The paradigm of field inversion applied to a wall bounded jet using PIV measurements as reference data

Master of Science Thesis

by R.J.A. van der Hulst

Aerodynamics

TU Delft

Faculty of Aerospace Engineering

Delft University of Technology

The paradigm of field inversion applied to a wall bounded jet using PIV measurements as reference data

by

R.J.A. van der Hulst

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday March 31, 2022 at 14:00.

Student number:	4307410	
Project duration:	February 1, 2021 – November 1, 2021	
Supervisor:	Dr. ir. A. Zubova,	Deltares
	Dr. R. P. Dwight,	TU Delft
Thesis committee:	Dr. S. J. Hulshoff,	TU Delft
	Dr. ir. E. van Kampen,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This thesis report was written as the final assignment required for completing my masters degree in Aerospace engineering at the Delft University of Technology. The thesis is on the application of the paradigm of field inversion on the underflow weir case using experimental data as reference and was carried out between 1st of February in 2021 and 31st of March in 2022 in collaboration with Deltares in Delft. The thesis will provide the details on the extensions to the paradigm next to its capabilities. The reader is expected to have a basic understanding of the maths and physics behind common CFD turbulence models. All executed work was performed and documented by R.J.A. van der Hulst (author). Dr. A. Zubova and Dr. R.P. Dwight provided in-depth direction, feedback, and corrections throughout the thesis. I want to thank Dr. A. Zubova for her patience and expertise during this graduation work. Research such as these can come with complications and more than once, present situations where goals have to be adapted. Without her expertise on the simulations in OpenFoam the thesis would not be at the level it is now. Also, I would like to thank her for pushing me to take a step back at times and see the total picture of the problems at hand, as it is all too easy to get lost in the tiny in-depth problems. Furthermore, I want to thank Dr. R.P. Dwight for his patience, guidance, and expertise throughout the thesis. A lot of the topics were fairly new to me, so without his, sometimes challenging, counter questions I would not have the understanding of the topic I have now. His expertise has been a great contribution to the completion of this thesis.

R.J.A. van der Hulst
Delft, March 2022

Abstract

To this date, simulating the dynamics of a fluid remain extremely expensive for most practical design problems. The large range of length and time scales to be resolved makes it especially computationally heavy. In engineering applications, the standard is still the RANS approach for CFD modelling. Most commonly the so-called two-equation models are used. The use of CFD in the aerospace design process is still severely limited by the inability to accurately and reliably predict turbulent flows with significant regions of separation. More accurate modelling approaches including LES are often not practical to use in engineering applications.

In the recent work of Deltares, such problems also arise. Research has been performed on water flow behind an underflow gate. The flow phenomenon that occurs closely resembles that of a wall-bounded jet. The reason such research has been performed is to better predict the turbulent behaviour of the jet downstream to predict possible damage to sediment. Seven weirs on the Meuse are planned to be renovated or replaced. Currently, bed protection is designed using physical scale models. Ultimately numerical models are to be the new standard for designing bed protection behind an underflow weir.

In their work experiments have been performed to acquire PIV data of the velocity field. This has been compared to their results of multiple CFD simulations which have been performed with different levels of fidelity. Several gate openings, changing the effective Reynolds number, have been used to compare the experimental results with the simulations. They concluded that although the velocity field solution of the simulations is good enough for engineering practices, all simulations show a mismatch in the area of the shear layer between the jet and the main flow.

In recent years more and more research has been done in the applications of machine learning. The capabilities of ML have increased rapidly and are now also used in closure modelling. Alongside this are the continuously improving experimental capabilities which allow for much higher resolution information. The combination can be used for data-driven techniques to improve upon current RANS models.

In recent work, the paradigm of field inversion has been proposed. Here instead of calibrating the modelling coefficients, a corrective field is used to effectively address the modelling deficiency. The correction field has been applied to the production term of the specific turbulent dissipation rate transport equation. To infer the values of the highest probability for the corrective field inverse methods are proposed. The method proposed uses Bayesian inversion, which includes an optimisation process. As the problem consists of a large number of variables normal optimisation processes are too expensive. Therefore, the adjoint method is proposed to compute gradients efficiently.

In this work, the goal is to extend upon the recent work on field inversion. The paradigm will be applied to the underflow weir case of Deltares. The experimental data will be used to infer a spatially varying corrective field used to correct the simulations to improve their predictive capabilities. As the density of data points is often far smaller than the density of simulation cells, and often data sets do not cover the whole simulation domain the paradigm is extended to accept imperfect data.

It was found that with these extensions a correction field could be found that can lower the cost function by a factor of two. The prediction of flow features behind the weir were predicted more accurately using the newly found corrected model.

List of Figures

Contents

I

Introduction

1

Problem Description

To this date, simulating the dynamics of a fluid remain extremely expensive for most practical design problems. The large range of length and time scales to be resolved makes it especially computationally heavy. Therefore different ways of filtering are being used to reduce the computational time of such simulations, which introduce additional assumptions limiting their accuracy. An example is the Reynolds averaged Navier Stokes approach where time-averaged variables are solved and the influences of fluctuations are modelled. While using filtered approaches some modelling has to be done to close the system. And this is where some inaccuracies are introduced.

In engineering applications, the standard is still the RANS approach for CFD modelling. Most commonly the so-called two-equation models are used. Very common RANS two-equation models are k-epsilon and k-omega models, which were introduced in 1972 [10] and 1941 [**?**] respectively. Numerous adaptations have been proposed to improve these models such as [15] and [38]. As can be seen, these models are fairly old but are still used today.

The use of CFD in the aerospace design process is severely limited by the inability to accurately and reliably predict turbulent flows with significant regions of separation [29]. More complex RANS models have been proposed such as the Reynolds Stress Transport method, where the Reynolds stresses are modelled directly. However, currently, RST models are not commonly used as they lack robustness and are occasionally less accurate [29].

More accurate modelling approaches include LES and combinations of LES and RANS. However, these models involve significantly higher computational times and are often not practical to use in engineering applications.

Managing the vast amounts of data generated by current and future large-scale simulations will continue to be problematic and will become increasingly complex due to changing HPC hardware. Therefore it seems that CFD for engineering applications has reached its plateau [29].

However, in recent years more and more research has been done in the applications of machine learning. The capabilities of ML have increased rapidly and are now also used in closure modelling. Alongside, this are the continuously improving experimental capabilities that allow for much higher resolution information.

The combination can be used for data-driven techniques to improve upon current RANS models. The models used in CFD modelling always have been data-driven in a sense. Where models are derived from theory and a set of model coefficients are used to tune the response of the model with available experimental data.

Calibration methods for model coefficients have been proposed over the last decade. Methods include least squares optimisation and Bayesian procedures to infer values for the model coefficients [17] [3][6][7][14][22]. However errors due to assumptions made in the model still remain.

Also in the recent work of Deltares [1] such problems arise. Research has been performed on water flow behind an underflow gate. The flow phenomenon that occurs closely resembles that of a wall-bounded jet. The reason such research has been performed is to better predict the turbulent behaviour of the jet downstream to predict possible damage to sediment. Seven underflow weirs on the Meuse are planned to be renovated or replaced. Currently, bed protection is designed using physical scale models. Ultimately numerical models are to be the new standard for designing bed protection behind an underflow weir.

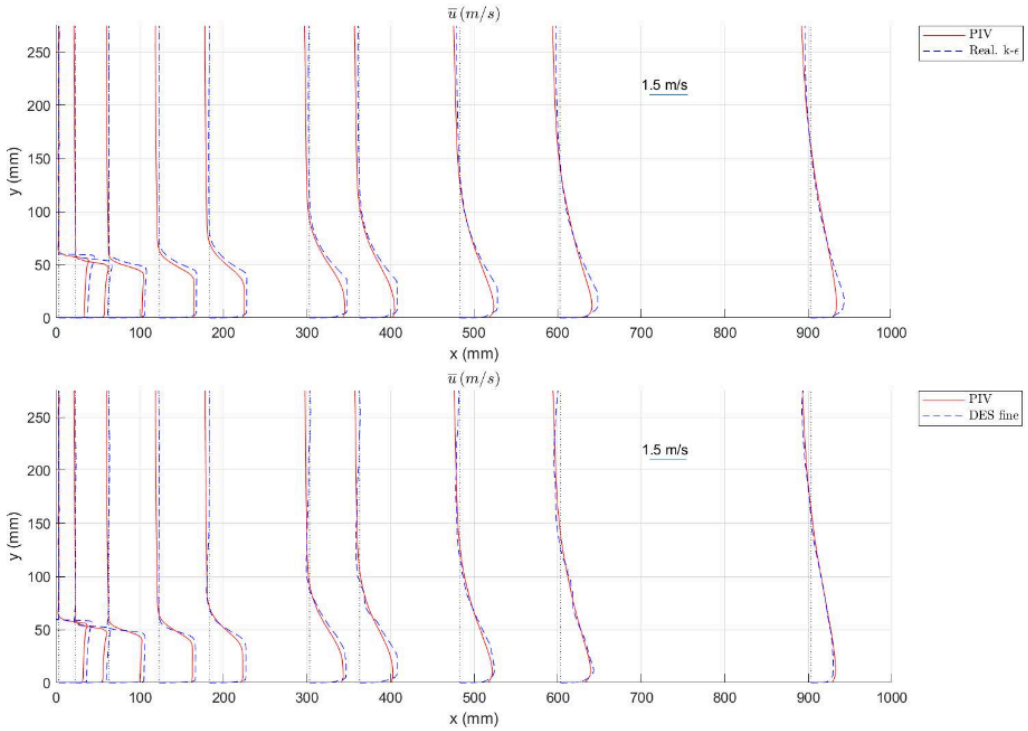


Figure 1.1: Profiles of streamwise mean velocity: PIV measurements (top) vs CFD calculations employing different turbulence models (bottom) [1].

In their work experiments have been performed to acquire PIV data of the velocity field. This has been compared to their results of multiple CFD simulations which have been performed with different levels of fidelity. Several gate openings, changing the effective Reynolds number, have been used to compare the experimental results of PIV and CFD. They concluded that although the velocity field solution of CFD is good enough for engineering practices, all CFD simulations show a mismatch in the area of the shear layer between the jet and the main flow, as can be seen in figure 1.1. Also, the correlation of turbulent kinetic energy from the CFD results is fairly poor, as can be seen in figure 1.2.

