effect on the MSE than the generation of a new spot of negative radial velocity, meaning that the MSE decreased.

Figure 6.13 is showing the behaviour of the prediction of the fields at time instant $t = 313ms$ (time snap with minimum MSE). The most important difference compared to the previous time snap is that the new pocket of high error has now increased drastically for the axial velocity seen in Figure 6.13d, whereas the rest of the field has relatively low error. This new error pocket, introduced in the previous paragraph, is caused by the newly occurring local minimum velocity at radius and height 0.00 and 2.62 times the pool diameter ($r = 0cm$, $z = 5cm$) not being captured by the prediction. The other extremes causing high errors have ascended and dissipated and their related errors have vanished, causing



(a) Accuracy of prediction of field of ρ

(b) Accuracy of prediction of field of p

(c) Accuracy of prediction of field of T

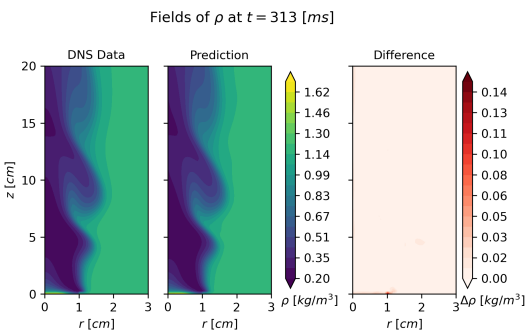
(d) Accuracy of prediction of field of u

(e) Accuracy of prediction of field of v

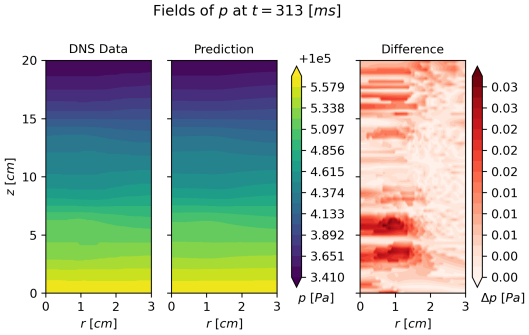
Figure 6.12: Performance of PINN in predicting flow fields in case of an unsteady flame at $t = 292ms$

a local minimum in MSE as shown in Figure 6.16b. The dissipating extreme values have a stronger effect on the MSE than the newly generated extreme value, meaning the MSE decreases. Note that the dissipation and emerging of the low velocity region are caused by puffing, the quasi-periodic behaviour of the flame explained in chapter 2. Interestingly, the time snap with lowest MSE does not have lowest maximum error. However, the zone with maximum error is smaller, meaning it still causes a lower MSE. Moving to the radial velocity predictions shown in Figure 6.13e, most important features seem to be captured slightly better than in previous time snaps. Furthermore, the most prominent difference compared to the previous results is that the plume of high positive radial velocity has propagated and dissipated further, decreasing the associated error to that region, as the maximum and strong gradients decreased: the features in this region are now better captured by the predictions. There is only one major region of high error, being at the bottom of the flame where the highly negative radial velocity has increased in time, having stronger gradients and causing a higher error in this region. However, as the other source of error diminished, the MSE almost halved compared to the MSE at the first time snap.

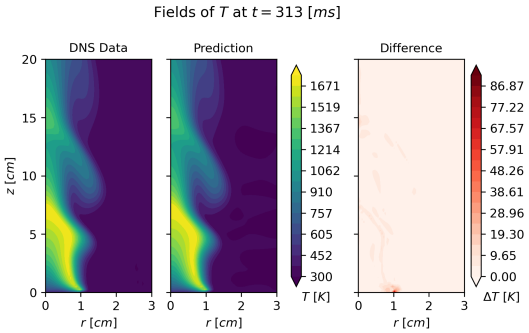
Similar to in case of steady flow, line plots of the velocities can be made to show that the PINN struggles to capture the peaks to its full extent. In Figure 6.14 the velocities are shown over height h at radius 0.42 times the pool diameter ($r = 0.8cm$) and time $t = 313ms$. It is clear from Figure 6.14a that the minimum and maximum between radius and height 1.57 and 3.66 times the pool diameter ($z = 3cm$



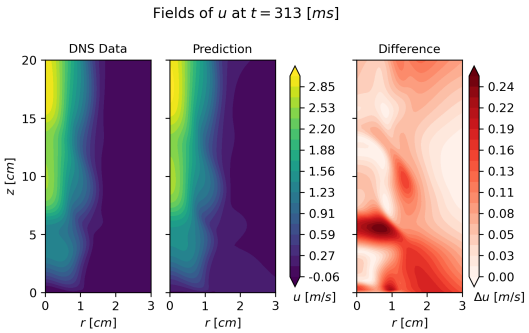
(a) Accuracy of prediction of field of ρ



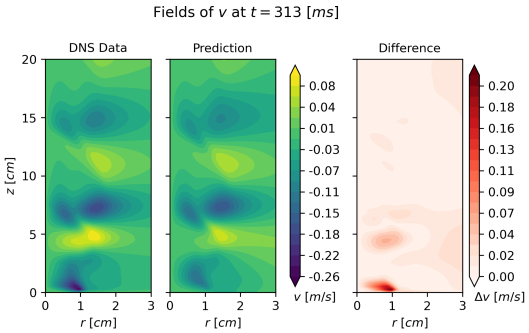
(b) Accuracy of prediction of field of p



(c) Accuracy of prediction of field of T



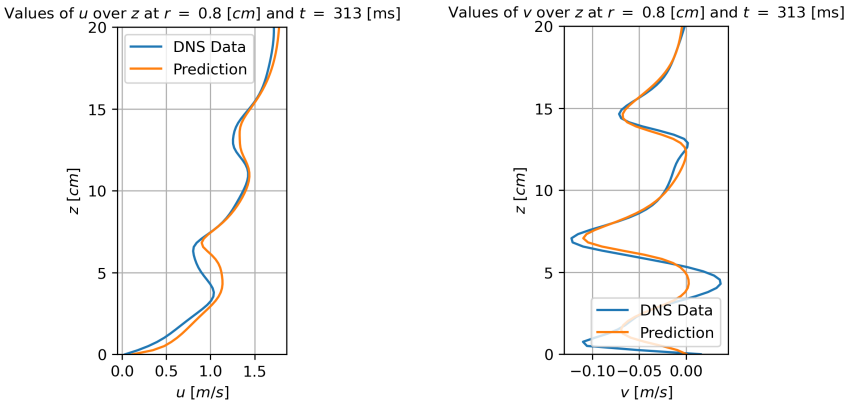
(d) Accuracy of prediction of field of u



(e) Accuracy of prediction of field of v

Figure 6.13: Performance of PINN in predicting flow fields in case of an unsteady flame at $t = 313ms$

and $z = 7cm$) are worst captured, where the z-coordinate of maximum and minimum axial velocity are not correctly predicted, potentially due to unsteady effects not being fully captured correctly. For the radial velocity, it can be seen in Figure 6.14b that the z-coordinate of maxima and minima are generally correctly predicted, albeit that the value of the peak is not always accurate, especially near the bottom of the domain. At the other time snaps, similar line plots are obtained. The main difference is that the extreme values are higher (in absolute sense) for the radial velocity at other time snaps, and captured with higher inaccuracy by the prediction. One can plot the residuals over the domain as well, as has been done for the steady case as well, leading to Figure 6.15 for time snap $313ms$. The high residuals matching with the high error in radial velocity near the pool are still there, similar to in case of steady flow as has been seen in Figure 6.4. Another region with high residuals is found near radius and height 0.00 and 2.62 times the pool diameter ($r = 0cm$, $z = 5cm$), which is close to the region of high error in axial velocity seen in Figure 6.13d. Surprisingly, there are more regions at different heights z near a radius of 0.00 times the pool diameter ($r = 0cm$) where the residuals are high, which can not directly be coupled to high errors in the predicted velocity fields or other predicted flow states. The reason for this is unknown, but might have to do with the fact that the automatic differentiation used has difficulties finding gradients near the boundary of the domain.