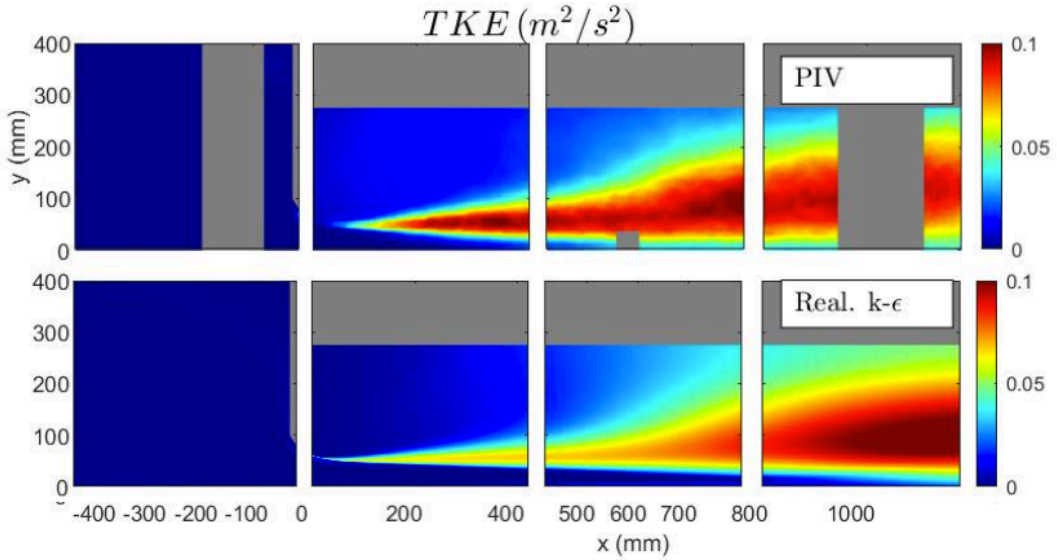


Figure 1.2: Contours of turbulent kinetic energy: PIV measurements (top) vs the realisable k-epsilon model (bottom) [1].

2

Previous Work

In the work of Parish and Duraisamy [18] the paradigm of field inversion is proposed. Here instead of calibrating the modelling coefficients, a corrective field is used to effectively address the modelling deficiency. A functional relationship for the corrective field and the solution is sought to be found using ML techniques. The ML model is ultimately used to be able to predict the corrective field for a problem outside of the data set. This paradigm thus consists of two steps, first field inversion using the Bayesian framework is used to infer corrective fields for a sufficiently large number of problems. After that ML is used to generate a predictive model based on the flow features of the solution.

A model of a physical system, such as in fluid dynamics, can be described as:

$$\mathbb{R}(\boldsymbol{Q}, \boldsymbol{M}(\boldsymbol{Q})) = 0 \tag{2.1}$$

Where the operator $\mathbb{R}$ contains the governing equation of the system, often in the form of PDE's, $\boldsymbol{Q}$ contains the model variables, and $\boldsymbol{M}$ contains the model equations.

In the work of Parish and Duraisamy [18] the model will be replaced by a stochastic system:

$$\mathbb{R}(\boldsymbol{Q}, \boldsymbol{M}(\boldsymbol{Q}, \beta(\omega))) = 0 \tag{2.2}$$

Where $\beta : \Omega \ni \omega$ is a random function resulting from the machine learning process. Realisations of β are spatial varying field variables, used for correcting the modelling equations. The spatial varying field β is a result from the functional relationship $\beta(\eta)$ where $\eta(\boldsymbol{Q})$ are the input features available from the results of the closure model, which are generally an extension of the model variables.

In order to infer the values of highest probability for the corrective field β inverse methods are proposed. The method proposed uses Bayesian inversion, which includes an optimisation process. As the problem consists of a large number of variables normal optimisation processes are too expensive. Therefore, the adjoint method is proposed to compute gradients.

In later work of van Korlaar [31] this paradigm further extended to principal flow cases such as periodic hills. Also, the continuous adjoint is proposed in order to work seamlessly with the common CFD solver OpenFoam. It was found that the corrective term could accurately be predicted for unseen higher Reynolds number cases.

In figure 2.1 the results from field inversion are shown for a periodic hills case. As can be seen, the corrective field term is almost equal to one in the bigger portion of the domain. Only near the downward slope of the hill, the corrective term becomes lower than one, meaning it is correcting the model equations.

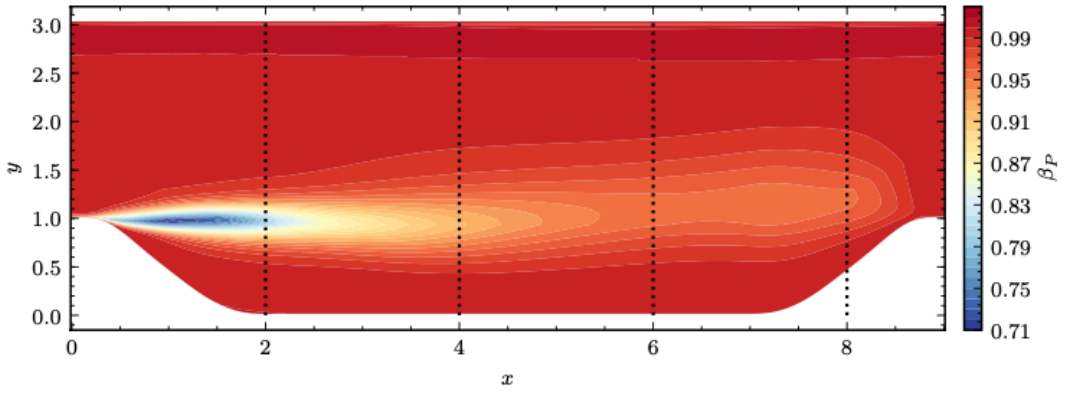


Figure 2.1: Inferred corrective function for periodic hills (Re = 5,600). [31]

3

Research Objective and Questions

For this research, it is proposed to extend upon the work done by Parish and Duraisamy [18] and van Korlaar [31]. The paradigm of field inversion is going to be applied on the wall-bounded jet, occurring behind the underflow weir, as researched by Deltares [1].

Therefore the research objective is going to be:

"***To improve the RANS closure modelling for predicting wall-bounded jet flows behind a weir extending on the paradigm of field inversion using experimental PIV data as reference data.***"

From this objective, several research questions can be formulated. The questions sought to be answered in this research are:

1. Can the paradigm of field inversion be applied on more complex problems like wall-bounded jet flows behind weirs?
   (a) Is a corrective term capable of improving a complex k-omega simulation with respect to a PIV baseline? And to what extent?
   (b) Can the continuous adjoint formulation be used on more complex and unstructured grids?
2. Can experimentally obtained data from PIV be used for the field inversion paradigm?
   (a) Is it possible to apply field inversion with courser reference data?
   (b) Is it possible to apply field inversion with limited reference data?

II

Background

4

Weirs and Turbulent Wall Jets

In this chapter, some background is given on the problem of this literature review.

In the Netherlands, water management is a key aspect of society. The name 'The Netherlands' can be roughly translated to 'the low lands'. This is true as almost one-third of the land lies below sea level. This makes the land susceptible to flooding. Almost two-thirds of the land is in danger of floods. The danger of flooding does not only come from the sea, the rivers flowing through the Netherlands form a great threat as well.

Three major European rivers end in the Netherlands, namely the Rhine, Meuse, and Scheldt. Throughout the centuries there are many stories of river floods that caused much loss of life. The first large construction on these rivers was performed by the Romans, building a dam in the Rhine. As water defences improve, the severity of floods and dike bursts also increased as more people relied on river defence.

The continuous battle against the water has resulted in incredible structures such as the Dutch Delta Works. Across the Netherlands, various structures can be found which protect against disasters in multiple ways.

But the management of water for transport is equally as important for the Dutch. Having multiple connections between the major rivers and smaller rivers makes an elaborate network of these so-called waterways. Dams, weirs, and locks are built to control the water levels of the rivers and to allow for large freight ships to move along the waterways.

4.1. What is a Weir

One structure used to control the flow of waters through rivers is a weir. A weir is a barrier across the width of a river or stream. It differs from a dam in that they control the flow rate of the river, as opposed to containing the water. This is commonly done by an overflow, where water can freely flow over the top of the weir. But an underflow weir can be used too, where an opening is present at the bottom of the weir. Weirs are commonly used to control the water levels of the river upstream, to make them more navigable for ships and for flood prevention.

During periods of high discharge, for example in spring, weirs are very effective in controlling the flow rate of the water. Sluice gates can be used to either increase or decrease the volume of water flowing downstream. For this reason, such structures, are often placed upstream of towns or cities. The change of floods is effectively mitigated in these cases.

Also, weirs can be used to manipulate the characteristics of the river. A weir can be very effective in calming a river. Ships can then more easily navigate the river as it is not affected by strong currents or turbulent regions. Locks are commonly combined with weirs on large rivers, for ships to be able to cross the water level jump.

4.2. Problem with Underflow Weirs

A proper design is thus of utmost importance as weirs are of great importance for the water infrastructure. One problem that occurs with underflow weirs is damage to the sediment behind the structure. As one can imagine, depending on the height difference, water is pushed underneath the weirs with high pressure. A

strong jet-like flow is created under and behind the weir. Protection must be applied to the river bed to prevent damage.

If proper protection is not present, over time large underwater cavities and trenches can start to exist, strengthening this effect even more. Concrete can be used to protect the riverbed directly behind the underflow weirs. However, construction of this type of bed protection can be rather expensive, so should therefore be limited.

Because of this problem, further investigation is needed in the flow structures behind underflow weirs. Designing a weir that limits damage to the riverbed can save a lot of money. But as well, being able to predict the flow behaviour better can help predict when maintenance might be needed.

4.3. Turbulent Wall Jets

Due to the pressure difference over a weir, water is pushed through a slot into a volume of stagnant or slowly moving fluid. This creates a jet-like flow structure behind the weir. High-velocity water enters a volume of water of low average velocity. The high-velocity jet stream is on the bottom bounded by the riverbed. The flume of the jet can thus only expand laterally and upward. These flow types have been studied extensively as they are present in many applications. Some examples are fuel injection in an engine, the exhaust from a rocket engine, and the jet stream in the atmosphere. An added complexity, in this case, is the free surface between the water and the air. This makes the turbulent wall-bounded jet not only dependent on the Reynolds number but on the Froude number as well. Due to the nature of the flow field, complex turbulent interactions are present with large-scale differences.