(a) Axial velocity

(b) Radial velocity

Figure 6.14: Predicted velocity distributions over height z at radius 0.42 times the pool diameter ($r = 0.8cm$) at $t = 313\ ms$ of unsteady flame compared to DNS data

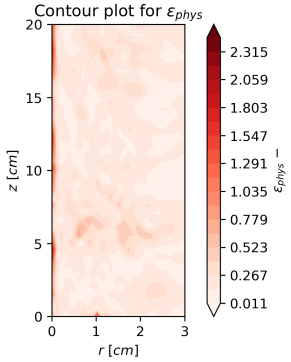
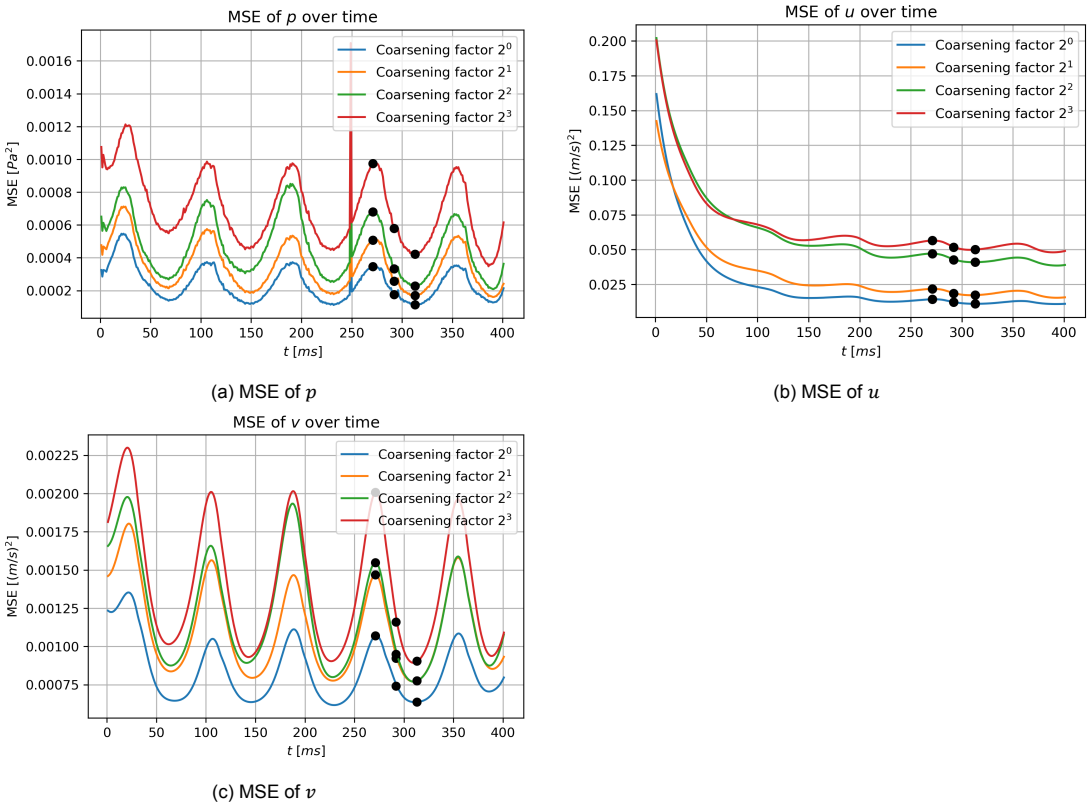


Figure 6.15: Combined residuals of the mass and momentum equations over the domain in case of the standard case in unsteady flow at $t = 313ms$

6.2.2. Effect of Coarsening

Similarly to the coarsening for the steady case, the coarsening for the unsteady case is performed by removing part of the data spatially, where the coordinate spacing is increased by a factor in the entire domain. The temporal resolution is not touched as this is out of the scope of this research and as a temporal resolution of $1ms$ already seems reasonably low. It can be seen in Figure 6.16 that coarsening the data increases the MSE, where the most prominent difference is to be seen between values of 2^1 and 2^2. It is not surprising that the MSE increases when decreasing the amount of points: local sharp peaks are less noticeable from the (coarsened) data. The reason why there is such a large difference between 2^1 and 2^2 might have to do with a large feature in the data not being visible anymore for coarsening factor 2^2 while it still was for coarsening factor 2^1. It is interesting to see that the MSE increases less by coarsening at the times of minimum MSE compared to times at maximum MSE. Predicted fields are plotted for the highest coarsening for which main features are still to be recognized in the predicted velocity fields, which is in case of coarsening factor 2^3. This means the amount of points decreased by a factor of 64, and the lowest spacing between the points found in the domain is approximately $1mm$.

In Figure 6.17, the predictions of the PINN when providing data spatially coarsened by a factor of 2^3 is shown for $t = 313ms$. It is chosen to only show it at this time snap as showing the predictions at the other time snaps does not give more insight in what happens when coarsened data is used, rather it shows the elements discussed in subsection 6.1.1, namely why the MSE fluctuates and how this relates to the dissipation and generation of vortices. It is clear that coarsening the data generally seems to deteriorate the predictions, which can be most prominently seen by looking at the absolute error of the axial velocity in Figure 6.17d. The velocity prediction in the far field has increased up to the point that the maximum error made is around radius and height 1.57 and 10.47 times the pool diameter



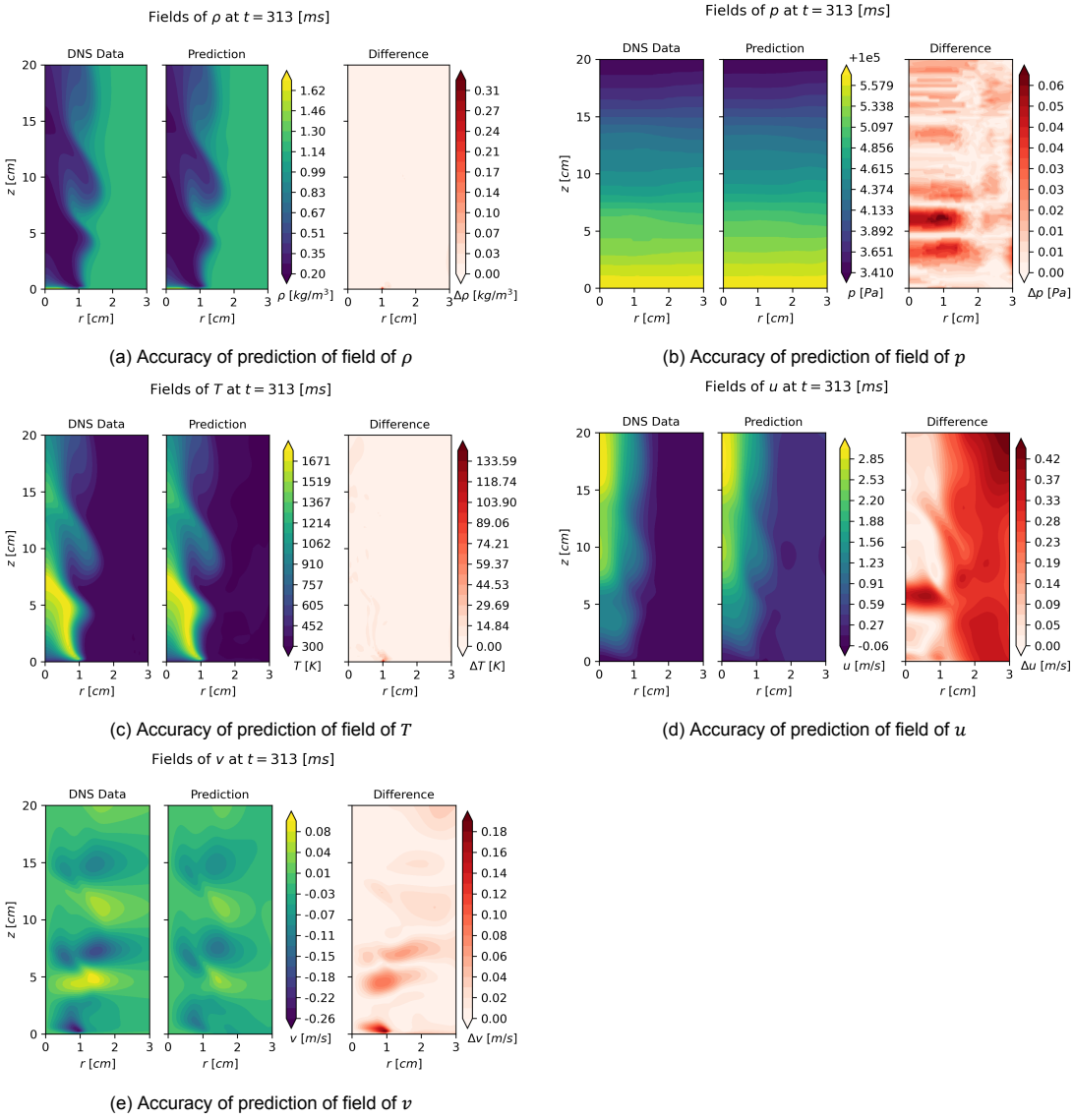
(a) MSE of p

(b) MSE of u

(c) MSE of v

Figure 6.16: MSE of pressure, axial velocity and radial velocity over time with different levels of coarsening

($r = 3cm$, $z = 20cm$), which is quite surprising. A reason for this could be that the coarsening made the points in the far field too sparse, as they were already quite sparse in the data coming from the DNS. This would also cause the effect that the far field error is weighted lower as it contains few points. Interestingly, coarsening the data does not have such a strong deteriorating effect on the prediction of the radial velocity field seen in Figure 6.17e as it did for the axial velocity, even though it can be seen that the features are captured inferiorly and the error field has more regions of slightly increased error in case of high coarsening compared to when no coarsening is applied.



(a) Accuracy of prediction of field of ρ

(b) Accuracy of prediction of field of p

(c) Accuracy of prediction of field of T

(d) Accuracy of prediction of field of u

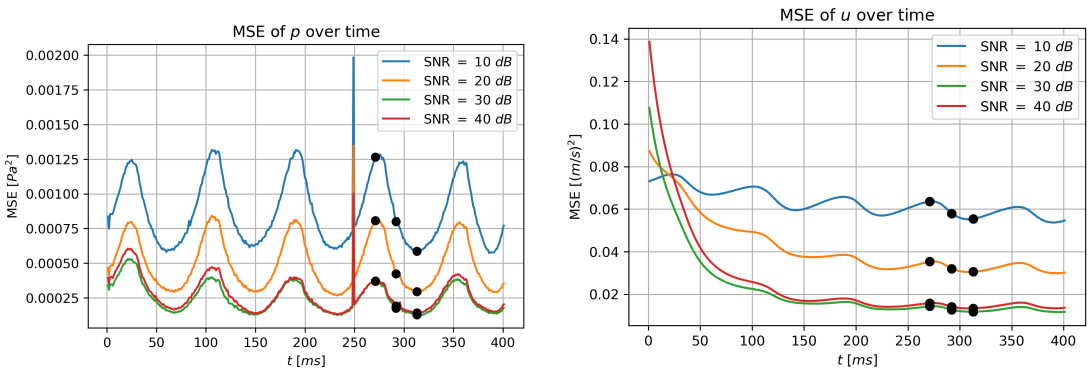
(e) Accuracy of prediction of field of v

Figure 6.17: Performance of PINN in predicting flow fields in case of an unsteady flame at $t = 313ms$ with coarsening factor 2^3

6.2.3. Effect of Noise

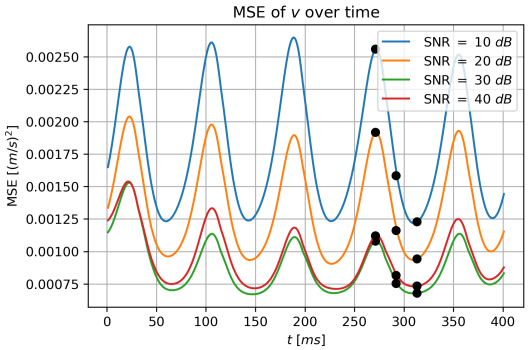
In Figure 6.18, the effect on the MSE of the predictions of adding noise with a certain SNR is shown over time. Firstly, it is seen that SNRs of 30 and 40 hardly change the MSEs compared to when no noise is added. Secondly, it is seen that way higher noise levels are acceptable than in the steady case. This might be explained by the fact that the noise is completely white, meaning that it is quasi-random in space and time. This means there is more information for the same coordinate in space available, namely all 400 time snaps, all containing a different error, on average being close to zero, as there is no bias. For the steady case, there was also no bias but just one time snap was available (as the flow is steady), meaning that the PINN had to deal with very wrong data for certain regions with high error in provided data. The results with SNR $= 10$ for the unsteady case are deemed still acceptable as the main features are still visible, thus it is decided to show the predictions for that case.

Looking at Figure 6.19, the predicted fields are seen for the case where data with a SNR of 10 is provided at $t = 313ms$. Again, it is decided to only show the results at this time snap as showing the predictions at the other time snaps does not contribute to understanding the effect of noise on the predictions. Interestingly, even though the very high source of noise, the fields for the provided data still match very well with the actual fields without noise, as seen in Figure 6.19a, 6.19b and 6.19c. This is thought to be mainly caused by the effect of time discussed in previous paragraph, where the bias



(a) MSE of p

(b) MSE of u

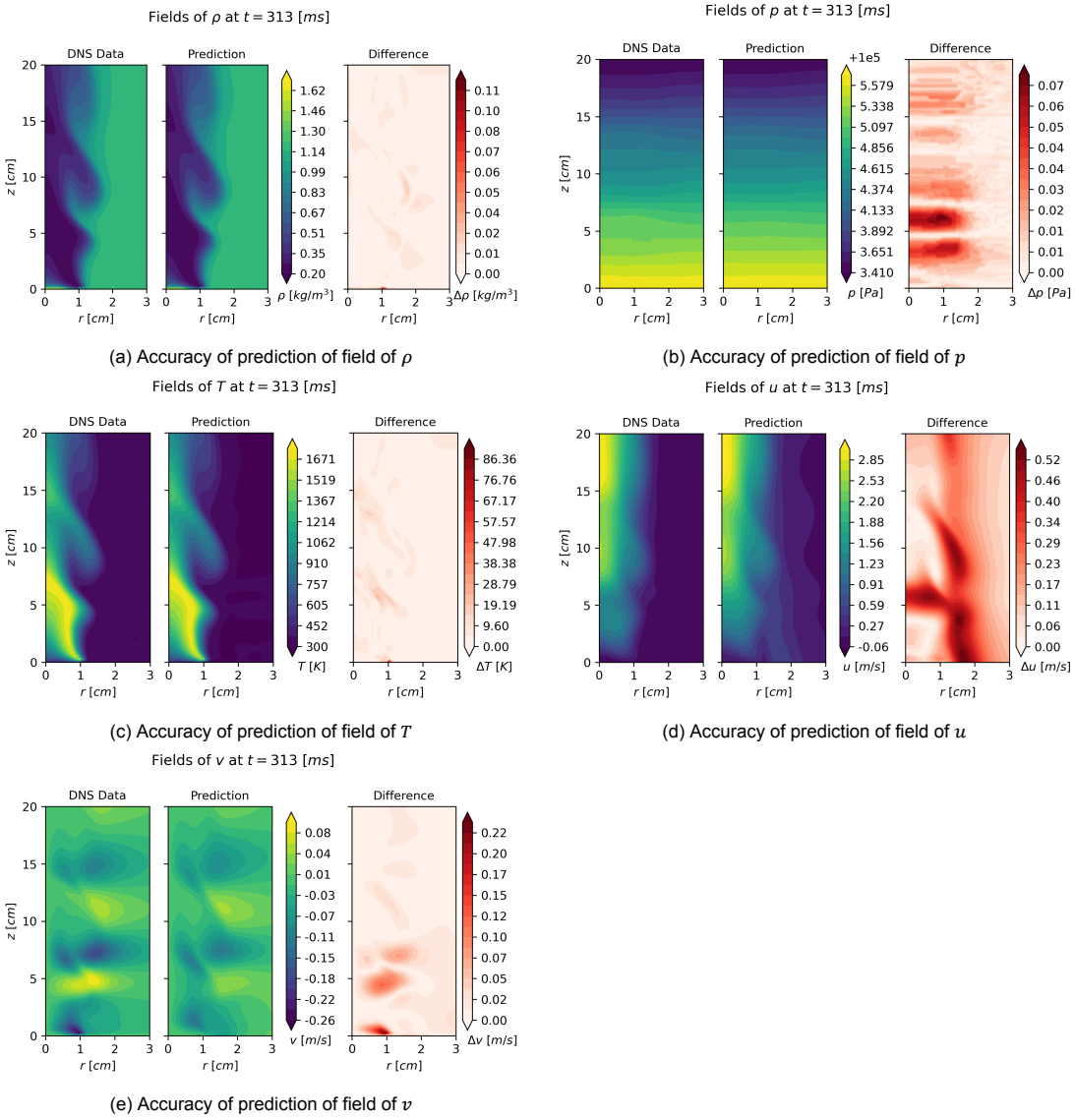


(c) MSE of v

Figure 6.18: MSE of pressure, axial velocity and radial velocity over time with different levels of noise

is zero over time, meaning that the PINN is able to smooth the predictions out well. Furthermore, the maximum error in the fields of the provided data did not come from the PINN not being able to match the provided data, but from the fact it aims to both match the provided data as well as minimizing for the residuals of the NS equations. This is known as the PINN predicts the provided data better during the phase where it only trains on the measurement. It should be noted that after training only on the measurements, the predictions were worse in case noisy data was provided compared to when data without noise was provided, logically.

Looking at the axial velocity field in Figure 6.19d, the first peak feature disappears from the prediction, and the error is generally much higher everywhere compared to when no noise is added. This can be reasoned by the fact that high levels of noise change the optimization domain heavily and can lead to a (wrong) bias in prediction of the thermodynamic states. This influences the NS equations and leads to biased computations of residuals and thus the loss function, causing higher errors in velocity predictions. Moving to the radial velocity predictions seen in Figure 6.19e, the added noise generally leads to the features being captured inferiorly. However, the difference for the radial velocity is way less prominent than for the axial velocity field. For the purpose of understanding how much noise SNR $= 10$ actually is, one can look at the noisy data fields and its difference with the actual DNS data fields in Figure 6.20. Interestingly, the PINN smooths out the predictions for these fields, as the predicted density, pressure and temperature fields look more similar to the fields from the DNS data than to the noisy fields.