In this research, the focus is limited to the flow behind an idealised weir with a flat bottom river bed behind it, the underflow gate. Therefore flow is going to be simplified as a plane wall-bounded turbulent jet. The most key features of a turbulent wall jet will be discussed. However, the research on this was limited as the complete theory behind wall jets is not as relevant for achieving the goal of the research.

4.4. Idealised Plane Wall Bounded Jet Flows

In this section, the literature around an idealised turbulent plane wall-bounded jet is discussed. In this case, this means a slit of infinite width, a smooth wall, and no free surface.

A wall jet can be defined as a shear flow parallel to a wall where, because of the initially supplied momentum, at any station, the stream-wise velocity over some region within the shear flow exceeds that of the external flow [13]. Turbulent jets are created by a pressure drop over a small opening. In the case of a wall jet, the turbulent flow is travelling parallel to a wall. The wall causes asymmetry of the jet, as a turbulent boundary layer forms on the wall. The jet is assumed symmetric along the lateral dimension, making it a plane wall jet, as the slit is of infinite width in the idealised case. A schematic sketch of a plane turbulent wall jet is shown in figure 4.1.

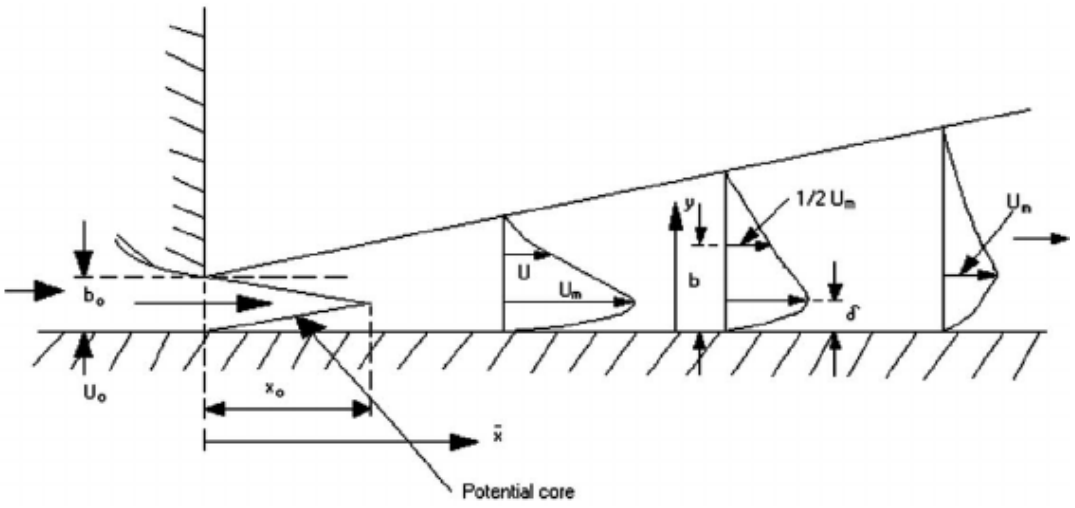


Figure 4.1: Sketch of a plane turbulent wall jet [25].

4.4.1. Flow Regimes

The flow field of a turbulent wall jet is of great interest in the field of turbulence as it presents a two-scale character. The developed jet can be divided into a lower region and an upper region. Here the lower region is similar to a turbulent boundary layer while the upper region resembles a free jet. The interaction of the scale difference of turbulence in the two regions creates a complex flow field and has a big influence on the development of the turbulent wall jet.

In figure 4.1 several length scales are used to define the regions. At the slit, b_0 denotes the height of the opening and U_0 is the uniform flow velocity through the opening, known as the core of the jet. As the core of the jet travels downstream it interacts with the wall and stagnant fluid around it. A boundary layer starts to form at the wall, growing in size the farther away it is from the opening. Also, a mixing layer starts to form as the jet shears and mixes with the surrounding fluid. This layer grows in size as well the farther away it is from the slit. At a certain distance, the mixing layer and boundary layer have grown enough to meet each other, as the momentum of the core of the jet has been transferred away. The wedge-like shape these flow structures form is called the potential core. The length between the slit and the point in which the two layers meet, the potential core length, is given by the distance x_0. In the potential core, the mean velocity is equal to U_0 [25]. Also, the length of the potential core is empirically found to be 6.1 to 6.7 times the slit opening b_0 [21].

Behind the potential core starts the fully developed region. This region shows a complex turbulent two-scale character. The lower portion resembles a turbulent boundary layer, while the upper region is similar to a free jet. The difference in turbulent length scales makes for a complex interaction, defining the geometry of the jet [8].

In this region, the transverse velocity profile of the stream-wise velocity does keep its geometrical shape. This means that, although this region grows with size, the shape of the velocity profile stays constant. The maximum velocity, denoted by U_m, is at its maximum at the end of the potential core and decreases to the zero or the free stream velocity at some distance behind the slit. As the streamwise momentum is transferred laterally the maximum velocity drops and the height of the jet increases.

The height of the maximum velocity U_m from the wall is given by δ and represents the height of the boundary layer. The height of the jet, b is defined as the distance from the wall to the point where the velocity has dropped below half of the maximum velocity U_m. Obviously, there still is some positive streamwise velocity present above the definition of the jet height, however, this is analogous to the scaling used for free jets. The plane wall jet follows a Gaussian velocity profile in the direction of the free shear layer, equal to the free jet.

As the fluid exits the slot it interacts with the surrounding stagnant fluid. Due to viscosity and turbulent mixing the stagnant flow gets drawn into the developing wall jet, which is known as entrainment [13]. The initial momentum of the jet is transferred towards the entrained fluid, causing the jet to grow. This forms a recirculating flow region above the wall jet due to the entrainment of a finite stagnant fluid [8]. This effect could possibly alter the shape and magnitude of the jet's velocity profile.

4.5. Experimental Results

Plane wall jets have been researched to quite an extent. In this section, the most important results and findings, relevant to this literature review, will be discussed.

4.5.1. Inlet and Potential Core

As flow is pushed through a slit, the flow is accelerated to the maximum velocity U_m. Because of this a boundary layer already forms in front of the slit. The flow going through the slit is thus not completely uniform. This has been clearly shown in the experimental studies performed by Eriksson [8]. In figure 4.2 the velocity and turbulence intensity are plotted over the slot height, measured using Laser-Doppler techniques. In this experiment, the laminar boundary layer height showed to be almost 40% of the total height of 3.6mm. The turbulence intensity was rather low, only 1%, however, inside the boundary layer, it doubled.

4.5.2. Developed Region

For mean velocities in the developed region, the most data is available. As discussed before, the shape of the transverse velocity profile does not change in shape when moving downstream. Data from experiments as shown by Rajaratnam[21] is shown in figure 4.3. As shown, the maximum velocity drops moving away from the slit as the height of the jet increases.

The shape of the transverse velocity profile can be made self-similar. The velocity is scaled with the maximum velocity U_m and the y-axis is scaled with the jet height b as shown in figure 4.4. Here the normalized

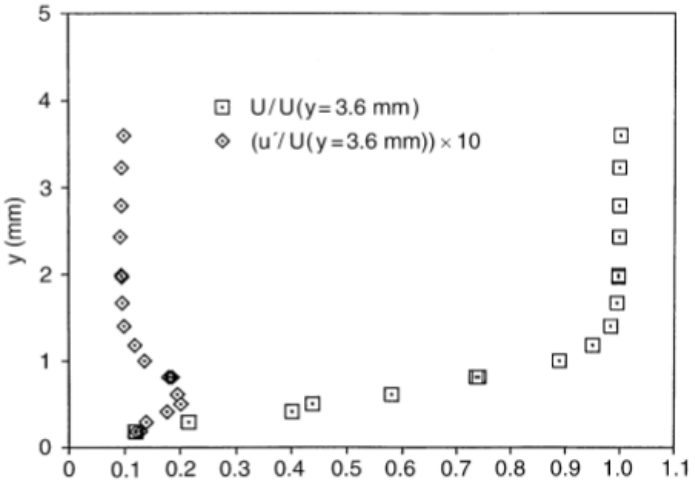


Figure 4.2: Mean velocity and turbulence intensity profiles at inlet [8]

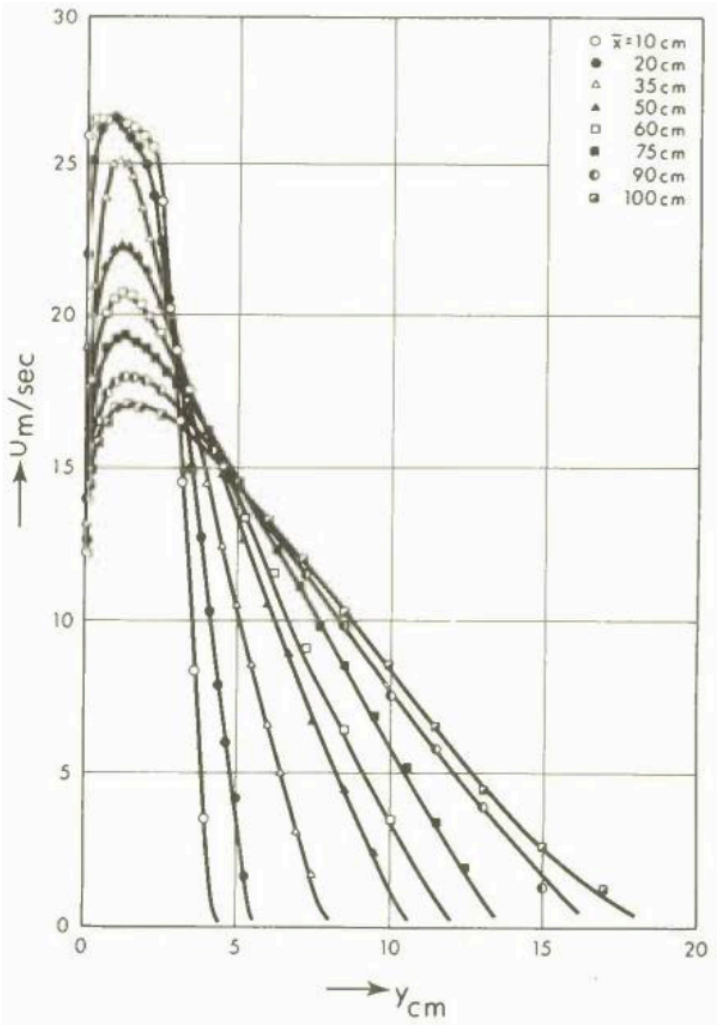


Figure 4.3: Velocity profiles in a plane wall jet [21].

height is given as η. The velocity profile can be represented by equation 4.1 [32].