(a) Accuracy of prediction of field of ρ

(b) Accuracy of prediction of field of p

(c) Accuracy of prediction of field of T

(d) Accuracy of prediction of field of u

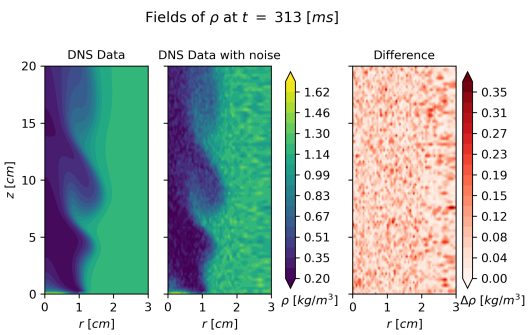
(e) Accuracy of prediction of field of v

Figure 6.19: Performance of PINN in predicting flow fields in case of an unsteady flame at $t = 313ms$ with SNR = 10

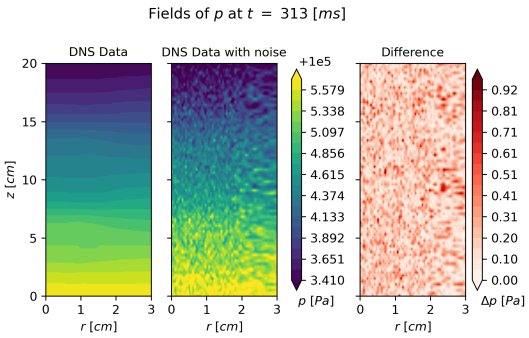
6.2.4. Effect of Reducing Provided Flow States

In Figure 6.21, the effect of removing flow states on the MSE over time is plotted. It is important to note that the same steps were taken as in subsection 6.1.4, where the density and pressure were assumed from the ideal gas law and hydrostatic equation respectively, in case they were not provided. This has been decided as removing one state from the provided ρ, p, T already gives poor results in which none of the features can be recognized like in the steady case (subsection 6.1.4), meaning that any assumption on the non-measured quantity could only potentially improve the results. The reason why the predictions are poor in the case of only two or less out of three from the density, pressure and temperature are provided might have to do with the uniqueness, as discussed shortly in subsection 6.1.4 as well.

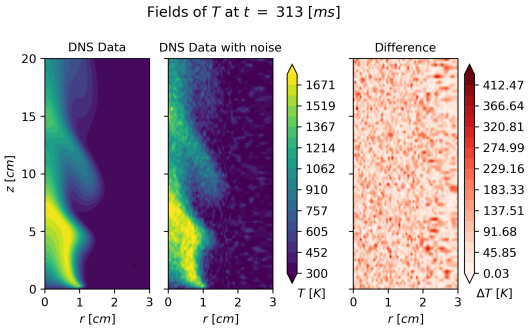
Moving on to the MSE graphs, it is also interesting to plot the MSE of the density for this case, as it is assumed using the ideal gas law in two out of the three cases shown. The MSE of the density is thus shown in Figure 6.21a, clearly showing being lowest when using the actual provided density, followed up by the cases were either just the density is assumed from the ideal gas law, or where both the density and pressure are assumed from the ideal gas law and hydrostatic equation. Interestingly, the MSE related to the last two is still very low, meaning the assumption of the density is quite accurate overall. Moving to the MSE of the pressure shown in Figure 6.21b, it is seen that the case where all



(a) Noisy field of ρ



(b) Noisy field of p



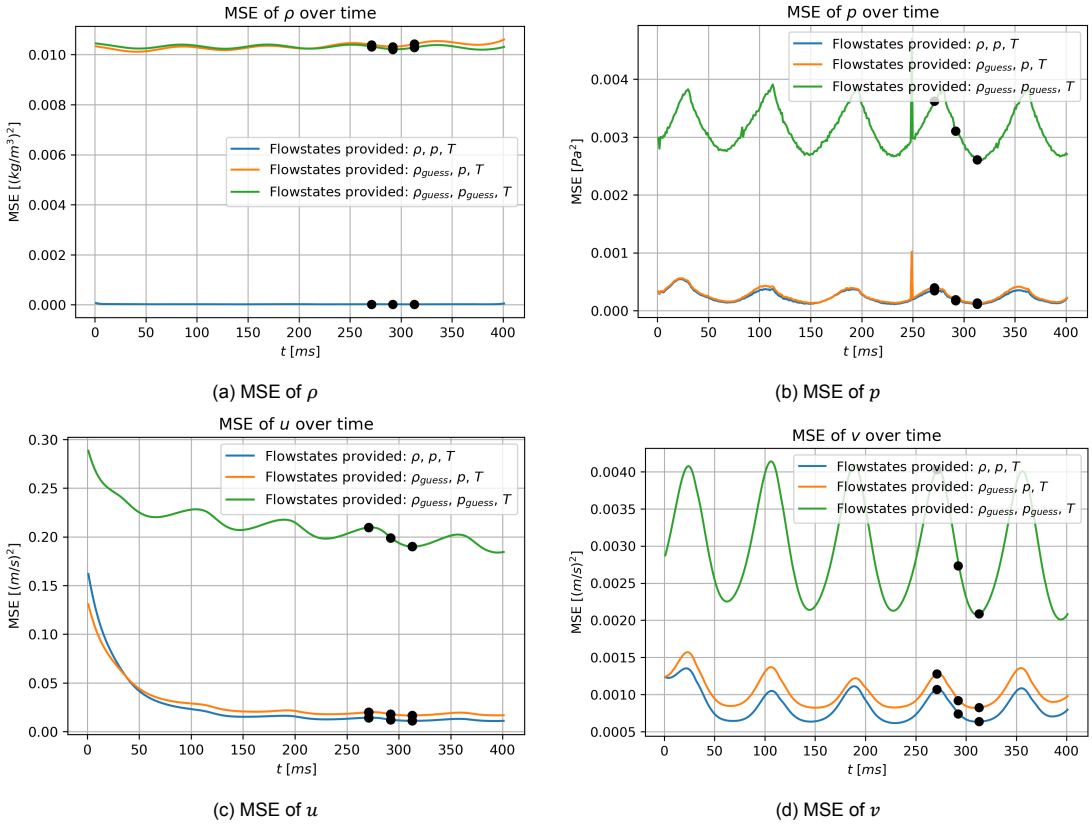
(c) Noisy field of T

Figure 6.20: Noisy fields provided to the PINN at $t = 313ms$ in case of SNR = 10

states are provided and the case where the density is assumed from the ideal gas law almost coincide. This is as in both these cases, the pressure is matched to the provided DNS data. Lastly, the case where the pressure is assumed from the hydrostatic equation is well above these, even though the MSE is still quite low, meaning the prediction is quite accurate overall.

For both the axial and radial velocity MSE plots shown in Figure 6.21c and 6.21d, it is clear that the case where both the density and pressure are assumed from the ideal gas law and hydrostatic equation leads to highly inaccurate results for both velocity components. Apparently, the degraded prediction in pressure field leads to high errors in the velocity fields: the slight oscillations seen in the pressure field in Figure 5.4b and 5.5 are so connected to the vortices and their effect on the velocities, that removing these oscillations by assuming a linear pressure deteriorates the predictions of the velocities heavily. This is the reason why this assumption still worked well for the steady case, and not in case of the unsteady flow: in the steady flow there are hardly any fluctuations on top of the linear pressure. The deteriorating effect is larger than in case of the highest artificially added noise and the strongest coarsening of the provided data discussed in previous subsections. Even though assuming both the density and pressure from the ideal gas law and hydrostatic equation was not successful, assuming only the density from the ideal gas law still works very well, as it only slightly increases the MSE for both the axial and radial velocity. To show the effects of this method, the predicted fields are discussed subsequently. Even though the case of assuming pressure on top of it gives poor results, this case is also shown as it gives insight in what is happening that causes the error to increase.

In Figure 6.22 the predictions in case the density is assumed from the ideal gas law are shown at time snap $t = 313ms$. Again, it is decided to only show the results at this time snap as showing the predictions at the other time snaps does not contribute to the understanding of the effect of assuming the density from the ideal gas law. It is seen in Figure 6.22a that the density is predicted very well overall, accept at the origin (bottom of the flame). Note that this is what was expected from the ideal gas law assumption: the constant specific gas constant taken from air is very wrong at the flame, due to the composition being extremely different than the one of air, resulting in a wrongly predicted density. Looking at Figure 6.22d, one can see that also the main features of the axial velocity field are well captured. Furthermore, the prediction and associated error look very similar to the prediction without removing the density information (Figure 6.13d), where the major difference is the very high