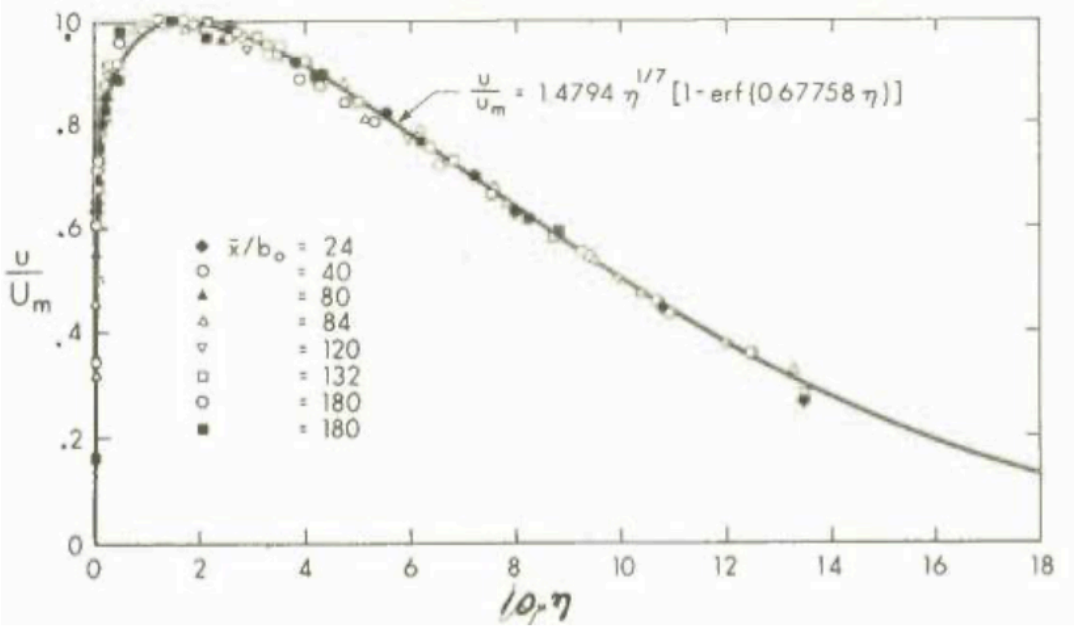


Figure 4.4: Similarity of velocity profiles in plane wall jets [32].

$$\frac{u}{U_m} = 1.48\eta^{\frac{1}{7}}\left(1 - erf(0.68\eta)\right) \tag{4.1}$$

The maximum velocity of the jet decreases when moving downstream. This is already shown in figure 4.3. In figure 4.5 it is shown from several experiments how the maximum streamwise velocity U_m is decreasing and the jets half-width is increasing when moving downstream. The half-width of the jet is shown to vary linearly with x. The U_m however seems to decrease with the square root of x. Also, close to the slit, U_m does not seem to decrease. An empirical equation can be found for the decay of U_m and is given in equation 4.2 [1]. From this equation, it can be shown that for $x < 12.5b_0$ the streamwise velocity does not decrease at all.

$$\frac{U_m}{U_0} = \frac{3.5}{\sqrt{\frac{x}{b_0}}} \tag{4.2}$$

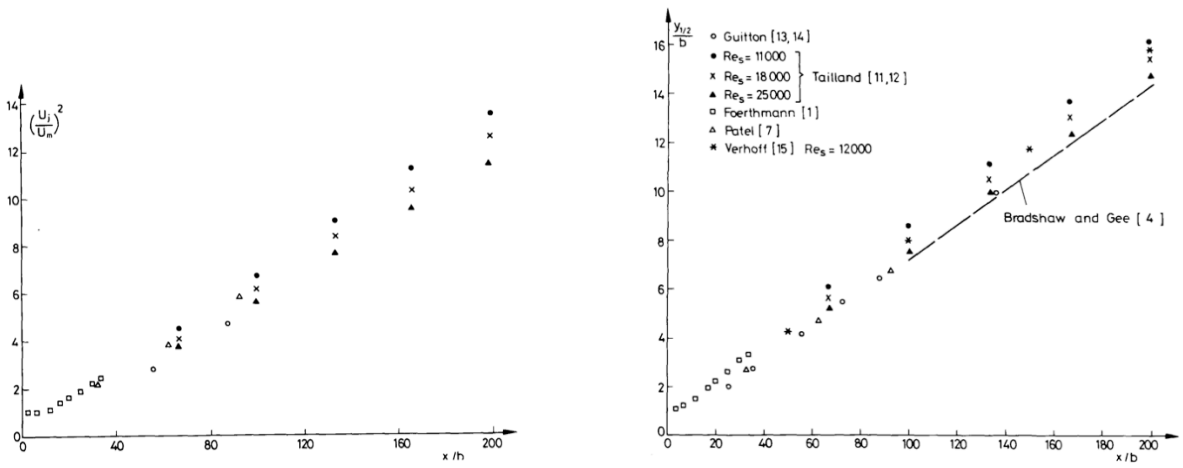


Figure 4.5: Variation in maximum streamwise velocity and half width moving downstream [13].

4.5.3. Turbulence Properties

Limited research has been done on the Reynold stress components. However some experimental data is available [13] and is presented in figure 4.6. Here the Reynold stresses are compared between different studies. It shows that the measurements are all in the same ballpark however still present some differences.

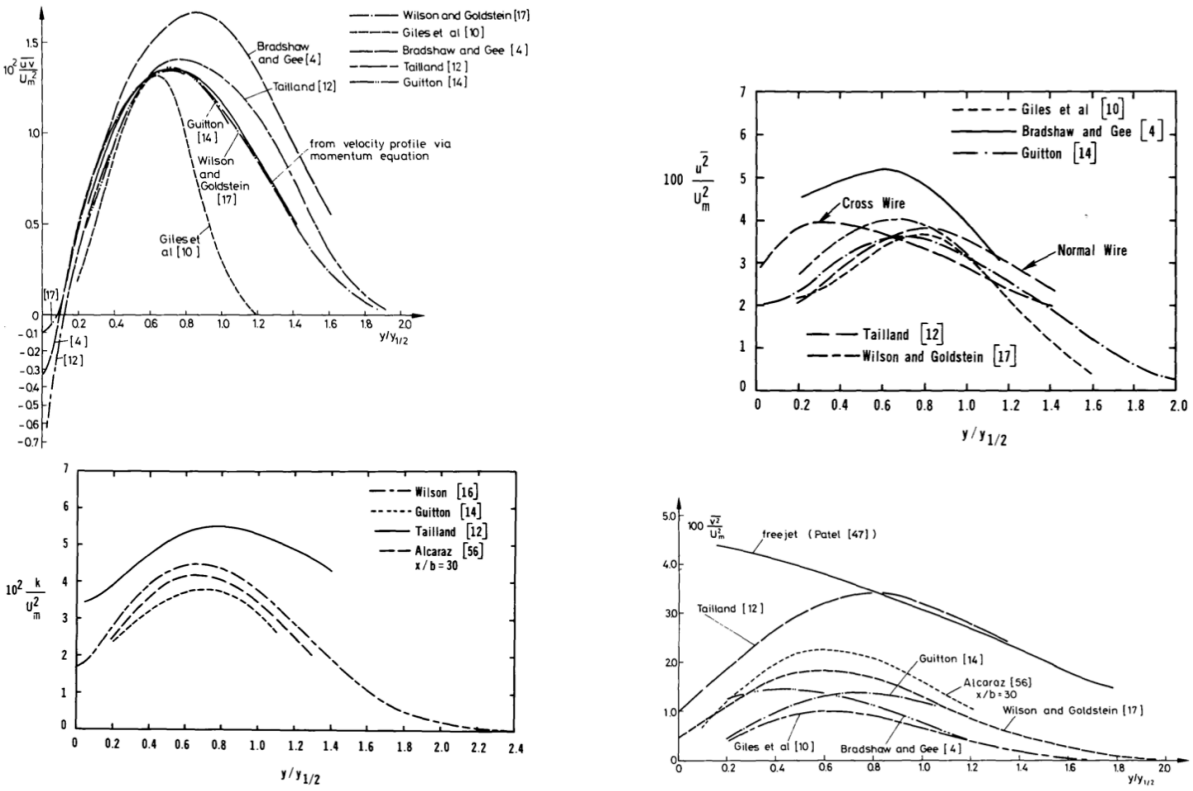


Figure 4.6: Reynolds stresses and turbulent kinetic energy profiles across the turbulent wall jet in stagnant surroundings [13].

5

Computational Fluid Dynamics

Computational fluid dynamics is field of fluid mechanics where the partial differential equations involved in fluids are solved numerically using computers. Generally, the Navier Stokes equations are solved in a numerical way. The domain of interest is discretised, forming a grid of cells. The Navier Stokes equations are solved in their numerical form on this grid normally using finite volume methods.

This chapter will first give a broad overview of CFD simulations. After that literature around RANS CFD models is explored. The focus will mainly be on the simpler and more commonly used linear models as these will be used during the research. Especially the terms in the equations are explored and their relative interaction.

The knowledge gained here is then used as a solid background for the next step; data-driven methods for improving turbulence modelling, as discussed in chapter 6.

5.1. Navier Stokes Equations

In physics, and especially in fluid mechanics, the Navier Stokes equations are used to describe a flow field. The equations resemble the conservation of mass, momentum, and energy. In equation 5.1 and 5.2 the mass and momentum equations are shown respectively.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \tag{5.1}$$

$$\rho \frac{D\mathbf{u}}{Dt} + \nabla p - \nabla \cdot \tau - \rho g = 0 \tag{5.2}$$

Where $\mathbf{u}$ is the flow velocity vector, ρ the density of the fluid, p is the pressure, g represents the body accelerations acting on the continuum, and τ the deviatoric stress tensor.

The problem with solving these equations numerically is the large range of length and time scales in a fluid. Because of this, the grid where the equations are solved should be fine enough to capture the smallest length scales, but also large enough to capture the largest length scales. Also, the time step should be small enough to capture the smallest time scales, but large enough as well to make sure the largest time scales are captured.

The ratio between the smallest and largest length scales can be derived following Kolmogorov's hypotheses. It states that kinetic energy enters turbulent flow at the largest scales of motion. This then breaks down to smaller scales all the way to the smaller scales. It can then be derived that the ratio between the smallest and largest length scales is given by equation 5.3 [19].

$$\frac{\eta}{L} \sim \mathrm{Re}^{-\frac{3}{4}}, \quad \frac{u_\eta}{U} \sim \mathrm{Re}^{-\frac{1}{4}}, \quad \frac{\tau_\eta}{T} \sim \mathrm{Re}^{-\frac{1}{2}} \tag{5.3}$$

As shown the ratio between the smallest and largest length scales is directly linked to the Reynolds number. For small Reynolds numbers, the ratio is thus small enough that a direct numerical approach is possible for solving the Navier Stokes equations. Although for engineering applications the Reynolds numbers can be of order 10^8. For these applications, it becomes unfeasible to solve the equations directly.

5.2. Turbulence Modelling

As discussed in the previous section solving the Navier Stokes equations numerically for larger Reynolds numbers becomes computationally too expensive. Normally for engineering purposes, we are interested in the largest length scales. However, the effect of the smaller length scales can not be neglected. Having a fine grid and small-time step to numerically solve them directly is not feasible. Therefore mathematical models are constructed to predict the effect of those smaller time scales. Multiple methods have been found ranging in accuracy and computational cost.

5.2.1. Large Eddy Simulations

An example of a method to reduce the computational cost is large-eddy simulations. In this approach, the Navier Stokes equations are solved numerically up to a determined length scale. Further below this scale the effects on the larger scales are modelled with the help of a mathematical model. LES thus spatially filters the direct numerical approach.

This approach can greatly reduce the computation time compared to direct numerical simulations. Also, the nature of the simulation is still unsteady as only the smallest time scales are filtered out.

5.2.2. Reynolds Averaged Navier Stokes

Another example of approaching this problem is the Reynolds averaged Navier Stokes approach. In this approach, the assumption is made that, especially for engineering practices, only the time-averaged flow variables are of interest. One can imagine that when designing a road car only the average drag over time is of importance. Assuming the drag does not oscillate severely, the fluctuation will not matter that much. What is more interesting is how it impacts fuel consumption over time.

The smaller fluctuations in the flow variables do have an impact though on the time-averaged solution, and therefore a mathematical model is constructed to model these effects.

The flow variables are thus divided into two parts, a time averaged part and a fluctuating part. The Reynolds averaged Navier Stokes equations can be derived from the original Navier Stokes equations and are shown in equation 5.5. The RANS equation shown here is in its in-compressible form and is written using the Einstein notation.

$$\frac{\partial \bar{u}_i}{\partial x_j} = 0 \tag{5.4}$$

$$\rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \rho \bar{f}_i + \frac{\partial}{\partial x_j}\Big[-\bar{p}\delta_{ij} + \mu\Big(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\Big) - \rho\overline{u_i' u_j'}\Big] \tag{5.5}$$

Here the averaged values are given with a bar over the variable and the fluctuating terms with an apostrophe. As can be seen from the equation all the terms concerning time are dropped out as we are only interested in the time-averaged values. An extra term emerged though which represents the effects of the fluctuating part of the velocity field. Because of this term, additional modelling is needed to close the RANS equations. This term is often called the Reynolds stress term.

For modelling of the Reynolds stress term, several approaches are constructed. The relevant approaches to this research will be discussed.

5.3. RANS models

As discussed in the previous section, when using the RANS approach an additional model has to be constructed for the Reynolds stress term. Several methods have been proposed throughout the years. The models range in their accuracy and robustness. The common factor is modelling the Reynolds stress term in one way or another. By doing so the assumption that the flow is steady is already made as the governing equations determine the time-averaged values.

5.3.1. Linear Eddy Viscosity Models

The most straightforward approach is the linear eddy viscosity model. In these models, the Reynolds stress term is separated into an isotropic and a deviatoric part, as shown in equation 5.6.

$$R_{ij} = -\overline{u_i' u_j'} = 2\nu_t S_{ij} - \frac{2}{3}\delta_{ij} k \tag{5.6}$$

Where $k = \frac{1}{2}\overline{u_i' u_i'}$ is the turbulent kinetic energy, ν_t the eddy viscosity and S_{ij} is the mean strain rate tensor.

The observation is made that turbulence leads to momentum exchange between fluid elements. The deviatoric part of the Reynolds stress tensor is assumed to be proportional with eddy viscosity to the mean shear rate. This assumption is known as the Boussinesq assumption.

The eddy viscosity can be calculated in different ways. It can be calculated directly from known values in the mean flow or one or more equations can be used to calculate the field. An example is the $k-\epsilon$ model [13]. This is a two-equation model, meaning it uses two equations to calculate the eddy viscosity, see equation 5.7.

$$\nu_t = C_D \frac{k^2}{\epsilon} \tag{5.7}$$

Where C_D is a model constant and k, the turbulent kinetic energy, and ϵ, the turbulent dissipation rate, two extra variables which are to be solved for with two extra transport equations.

These models are quite efficient as only four equations in total have to be solved, the mass, momentum, and two transport equations. The problem here is that eddy viscosity models assume the Reynolds stress is proportional to the mean shear rate. This leads to the problem that they can not distinguish the effects of the individual components of the Reynolds stress tensor. So an-isotropic influences are failed to be predicted, such as streamline curvature and directional forces.

Another example is the $k-\omega$ model where the specific turbulent dissipation rate is solved for and used to calculate the eddy viscosity [37]. The different models have several advantages and disadvantages with respect to each other. Also improvements and adaptations have been made such as the improved $k-\omega$ model [38] and the blending of the $k-\omega$ and $k-\epsilon$ models [15].

5.3.2. Reynolds Stress Models

Another approach of modelling the Reynolds stress term is by directly solving the model's transport equations. Several models have been constructed, modelling the terms more accurately than the other. By design, these models give a far improved result over the linear eddy viscosity model. They do not have the Boussinesq assumption which means they are much stronger in predicting flows with strong curvature or swirl, anisotropic turbulence.

The big problem with RSM models is that they are much more computational expensive because of the extra transport equations to be solved for. Also because of these extra equations, the convergence of an RSM model can be very slow or even unstable.

6

Data Driven Methods

As discussed in chapter 5 several methods have been developed for solving the problem of the large range of length scales present in turbulent flows. Every method has is its own assumptions and generally its a question of how much time to spend on achieving better accuracy. In general engineering applications, it is often not possible to use the higher fidelity models as resources are limited. In recent times methods have been developed to improve the lower fidelity models. This only has been kick-started recently due to rapid development in machine learning and the availability of larger data sets. In the last decades, as computers have become more powerful, DNS simulations are more and more feasible for the low Reynolds number problems. The data generated is invaluable to gather modelling insights [4]. There has been a shift in focus from constructing new models towards improving models using machine learning techniques.

Several methods have been developed in improving turbulence modelling and are discussed in the sections below.

6.1. Model Coefficients

Calibrating or tuning the coefficients of an existing model focuses on incorporating evidence from experimental or DNS data [4]. It is assumed the model coefficients are the dominant source of uncertainty in the model. This process has led to a sharp increase in model variants. A definite drawback is the difficulty of assessing its predictive capabilities in general [4]. Tuning the coefficients can be done directly, by getting the variable of interest as close to experimental or DNS data as possible. Another, more used method is statistical inference, where the uncertainty of the data is also accounted for. Potential discrepancies between the model prediction and the data can also be included [4]. The statistical inference is based on the Bayes theorem.

This approach is still fairly recent. In 2011 the first articles appeared [17] [3] aiming to generate a probability function for model coefficients. They used DNS data for plane channel flows to determine the posterior probability distributions for model parameters. They did this for several turbulence models and found that not for every model an improvement could be found. This means that the model is more limited by the assumptions than by wrongly calibrated coefficients. Later studies [6][7][14][22] continued with this approach. Several flow cases were used to infer a probability function for the model coefficients of interest. It has been tried to combine the probability functions of the flow cases to be able to predict an unseen case. Most of the studies used a Markov chain Monte Carlo method to construct the probability density function. More recently, experimental data have been used for this purpose [23].

Although improvements have been found in every study the problem remains that the model errors are still significant, as shown in figure 6.1. Also, a bigger problem is that the calibrated coefficients hardly generalise across a large range of problems. As a result, no improved models are found for engineering applications. However, these methods can be used to improve the predictions for specific engineering problems.

6.2. Model Form Error

Earlier efforts were made to quantify the model form error of RANS turbulence models[16]. A Reynolds stress discrepancy tensor was constructed to account for the uncertainties. This, in contrast to calibrating model coefficients, only suffers from the assumptions made deriving the Reynolds averaged Navier Stokes equations

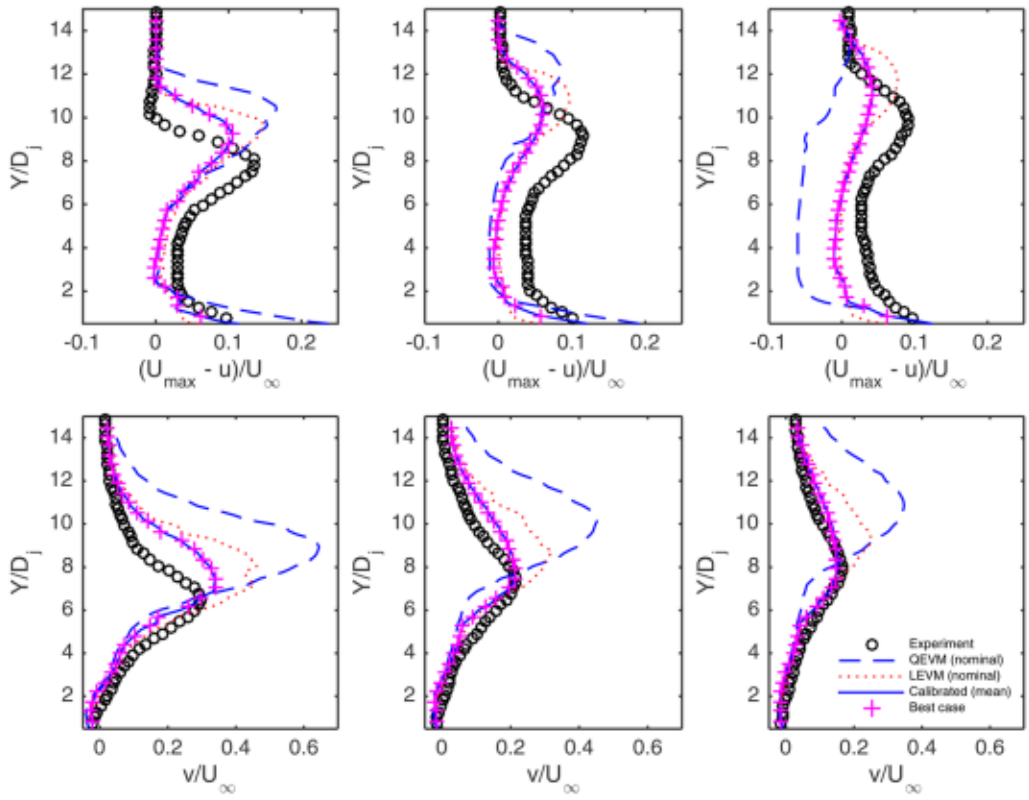


Figure 6.1: Streamwise velocity plots of a jet in cross flow at three locations downstream of the jet. Two models are compared with experimental data. A calibrated model using the experimental data is also shown in the plot. [23]

and not from the assumptions made in the model. They described the discrepancy tensor by stochastic differential equations which are simpler than the RST equations. This was the starting point for this approach and many following studies.