(a) MSE of ρ

(b) MSE of p

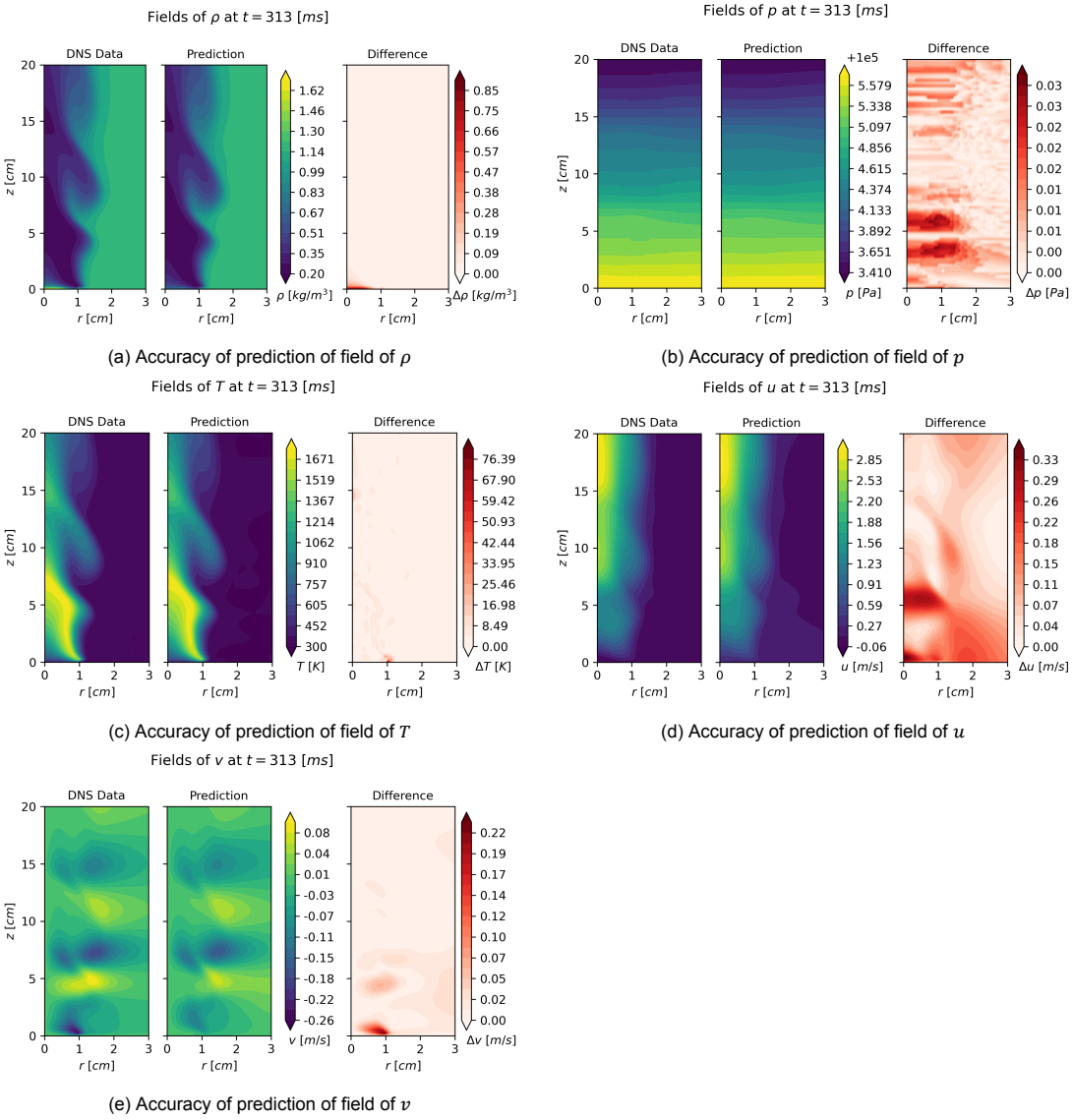
(c) MSE of u

(d) MSE of v

Figure 6.21: MSE of density, pressure, axial velocity and radial velocity over time with different provided flow states

error near the origin. This error is associated with the high error in density caused by the wrongly assumed specific gas constant at the flame. The prediction and error of the radial velocity seen in Figure 6.22e look almost identical to the one from Figure 6.13e, the only difference being that the error is slightly higher when the ideal gas law was used to find the removed density.

In Figure 6.23, the predictions and related error fields are seen at time snap $t = 313ms$ in case both the density and pressure are assumed from the ideal gas law and hydrostatic equation. The density field looks very similar as for the case when only the density has been removed and assumed from the ideal gas law: the density is accurate everywhere except close to the origin, due to the wrongly assumed specific gas constant (Figure 6.23a). The pressure field prediction is relatively accurate as well, surprisingly. Apparently, the flow does not change the pressure field much from the case of pure linear pressure caused by the hydrostatic law. There are small oscillations in pressure in the exact data, which are not there in the prediction, as seen in the error. It is seen that the alternating error pattern has become higher, as seen in Figure 6.23b. The prediction of the axial velocity seen in Figure 6.23d captures most important features, but the smaller features and details are lost, such as for example the local minimum in velocity at radius and height 0.00 and 2.62 times the pool diameter ($r = 0cm$, $z = 5cm$). Furthermore, the far field velocity is wrongly predicted, causing the highest error. For the radial velocity, all features seem to be lost, as can be seen in Figure 6.23e.



(a) Accuracy of prediction of field of ρ

(b) Accuracy of prediction of field of p

(c) Accuracy of prediction of field of T

(d) Accuracy of prediction of field of u

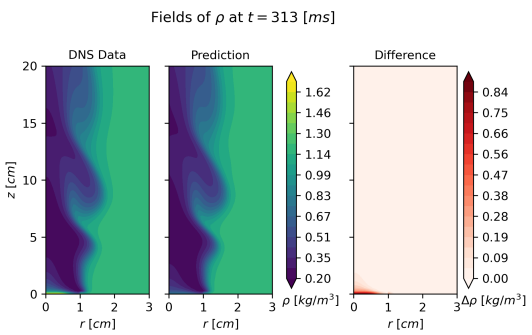
(e) Accuracy of prediction of field of v

Figure 6.22: Performance of PINN in predicting flow fields in case of an unsteady flame at $t = 313ms$ with rhoguess

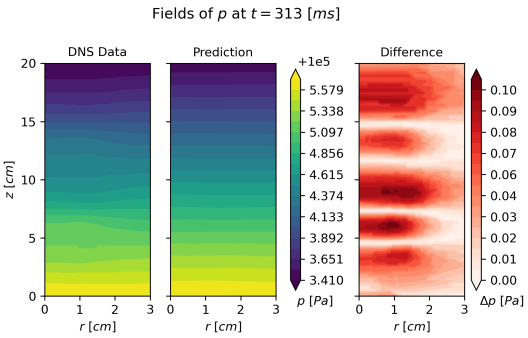
6.2.5. Predictions on equally spaced grid

It has been seen for the steady case that it was possible to provide equally spaced data instead of the data coming from the DNS results directly. The same steps are taken in the unsteady case. The main drawback here is that if you take the lowest spacing everywhere in the domain, you get an extremely large data set, meaning the decision has been made to coarsen directly to the spacing with $1mm$, and to see if the predictions are similar to when the unequally spaced data is used. Also, it has been decided to double the batch size, as it is important that more points per iteration are used for the optimizer to get a more accurate value of the loss at every iteration step, because the amount of points to train on has increased drastically as well.

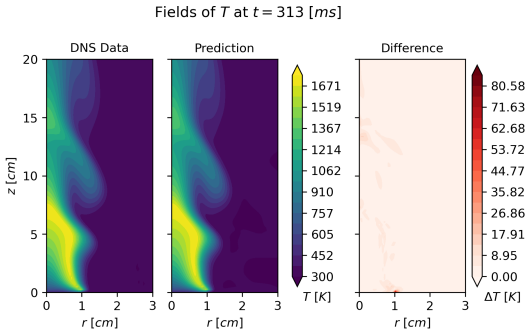
One can see the MSE over time in Figure 6.24, where it is seen that the MSE of the radial velocity (Figure 6.24b) has strongly increased and the MSE of the axial velocity (Figure 6.24a) has just slightly increased compared to Figure 6.16 with factor 2^3, interestingly. The reason for this difference is not fully clear. It might have to do with the added weighing function (exponential, like in the steady case) not being perfect and thus adding an error to the results. However, this can not be the full explanation as the exponential function seemed to work in the steady case (subsection 6.1.5). It might also have to do with the fact that the amount of points has increased drastically, even when coarsening is used and the batch size has been increased. This means that the batch of points has such a strong bias in every



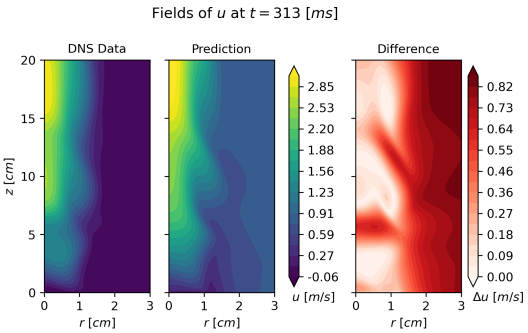
(a) Accuracy of prediction of field of ρ



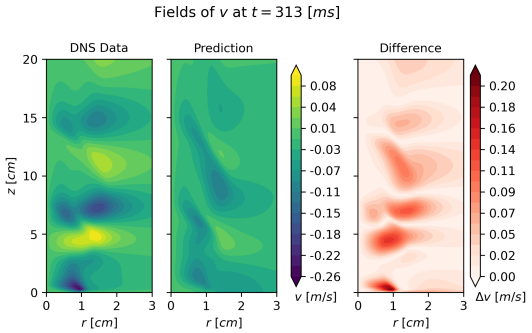
(b) Accuracy of prediction of field of p



(c) Accuracy of prediction of field of T



(d) Accuracy of prediction of field of u

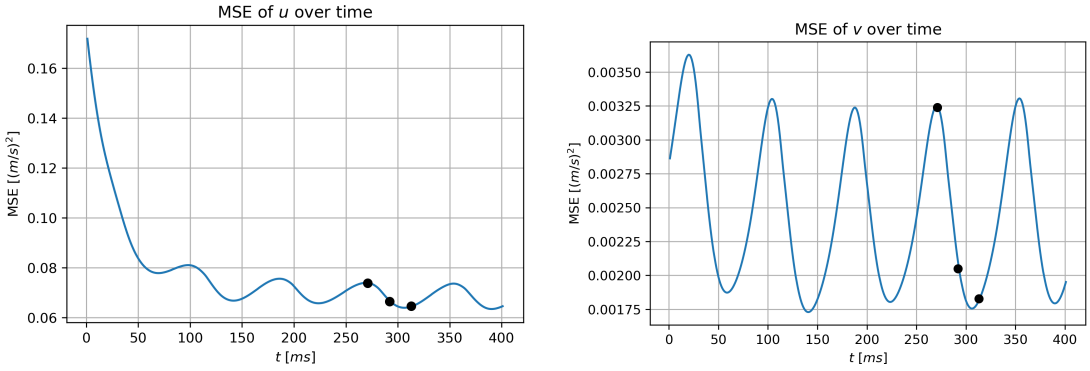


(e) Accuracy of prediction of field of v

Figure 6.23: Performance of PINN in predicting flow fields in case of an unsteady flame at $t = 313ms$ with rhoguess-pguess

iteration, the optimizer updates the weights and biases in such a biased manner, that the network does not improve anymore at some point. This theory is backed by the fact that when the 'standard' batch size (the one used in all previous cases) is used here, the results are much worse. Unfortunately, it is impossible to increase the batch size to a higher value, as this can not be handled anymore by the optimizer/workstation (memory/computational issue).