6.2.1. Inference Based

Two avenues have been discovered in the past twenty years. The first and earliest method uses the difference in a variable of interest, such as the velocity field, to infer the discrepancy in a turbulence field. Studies have been performed on determining the discrepancy in the Reynolds stress tensor [16], its magnitude and anisotropy [5][26], or on the turbulent viscosity field [34]. The main idea is to compare a variable of interest, for example, the velocity field, from model to high fidelity data. The high fidelity data is used to statistically infer a discrepancy field needed to correct the model to better fit the high fidelity data. This approach involves a large-scale statistical inference, where an optimum, or maximum a priori is sought to be found in the discrepancy field. This has been done using adjoint-based methods [9] to efficiently determine the gradients for the optimisation progress. Also, gradient-free methods have been reviewed, using Ensemble Kalman methods [11].

In recent studies [18][31][27] a correction factor has been used to account for the uncertainty. In these studies, the correction factor is applied to the production term of the ω equation in $k-\omega$ turbulence model. Using the described paradigm of field inversion, from the differences in the velocity field, they infer the correction field used to correct the model form error. This approach shows clearly where the model is lacking in its physics, as the correction factor is applied to the model transport equations. In contrast, the discrepancy fields in earlier discussed studies will only give info about the error made in the output of the model.

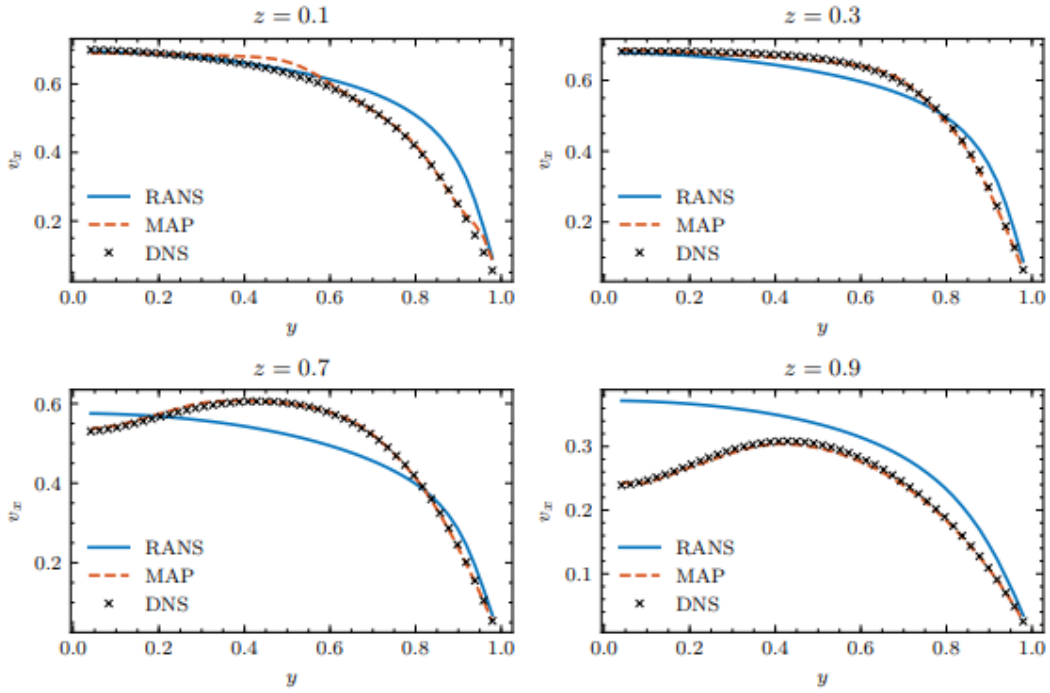


Figure 6.2: Streamwise velocity profiles along various spanwise locations in a square duct. A standard k-ω turbulence is compared against DNS data and the results of the standard model corrected by an infered correction field.

It is in some cases a better alternative than optimising the model coefficients, looking only at the accuracy, as model form errors are also accounted for. A drawback is that these methods are computationally heavy. The adjoint methodology used for determining the gradients to infer the discrepancy field introduces more transport equations to be solved.

Also, the capabilities of being able to correct the model form errors depend strongly on where the discrepancy field, or correction factor in the case of figure 6.2. When the location of this field is still within the model, some assumptions will remain in effect.

In some of these studies, several machine learning algorithms have been used to train a model which uses easily extract-able features from the low fidelity model to predict the discrepancy field needed to correct the model form errors.

6.2.2. Direct Machine Learning

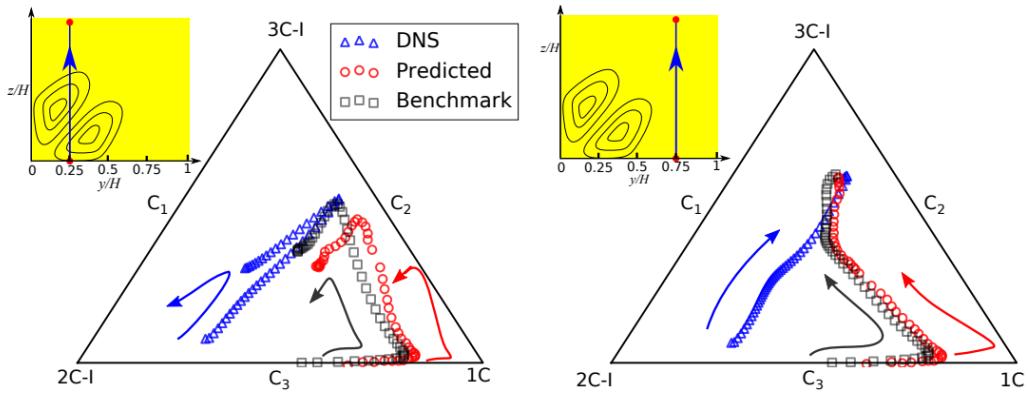
The second approach uses machine learning to construct a function for the discrepancies somewhere in the model. This approach was first presented as reconstructing discrepancies in the anisotropy tensor of the Reynolds stress tensor [30] and was a start for much more research. Full-field high fidelity data is needed as the discrepancy is directly calculated, taking the difference with the model. A feature set is selected, consisting of functions of readily available model variables, such as velocity gradients and turbulent quantities. The machine learning model is trained to map the chosen discrepancy field from selected features.

While training a model for the Reynolds stress tensor it is important that the physics are considered. For the Reynolds stress term, rotational invariance must be ensured [4]. A strategy used in later research is based on this and tries to formulate a generalised expansion of the Reynolds stress tensor. The assumption is made that the stress tensor is only dependent on the mean velocity gradient [19]. The stress tensor can then be built up as a sum of known functions of the symmetric and anti-symmetric part of the velocity gradient tensor, as shown in equation 6.1. Here $\mathcal{T}$ represents functions of the symmetric and anti-symmetric part of the velocity gradient tensor. Machine learning algorithms are then used to learn the coefficients c in order to be able to predict the stress tensor.

$$\tau = \sum_{n=1}^{N} c^{(n)} \mathcal{T}^{(n)} \tag{6.1}$$

In the initial study perturbations in the eigenvalues of the Reynolds stress tensor were used to correct the turbulence model. Later more studies were performed with a more understandable perturbation strategy. Instead of the eigenvalues, which may be difficult to comprehend afterward, the magnitude, anisotropy, and orientation of the Reynolds stress tensor were used [33][39].

In figure 6.3 and 6.4 the potential of the approach is shown, where it becomes clear the baseline RANS model can significantly be improved.



(a) $y/H = 0.25$ (b) $y/H = 0.75$

Figure 6.3: Barycentric map of the predicted Reynolds stress anisotropy for the rest flow (Re = 3500), learned from the training flows (Re = 2200, 2600 and 2900) The prediction results on two streamwise locations at y/H = 0.25 and 0.75 are compared with the corresponding baseline RANS model and DNS results in panels (a) and (b), respectively. [33]

In recent studies, a slightly different approach is taken. Instead of trying to correct all the model form errors like in earlier studies a RANS model is selected to be improved upon. The k-ω turbulence model is used where only the omega-transport equation is solved. Then by comparing to high fidelity data, the residual of the k-transport equation is computed, which would be needed to get the correct eddy viscosity stress tensor. Using the known variables in the simulation a library of candidate functions is built. Sparse regression machine learning models are used to learn coefficients to map the determined residual from the candidate

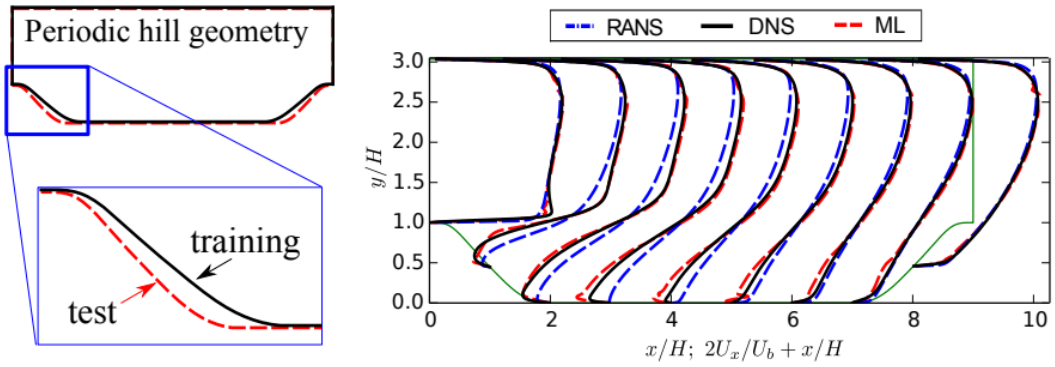


Figure 6.4: The stream-wise velocity by stabilised RSM at x/H = 1, 2, ..., 8. The test flow is the flow over periodic hill at Re = 5600. The training flow is at the same Reynolds number but has a steeper hill profile [39]

functions. This results in a correction model which is used to improve the standard RANS solution. This method is named SpaRTA [24].