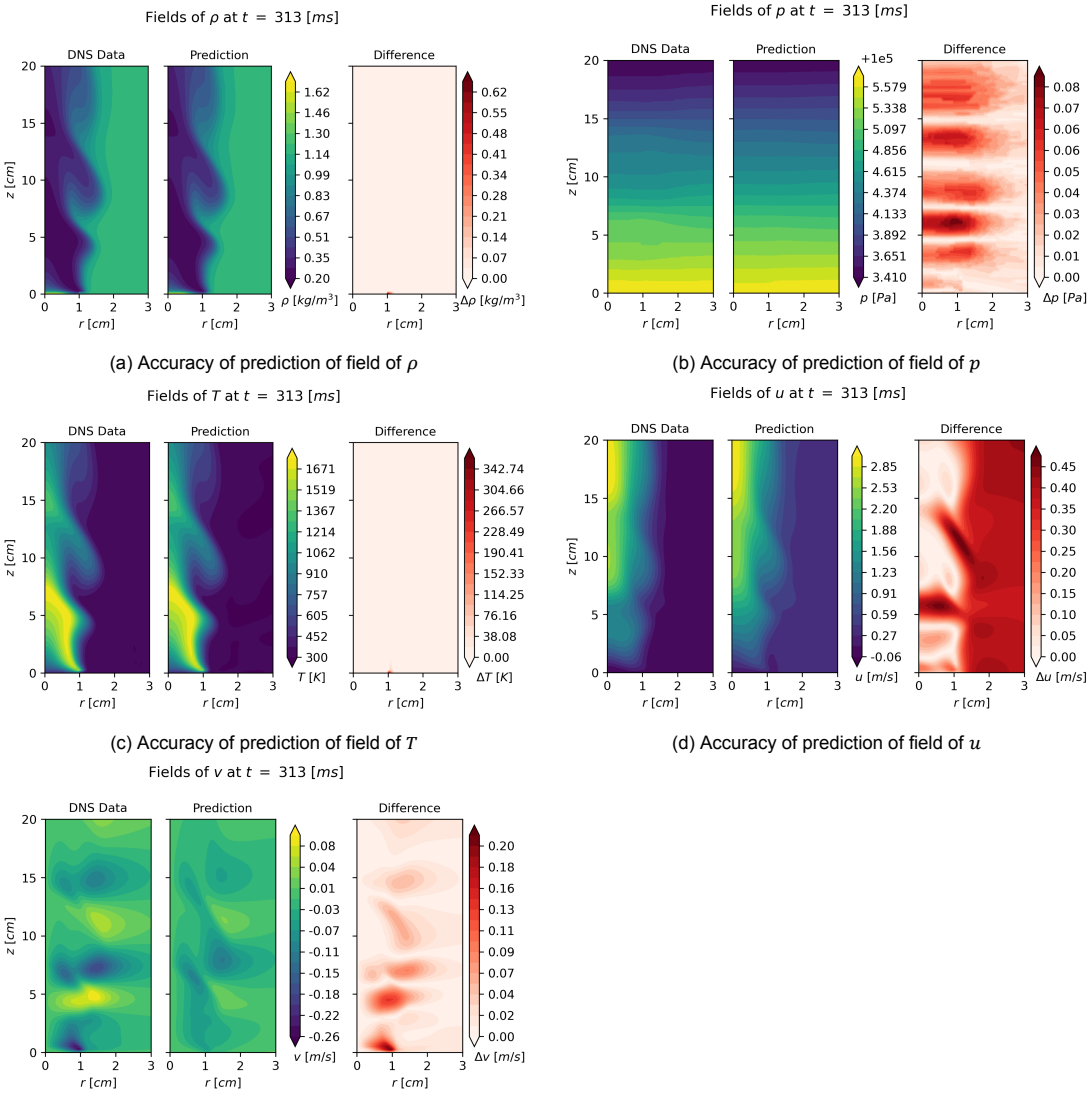
Looking at the predictions in Figure 6.25, it is seen that for the axial velocity (Figure 6.25d), it is quite similar to Figure 6.17d, with just a slightly increased error in some regions. However, when looking at the radial velocity prediction in Figure 6.25e, it is seen that most features have disappeared, and that the prediction using the DNS data still captures the main features better (Figure 6.17e).



(a) MSE of u

(b) MSE of v

Figure 6.24: MSE of pressure, axial velocity and radial velocity over time using equally spaced grid and coarsening factor 2^3



(a) Accuracy of prediction of field of ρ

(b) Accuracy of prediction of field of p

(c) Accuracy of prediction of field of T

(d) Accuracy of prediction of field of u

(e) Accuracy of prediction of field of v

Figure 6.25: Performance of PINN in predicting flow fields in case of an unsteady flame with equally spaced grid and coarsened data by factor 2^3

7

Conclusion

In this chapter, remarks are stated to come to a conclusion for this work. As aerodynamic experiments are valuable and it is not always possible to measure all flow states, reconstruction methods are important. This thesis focuses on HFM, which is a method based on a PINN and is able to recover flow states without any direct measurement on them by providing it with measurements on the other flow states. This is done by incorporating both the measurement error and residuals of the governing equations in the loss to be minimized. HFM is applied to a steady and unsteady two-dimensional pool fire in cylindrical coordinates to investigate the abilities of the framework in reacting flows with strong gradients. The data contains the density, pressure, temperature, axial velocity and radial velocity fields and comes from a DNS. The performance of the PINN is investigated by adjusting the provided data or the flow case and examining how it affects the accuracy of the PINN's predictions. The different adjustments to be considered are set out in the sub-questions introduced in chapter 4, aimed to be answered subsequently, after which the main research question can be answered.

Relating to the first sub-question, it is clear from chapter 6 that the HFM framework can be successfully used to retrieve the flow solution from only few provided flow states in pool fire problems. In this work, it has been shown by retrieving the velocity fields from the density, pressure and temperature fields. However, it should be noted that the predicted velocity fields do not perfectly fit the actual fields, where especially regions with high extremes and gradients are not predicted up to high accuracy. Whether the accuracy obtained is sufficient depends heavily on the application: in some cases the extremes are very important to be predicted accurately, whereas in other cases it might be most important to get a general picture of what the velocity fields look like. One could argue that for the radial velocity field, the predictions are not of sufficient accuracy to be used quantitatively (the value of extremes are usually highly erroneous), meaning it is more of a qualitative method, allowing to be able to recognize flow patterns and regions of positive and negative radial flow. The radial velocity field is hard to replicate to high accuracy for the PINN as it is almost zero everywhere, except for regions with very strong gradients. The predictions of the axial velocity field are generally of higher accuracy, mainly because the mean axial velocity is nonzero already and its fluctuating component is weaker relative to its mean velocity in unsteady flow. The best found axial velocity predictions have high accuracy, meaning that they could give quantitative information about the flow. Especially in steady flow, the predictions for the axial velocity fields are of very high quality. Considering the residuals, it is found that they indeed decrease to very low values around the domain, except in small regions where residuals of some order of magnitudes higher are found. The reason for this is that the PINN architecture most likely does not allow for a function of the required complexity, meaning that it can not reduce the residual to near-zero values everywhere. It makes a trade-off and finds the optimum which leads to relatively high residuals in a very small region, and almost zero residuals everywhere else.

Moving to the sub-question related to the reconstruction accuracy, it is found that the MSE for the axial velocity is relatively low, comparing to its mean, whereas for the radial velocity it is significantly high relative to the mean radial velocity. The most important results for the MSE are summarized in Table 7.1, where in the unsteady case the MSE is a range.

Table 7.1: Summary of MSEs in standard cases of steady and unsteady flow

Case	MSE u $[(m/s)^2]$	MSE v $[(m/s)^2]$
Steady standard case	0.0025	0.00033
Unsteady standard case	$0.0109 - 0.1618$	$0.00062 - 0.00135$

Moving to the sub-question about the spatial resolution requirements, it has been found that the accuracy decreases when reducing spatial resolution and that in this particular pool fire problem, the maximum spacing between points still leading to predictions where the main flow features could still be recognized is approximately $0.5 - 1.0mm$ (coarsening factor 2^2 to 2^3). This has been based mostly on the predictions on the points coming directly from the DNS data. Furthermore, similar results could be obtained in the case of equally spaced grids with appropriate weights to make up for the removal of the implicit strong weight at the flame. The effect of the coarsening on the MSE's can be seen in Table 7.2. Another important finding is that using an equally spaced grid instead of the data from the DNS grid has been shown to work, but a weighting function should be used as otherwise the solution converges to a constant. In the steady case, using the equally spaced grid even allowed to go to a coarsening factor of 2^3, as the far field was predicted better using this as the points there were less coarse compared to the DNS grid.