The advantage of these approaches over the paradigm of field inversion is that discrepancy functions can be constructed that can be employed within a class of flows sharing similar features [4]. This makes it much more generalisable as in essence new turbulence models are created, instead of correcting existing ones. Also, as there is no need to infer the desired discrepancy field as all the turbulence variables are present, this approach is much more computationally efficient. A big drawback however is that full-field high fidelity data is needed, whereas with field inversion limited data can be used. This makes it more desirable for the research community and less for the engineering community.

6.2.3. Combining Field Inversion and Machine Learning

The two techniques for improving upon the standard simulations can also be combined. In this approach, the paradigm of field inversion is used to infer a discrepancy field or correction factor using high fidelity data. After this first step machine learning is then used to formulate a functional model in order to predict the discrepancy field or correction factor for unseen cases.

When machine learning is directly applied to high-fidelity data, inconsistencies may arise. The turbulence variables in a RANS model may have a different meaning and usage compared to high fidelity DNS data. Turbulence models are formulated to represent first and second moments so turbulence values may assume a more operational role rather than the actual physical meaning [4]. This causes problems with the direct machine learning approach which can be alleviated by statistical inference.

After the first step of statistical inference, machine learning is used to construct a model that is capable of mapping the discrepancy from mean flow values and turbulence variables. The first reviews of this approach were quite promising with convincing improvements over the base models [5][18]. A schematic of this approach is shown in figure 6.5.

A big advantage of this method is that it can work with just limited data [5][26]. It has been shown improvements can be found with just using surface pressures or skin friction [28], figure 6.6. This is data that is much easier to extract from experiments and thus for engineering practices these methods can be a powerful tool. In section 7 this approach will be further studied.

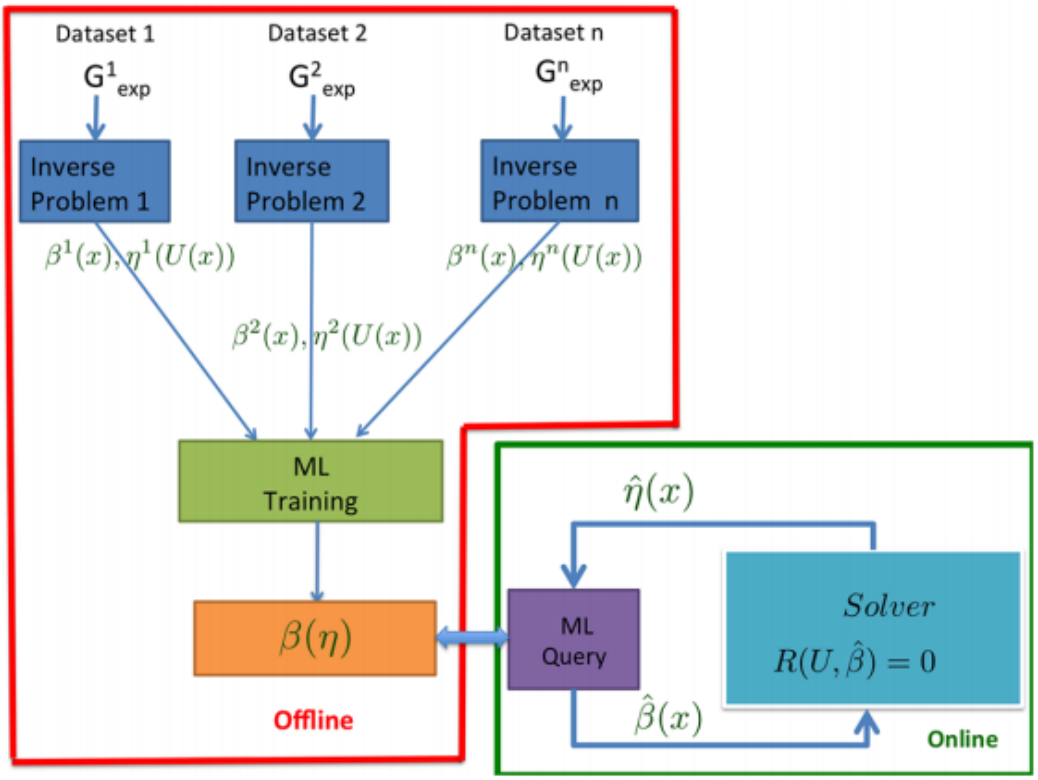


Figure 6.5: Schematic of field inversion and machine-learning framework [27]

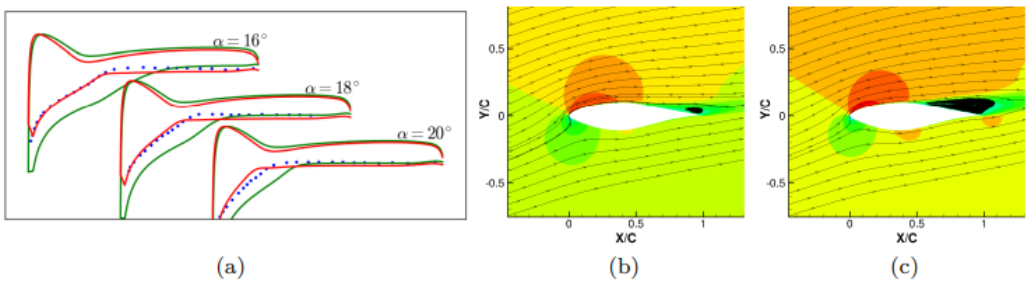


Figure 6.6: Field inversion and machine learning applied to turbulent flows over airfoils. (a) Pressure plot for the airfoil at different angles of attack (Green: baseline model, Red: Improved model by field inversion and machine learning, Blue: Experimental data. (b) Flow field predicted by the baseline model. (c) Flow field predicted by the improved model. [28]

III

Methodology

7

Field Inversion

For this research, the goal is to get the flow field of a low order simulation to better match high fidelity data, in this case, PIV measurement data. Often the lower order simulations do not give a satisfactory result for engineering or research purposes, while higher fidelity solutions are too computational heavy. Although predicting the general flow behaviour, the lower fidelity simulations often are still lacking in the secondary flow features. Therefore in this research, we would like to know-how, and how well, we can change the simulation in order to better match PIV data.

As discussed in chapter 6 several approaches to this problem have been taken in the last decade. A discrepancy field can be determined using a chosen field variable used somewhere in the model in order to correct the baseline model. This discrepancy field is often determined with the Reynolds stresses or a variable related to it. The discrepancy of the model is sought to be found in the variables it models. The problem in this case, however, is that only the velocity field is available as experimental PIV data will be used. Therefore the paradigm of field inversion is to be used to determine a discrepancy field in a less direct approach.

A normal simulation code is called a forward model. Knowledge about the problem and model constants are used as inputs and velocities, or other flow variables, are obtained as output. However in this case the inverse of this process is needed. From a velocity field we would like to know what model parameters we need to use, or what discrepancy there is in a certain field that needs to be corrected.

An example of this is the airfoil design of trans-sonic aircraft. In this example, it is important that the velocity of air over the airfoil does not exceed the speed of sound as the drag will go up drastically. With simulation codes, we can easily compute the Mach number of the flow around the airfoil when we know the airfoil and the flying conditions. However, if this problem could be inverted, setting a maximum Mach number as input and outputting the shape of the airfoil and flying conditions needed to achieve this would make this optimisation process much easier. The problem here is that multiple airfoil shapes and flying conditions can achieve this Mach number constraint. The forward problem often only has one solution, but the inverse problem has multiple, or an infinite amount of, solutions.

Therefore the inverse problem is often tackled from a probability point of view. Instead of knowing the exact input variables, we assume our prior information has a probability distribution. After this information is put through the simulation we will get a posterior distribution. Using the analogy of the trans-sonic aircraft, instead of using exact inputs of for example velocity, angle of attack, and Reynolds number, a probability distribution will be used. Using the simulation code the output variables with their posterior probability distributions, such as the lift and drag coefficient can be computed, as shown in black in figure 7.1 .

When the lift coefficient is measured the simulation code can be inverted to get increased knowledge of the flight conditions. For this Bayesian inference can be used and this is the core of the field inversion process. The probability of the flight conditions, the hypothesis, is called the prior probability and can be written as $P(H)$. The probability of the measured lift coefficient, the evidence, given the values of the flight condition is called the likelihood and can be written as $P(E|H)$. Using the knowledge gained by means of the measured lift coefficient, given the flight conditions, we can update the prior probability to get the posterior probability, using equation 7.1.

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{7.1}$$

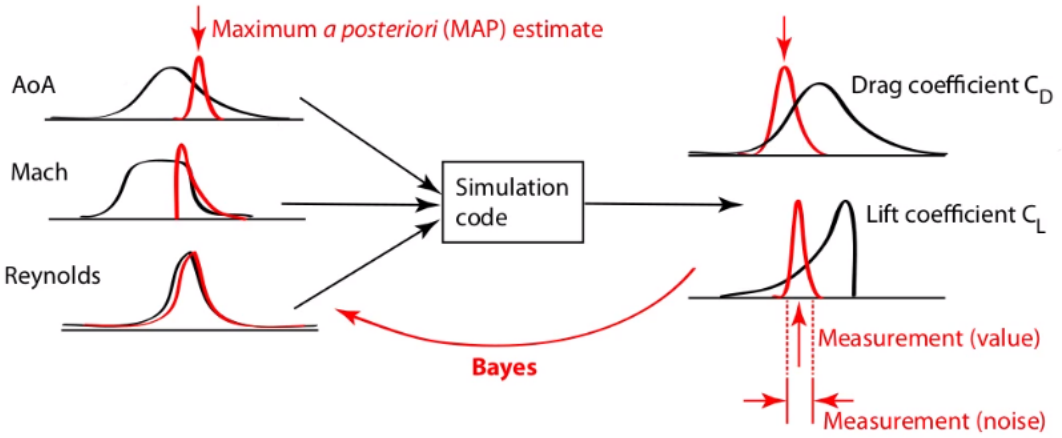


Figure 7.1: Schematic drawing of the airfoil design problem in a probability framework.