Table 7.2: Important architectural parameters and other settings

Case	MSE u $[(m/s)^2]$	MSE v $[(m/s)^2]$
Steady case coarsening factor 2^2	0.0221	0.00056
Unsteady case coarsening factor 2^3	$0.0480 - 0.2004$	$0.00089 - 0.00230$

Considering the sub-question related to noise, it was found that the PINN was more prone to noise in the steady case than in the unsteady case. This is most probably caused by the PINN being able to smoothen the noise out over the 400 different time samples for the unsteady case, whereas for the steady case it has only one time sample. For this reasoning, it is important to remember that the noise added is white noise in space and time, with an average bias of zero, meaning that on average the value of the flow state at a certain coordinate is adjusted by a value of zero. The most important results for the MSE in case of noise are summarized in Table 7.3. To answer the sub-question: the accuracy deteriorates when adding noise, with the PINN being more prone in the steady case, already showing a strong decrease in accuracy when extremely low noise levels are used (SNR of $50dB$), whereas in case of unsteady flow the accuracy increases only for extremely high levels of noise (SNR of $10dB$).

Table 7.3: Summary of MSEs providing noisy data in steady and unsteady case

Case	MSE u $[(m/s)^2]$	MSE v $[(m/s)^2]$
Steady standard case SNR $= 50dB$	0.0143	0.00032
Unsteady standard case SNR $= 10dB$	$0.0537 - 0.0764$	$0.00122 - 0.00265$

When reducing the number of provided flow states, it is immediately clear that the accuracy decreases drastically and no flow features can be recognized anymore. As the loss function still seems to converges to close to zero values, it might be caused by the fact that the problem would not be unique anymore. In fact, uniqueness is not proved even in case the density, pressure and temperature are all provided. To be able to still get relatively accurate results, assumptions on the density and pressure based on the ideal gas law and hydrostatic equations were made to be able to direct the PINN in the right direction. For the steady case this lead to the fact that only the temperature field needs to be provided, as long as the density and pressure assumptions are used as well, whereas in unsteady case at least the temperature and pressure needed to be given to the PINN, still assuming the density assumption. It is also clear that these assumptions deteriorate the accuracy, mainly the predictions get worse close to the pool, as there the ideal gas law with constant specific gas constant (of air) is not valid. One can see the main findings summarized in Table 7.4.

Table 7.4: Summary of MSEs providing fewer flow states

Case	MSE u $[(m/s)^2]$	MSE v $[(m/s)^2]$
Steady case provided: T	0.0113	0.00047
Unsteady case provided: T, p	$0.0166 - 0.1306$	$0.00082 - 0.00157$

Lastly, the sub-question related to the question comparing the accuracy of the PINN in steady case compared to unsteady case is considered. Other than the differences mentioned above for the noise and removing provided flow states, the steady velocity field predictions are of higher accuracy than the unsteady velocity field predictions. Two possible reasons for this are that the unsteady behaviour is harder to capture and that the unsteady flow has more regions with high gradients and extreme values than the steady flow. For the axial velocity, the steady case predictions are very close to being perfect, meaning they could be used for quantitative purposes for example when the value of the coordinates and value of the maximum axial velocity need to be found. However, in the unsteady case, the predictions are worse and these questions could not be answered fully correctly, meaning the method could be better described as being qualitative for the unsteady case, as the main features are still correctly captured. For the radial velocity, both in steady and unsteady case the predictions are insufficient for actually using the predicted extremes, so it also serves qualitative purposes only.

From the answer to the sub-questions, an answer to the main research question can be concluded. The question was stated as follows:

“To what extent can the concept of Hidden Fluid Mechanics be exploited in the problem of a steady and unsteady puffing pool fire, to enable the recovery of the full flow states from few known flow states?”

From the answers to the previous sub-questions, it is clear that HFM struggles with predicting strong gradients and extreme values correctly, especially in the radial velocity fields. The MSE of the radial velocity is of the same order as the maximum absolute radial velocity. This means that it is to be used mostly for visualizing flows and understanding flow phenomena, being able to see where positive/negative velocity pockets occur. For the axial velocity, the predictions are of better quality, where the MSE is an order of magnitude lower than the maximum absolute axial velocity. Especially in the steady flow, the prediction of the axial velocity is of high quality, reaching quantitatively correct results. In general, both for the axial and radial velocity, the unsteady flow case is predicted worse than the steady case.

For HFM to be able to be used in real-life measurements, instead of data coming from a simulation, it is important that it can deal with noise and that the required spatial resolution is not too high, as measurement equipment generally has noise and a maximum spatial resolution available. For the steady case, HFM seemed to be very prone to noise, where the prediction already suffered heavily from low noise levels with a SNR of $50dB$. In the unsteady flow case, the prediction only deteriorated when extremely high noise levels with a SNR of $10dB$ were used. The reason for this major difference is most probably explained by the artificially added noise being white, meaning random in space and time, but having zero bias. This means that at any coordinate, over 400 time snaps, the average noise added is very close to zero. This leads to the fact that the PINN can smoothen out the noise well over those measurements, whereas for the steady case it only had one measurement at every location, meaning that the average noise added at one location was not zero. To be able to make a fair comparison, one would need more realistic noise scenarios being more similar to noise in real-life measurements, as this would give a better representation of the behaviour in case noisy data is provided. Judging only from this work, one could compare the SNR in case of actual experiments to the maximum SNRs for which the PINN still worked well found in this work, to be able to say how realistic it is to apply HFM to a real measurement.

From the effect of coarsening, it is found that a maximum spacing of around $0.5 - 1.0mm$ is not too extreme to still be able to capture the main flow features. Similarly, here one could look at spatial resolutions from experimental equipment to be able to tell how realistic it is to apply the PINN to real experimental data.

Another aspect of the research question is ’few known flow states’, referring to the amount of flow states being provided to the PINN. It is found that it is required to provide at least the temperature and pressure in the unsteady pool fire case, whereas for the steady case only providing the temperature was sufficient. One should note here that assumptions based on the ideal gas law and hydrostatic equation needed to be implemented to aim the PINN in the right direction, as otherwise the predictions

obtained had nothing in common with the actual velocity fields anymore after removing just only one flow state from the density, pressure and temperature. This might be due to the PINN finding another possible solution satisfying the two fields provided, as it might be that the solution to the reconstruction problem would not be unique anymore in that case. It is emphasized that uniqueness is even an issue in case all three from the density, pressure and temperature are provided, as this is something that has not been proved yet. The difference between the steady and unsteady flow can be explained by the fact that the pressure field in the steady flow almost completely satisfies the hydrostatic law (constant pressure gradient), where this assumption is thus quite valid. In case of unsteady flow, fluctuations in the pressure fields occur and the hydrostatic law is not as valid (the pressure gradient is still relatively linear, but fluctuations occur), causing the predictions of the velocity fields to deteriorate extremely if this law is assumed. The impact of the amount of flow states required to be provided is that if more states need to be provided parallel in time, the feasibility of this actually being possible decreases. On the other hand, if for example only one flow state needs to be measured, the feasibility increases drastically.

Concluding, one can say that HFM can be used in both the steady and unsteady pool fire flow, but it should be noted that regions with extremes are not always predicted correctly, and that the handling of noise is still something to be investigated in further detail. The steady case seems easier to be actually implemented in case of real-life measurements, as only the temperature field was required to be measured to come to relatively accurate predictions.

8

Recommendations

When finding the answers to the research questions, even more questions of interest are identified, being answered by potential future research in the field. These potential future topics are discussed in this short chapter. Below, the main recommended questions to look at after the finishing of this thesis are summarized, after which they are explained one by one.

- What happens to the performance of the PINN if the spatial resolution of the different provided flow states is not equal?
- How do the spatial resolutions found in this thesis relate to the spatial resolution of measurement equipment available for the flow states?
- How does the spatial resolution required by the PINN depend on the problem scale and type and what is the best weighing function to be used when equally spaced data is provided?
- What is the effect of noise with bias?
- How does the amount of flow states required to find an approximate full flow solution depend on the problem type and how does this relate to the problem of uniqueness?
- What happens to the performance of the PINN when fewer flow states are provided and if the assumptions (ideal gas law, hydrostatic law) are more/less valid and what are other potential assumptions to be used in different problems and how do they affect the performance?
- Is there a way to force the PINN to predict the fields better in regions with strong velocity gradients?

Firstly, one of the research questions was related to coarsening the data in space. However, in all these cases it was assumed that the spatial resolution for all provided data was equal. In reality, however, the spatial resolution depends on the measurement equipment used, which is different for the different flow cases. It would be interesting to see how the PINN's predictions change when different spatial resolution for the provided flow states is used.

Secondly, it is not known how the spatial resolution required in this thesis relates to possible spatial resolutions from available equipment. Therefor, it would be valuable to compare the spatial resolutions required found in this thesis to the currently possible spatial resolutions from experimental research.

Furthermore, the spatial resolution required for the PINN to reach accurate results might be extremely dependent on the problem scale and type, meaning this might be a potential future research topic. A short discussion was present about using an equally spaced grid, where an exponential weighting function had been used as a proof of concept that the method can be used for equally spaced data as well. However, it is unknown what the best weighting is when an equally spaced grid is used, and if this depends strongly on the type and scale of the problem.

Another potential future research topic to consider is that of the effect of noise with bias. As the noise in this work was random in space and time, the PINN has shown to be quite good in smoothing high levels of noise out, especially in case of unsteady flow. However, in real-life scenarios, the noise

is usually not random in space and time, and can also have nonzero bias. The effect of these more realistic noise scenarios could be investigated when using a PINN to reconstruct flow states in pool fire flows, similar flows or any other flows.

Another interesting topic for future research might be that of the effect of using fewer flow states to reconstruct the full flow in different flows. It has been seen that removing one flow state from the density, pressure and temperature in this work already had a detrimental effect on the predictions, even though the loss seemed to converge to a low value, rising the question of uniqueness. Is this in general true when PINNs are applied to fluid problems, or is this somehow related to the type of problem it has been applied to here?

Furthermore, in this case the PINN could be pointed in the right direction by adding assumptions: the hydrostatic law and/or the ideal gas law were used to make an assumption on pressure and/or density. Even though this flow is reacting and thus does totally not have a constant specific gas constant, still the assumptions lead to nearly as accurate results as when all three flow states were provided. The questions remains if this is related to the problem, or if these (or other) assumptions to point the PINN in the right direction can (almost) always be used. There exist cases where these assumptions are even more valid, or are extremely invalid, where it would be interesting to see what it does to the predicted fields. In general it would be valuable to see the application of HFM to various other cases to get a better overview of its capabilities and setbacks throughout the aerodynamic spectrum of flow cases.

Lastly, it would be interesting to research if there is a way to aid the PINN to make better predictions in regions with strong velocity gradients. One of the main disadvantages found was that the extremes were not captured fully. The PINN predicted that there indeed were peaks at the right locations, but the values of these extremes were generally wrong, especially in the unsteady flow case. One could try adding higher weights to the residuals in these regions, meaning the NN prioritizes that particular region more. However, this would require the PINN to know the field upfront, which is not the case for the velocity fields to be reconstructed. A potential solution to this might be to first run the PINN like done in this work, and after that adjust the weights such that regions with high gradients in the previous prediction are prioritized. This could potentially be iterated multiple times and for example be done by making the weight dependent on the gradient averaged over the two velocity fields.

References

[1] P. Saini, C. Arndt, and A. Steinberg, "Development and evaluation of gappy-pod as a data reconstruction technique for noisy piv measurements in gas turbine combustors," *Experiments in Fluids*, vol. 57, Jul. 2016. DOI: 10.1007/s00348-016-2208-7.

[2] J. L. Callaham, K. Maeda, and S. L. Brunton, "Robust flow reconstruction from limited measurements via sparse representation," *Physical Review Fluids*, vol. 4, no. 10, Oct. 2019. DOI: 10.1103/physrevfluids.4.103907.

[3] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045.

[4] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data," *arXiv preprint arXiv:1808.04327*, 2018. DOI: 10.48550/arXiv.1808.04327.

[5] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020. DOI: 10.1126/science.aaw4741.

[6] C. Liu, L. Ding, M. Jangi, J. Ji, L. Yu, and H. Wan, "Experimental study of the effect of ullage height on flame characteristics of pool fires," *Combustion and Flame*, vol. 216, pp. 245–255, 2020, ISSN: 0010-2180. DOI: 10.1016/j.combustflame.2020.03.009.

[7] D. Moreno-Boza, W. Coenen, J. Carpio, A. Sánchez, and F. Williams, "On the critical conditions for pool-fire puffing," *Combustion and Flame*, vol. 192, pp. 426–438, Jun. 2018. DOI: 10.1016/j.combustflame.2018.02.011.

[8] *Action potentials and synapses*, https://qbi.uq.edu.au/brain-basics/brain/brain-physiology/action-potentials-and-synapses, [Online; accessed 13-July-2022], 2022.

[9] *Classification With Artificial Neural Networks*, https://azati.ai/disease-prediction-and-classification-with-neural-networks/, [Online; accessed 27-April-2022], 2022.

[10] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014. DOI: 10.48550/ARXIV.1412.6980.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90020-8.

[12] A. G. Baydin, B. A. Pearlmutter, and A. A. Radul, "Automatic differentiation in machine learning: A survey," *CoRR*, vol. abs/1502.05767, 2015. arXiv: 1502.05767.

[13] I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998. DOI: 10.1109/72.712178.

[14] V. Dwivedi, N. Parashar, and B. Srinivasan, "Distributed physics informed neural network for data-efficient solution to partial differential equations," *CoRR*, 2019. DOI: 10.48550/arXiv.1907.08967.

[15] M. Dissanayake and N. Phan-Thien, "Neural-network-based approximations for solving partial differential equations," *Communications in Numerical Methods in Engineering*, vol. 10, pp. 195–201, 1994. DOI: 10.1002/cnm.1640100303.

[16] Y. Shin, “On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes,” *Communications in Computational Physics*, vol. 28, pp. 2042–2074, Jun. 2020. DOI: 10.4208/cicp.OA-2020-0193.

[17] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, *Physics-informed neural networks (pinns) for fluid mechanics: A review*, 2021. DOI: 10.48550/ARXIV.2105.09506.

[18] T. Yang, X. Xia, and P. Zhang, “Vortex-dynamical interpretation of anti-phase and in-phase flickering of dual buoyant diffusion flames,” *Physical Review Fluids*, vol. 4, May 2019. DOI: 10.1103/PhysRevFluids.4.053202.

[19] J. Carpio, M. Sanchez-Sanz, and E. Fernández-Tarrazo, “Pinch-off in forced and non-forced, buoyant laminar jet diffusion flames,” *Combustion and Flame - COMBUST FLAME*, vol. 159, Jan. 2012. DOI: 10.1016/j.combustflame.2011.06.008.

[20] D. W. Carter, F. De Voogt, R. Soares, and B. Ganapathisubramani, “Data-driven sparse reconstruction of flow over a stalled aerofoil using experimental data,” *Data-Centric Engineering*, e5, 2021. DOI: 10.1017/dce.2021.5.

[21] N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz, “Shallow neural networks for fluid flow reconstruction with limited sensors,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, no. 2238, p. 20 200 097, Jun. 2020. DOI: 10.1098/rspa.2020.0097.

[22] K. Fukami, K. Fukagata, and K. Taira, “Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows,” *Journal of Fluid Mechanics*, vol. 909, Dec. 2020. DOI: 10.1017/jfm.2020.948.

[23] S. Barwey, M. Hassanaly, V. Raman, and A. Steinberg, “Using machine learning to construct velocity fields from oh-plif images,” *Combustion Science and Technology*, vol. 194, no. 1, pp. 93–116, 2022. DOI: 10.1080/00102202.2019.1678379.

[24] S. Cai, Z. Wang, F. Fuest, Y. J. Jeon, C. Gray, and G. E. Karniadakis, “Flow over an espresso cup: Inferring 3-d velocity and pressure fields from tomographic background oriented schlieren via physics-informed neural networks,” *Journal of Fluid Mechanics*, vol. 915, Mar. 2021, ISSN: 1469-7645. DOI: 10.1017/jfm.2021.135.

[25] H. Wang, Y. Liu, and S. Wang, “Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network,” *Physics of Fluids*, vol. 34, p. 017 116, Jan. 2022. DOI: 10.1063/5.0078143.

[26] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, “Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 365, p. 113 028, 2020, ISSN: 0045-7825. DOI: 10.1016/j.cma.2020.113028.

[27] E. Kharazmi, Z. Zhang, and G. E. Karniadakis, “Hp-vpinns: Variational physics-informed neural networks with domain decomposition,” *Computer Methods in Applied Mechanics and Engineering*, vol. 374, p. 113 547, 2021, ISSN: 0045-7825. DOI: 10.1016/j.cma.2020.113547.

[28] X. Meng, Z. Li, D. Zhang, and G. E. Karniadakis, “Ppinn: Parareal physics-informed neural network for time-dependent pdes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 370, p. 113 250, 2020, ISSN: 0045-7825. DOI: 10.1016/j.cma.2020.113250.

[29] A. Jagtap and G. Karniadakis, “Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations,” *Communications in Computational Physics*, vol. 28, pp. 2002–2041, Nov. 2020. DOI: 10.4208/cicp.OA-2020-0164.

[30] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, “Adaptive activation functions accelerate convergence in deep and physics-informed neural networks,” *Journal of Computational Physics*, vol. 404, p. 109 136, Mar. 2020, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.109136.

[31] M. Abadi *et al.*, “Tensorflow: A system for large-scale machine learning,” in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI’16, Savannah, GA, USA: USENIX Association, 2016, pp. 265–283, ISBN: 9781931971331.