The posterior probability, $P(H|E)$, is the probability of the flight conditions given the knowledge of the lift coefficient. The $P(E)$ is termed the marginal likelihood, which is the same factor for all hypotheses considered. Therefore as this is only a scaling factor it is often left out.

Using this method one can see in figure 7.1 that this extra knowledge can increase the knowledge of the prior conditions which in turn can improve the predictions on for example the drag coefficient.

Also in this research, the problem will be approached from a probabilistic standpoint. Bayes's theorem is going to be used to infer a discrepancy field in order to better match the PIV data. The discrepancy field is analogous to the flight conditions in the above example and the PIV data to the measured lift coefficient. The goal is to better predict the flow field and its secondary features which are analogous to the drag coefficient in this example.

In this chapter, the paradigm of field inversion is going to be discussed. The decisions behind the chosen path will be illustrated with the relevant literature as background.

7.1. Problem Formulation

In this section, the problem formulation, illustrated in the introduction, will be mathematically laid out.

A physical system which can be described using a set of non-linear equations or partial differential equations, such as the flows simulated using CFD, can be written down as:

$$\mathscr{R}_T(\boldsymbol{Q}_T(\boldsymbol{x},t)) = \boldsymbol{0} \tag{7.2}$$

Here the mathematical operator $\mathscr{R}_T$ contains the governing equations of the physical system. Such a truth model will be based around model variables, denoted by $\boldsymbol{Q}$. The truth model is dependent on the spatial coordinates and time.

Often it is not possible or very hard to determine such a truth model. Either not all factors at play are known or it is just too computationally expensive to implement all elements. Therefore physical systems are often modelled. A model of the system is created, taking into account only the relevant physical phenomena.

Such a model can be written down as:

$$\mathscr{R}_m(\boldsymbol{Q}_m(\boldsymbol{x},t),\mathscr{M}) = \boldsymbol{0} \tag{7.3}$$

In this case, the mathematical operator $\mathscr{R}_m$ contains the model governing equation where $\mathscr{R}_m \neq \mathscr{R}_T$. The model equations differ from the physical system set of non-linear partial differential equations. Either due to a lack of understanding of the underlying physics or from simplifying the problem the model equations built to best represent the physical system. The model variables $\boldsymbol{Q}_m$ are therefore also different to $\boldsymbol{Q}_T$. Sometimes from the modelling of the physical system, unclosed functions $\mathscr{M}$ arise. Exact values for these functions can result in accurate solutions however these are often difficult to obtain.

Therefore often a second set of model variables $\boldsymbol{Q}_s(\boldsymbol{x},t)$ are introduced. These variables can be determined by a second set of model equations $\mathscr{R}_s(\boldsymbol{Q}_m,\boldsymbol{Q}_s) = \boldsymbol{0}$ which themselves also introduce assumptions

which are not exact. The objective is to approximate for $\mathscr{M} \approx \boldsymbol{M}(\boldsymbol{Q}_m, \boldsymbol{Q}_s)$. A model for a physical system is thus often modelled as:

$$\mathscr{R}(\boldsymbol{Q}, \boldsymbol{M}(\boldsymbol{Q})) = 0 \tag{7.4}$$

Where $\mathscr{R}$ and $\boldsymbol{M}$ are a combination of the model equations and variables and their second set of model equations and variables to close the system.

As discussed in chapter 5, DNS is an example of a truth model $\mathscr{R}_T$ while RANS is an example of a model which only approximates the truth. The Reynolds averaged Navier Stokes equations are an example of the primary model equations $\mathscr{R}_m$. The extra equations used in the different models for closing the system are an example of the secondary model equations $\mathscr{R}_s$. The formulation for the eddy viscosity is in this case the model M.

For many RANS CFD models generalisability is still an issue as discussed in chapter 5. In order to get a valid result, the empirical constants in the model have to be adapted for the specific case. This leads to either a solution deviating from the actual physics or a difficult task in tweaking the model constants to match the solution to experimentally obtained data.

In the paradigm of field inversion, a spatially varying corrective field function is proposed to correct the model to better represent the true physical system.

The new system looks like this:

$$\mathscr{R}(\boldsymbol{Q}, \boldsymbol{M}(\boldsymbol{Q}, \beta(\omega))) = \boldsymbol{0} \tag{7.5}$$

Where $\beta(\omega)$ is the spatially varying corrective field term. The goal is to use β to correct the model equations to get a solution close to the high fidelity or experimental data. This replaces the need to empirically define the model constants.

This process can be repeated for a wide class of problems that are representative of the physical phenomena. The correction fields found can then be used as an input for machine learning algorithms to find a functional relationship between β and the input features η.

A functional relationship $\beta(\eta)$ can then be found, where $\eta(\boldsymbol{Q})$ are the input features which are available in the model. The values for β can then be mapped on the grid using the input flow field features outputted from the model for a case outside the training set. Unfortunately, this last step falls outside the scope of this research. The focus will be on finding the correction field for the complex case of an underflow weir with experimental PIV data as reference data.

7.2. Correction factor

The location of where the correction is going to be applied in the model has been varied in the literature. Earlier studies applied the correction on the eddy viscosity or the Reynolds stresses in order to maximise the model form error correction [30][33][39][12]. Later, studies were also performed on correcting fields deeper in the model itself. Studies have been done on correcting the production term in the k-equation [18][24] or in the ω-equation [27][31].

As discussed in chapter 6 the location of the correction influences the capabilities of the inferred model. Van Korlaar [31] compared two formulations of the correction factor. First, a correction was applied on the production term of the ω-equation and secondly applied on the eigendecomposition of the anisotropy tensor. The first correction is thus applied very deep in the model, while the second correction is applied outside of the model. He showed that the second formulation was much more capable of improving the model to represent the data. Especially in the case of square duct flow, the first representation was not capable of correcting the model. In that case, the main problem is the assumption made in the model itself. Therefore the second formulation was superior as it is capable of correcting these. The disadvantage however is that in the second phase of the paradigm a function is tried to be defined using machine learning that is capable of predicting the correction. It was shown that with the first formulation the machine learning model is much more capable of generalising over a wide range of cases. However, in cases where the predictions of the basic model were already quite good, the two formulations were equal in their capabilities of correcting, as can be seen in figure 7.2. Finally, it has been shown that a specific value of a term in the transport equations is in itself not as important as the values of terms relative to each other [20]. Therefore, when correcting a term deep in the model, the argument can be made that it is less important which term is chosen to be corrected.

In this research, the first formulation is going to be used. The basis $k-\omega$ model used predicts the flow quite well, so no real difference is to be expected between the two formulations. However, in the future, the goal is as well to create a new model that generalises well across a range of cases.

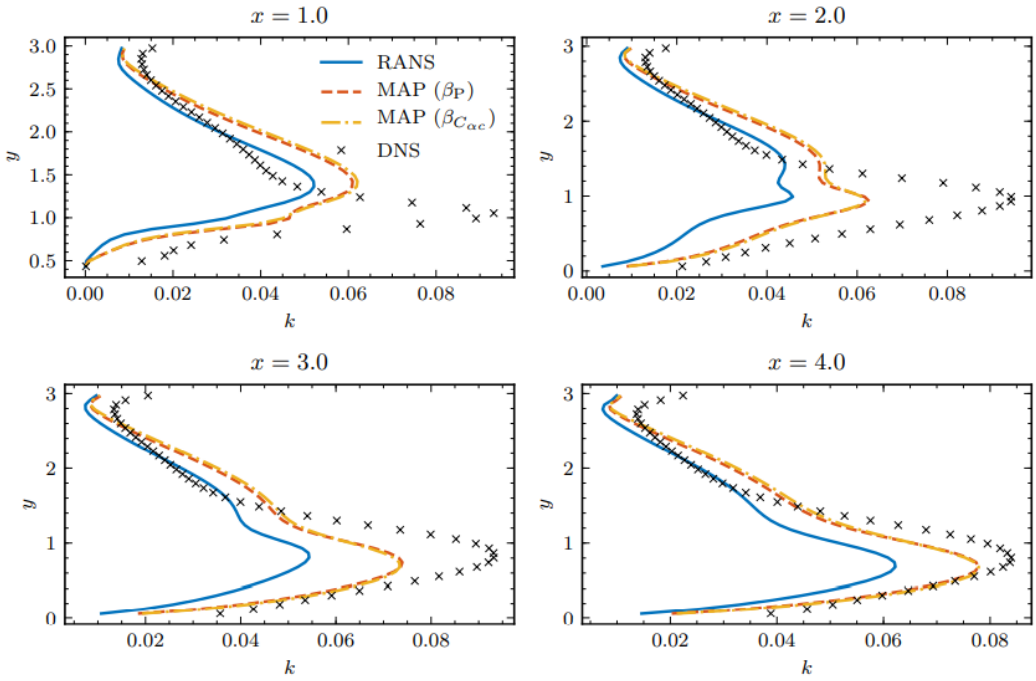


Figure 7.2: Comparison of the capabilities to correct k between the two correction formulations for the case of a periodic hill[31]

Figure 7.2 shows that the second approach, correcting the eigendecomposition of the anisotropy tensor, is, although marginally, in fact, better in some regions. But there is still a big gap in the reference DNS data. It can not be expected that a RANS CFD model, while corrected, can achieve the performance of a DNS simulation. There has definitely been a good step in the right direction but the new turbulence model will still suffer from its limitations.

7.3. Statistical Inference

The goal for field inversion is to find a corrective field term that corrects the model to give a solution closest to the experimental data. This can be mathematically written as: $h(\boldsymbol{\beta}_{true}) = \boldsymbol{d}_{true}$ where $\boldsymbol{\beta}$ is a spatially varying correction to the model. $h(\boldsymbol{\beta})$ is the forward model, in this research the $k-\omega$ RANS model. In this section, the procedure for field inversion is elaborated upon.

Following Bayes's theorem the posterior distribution is given by:

$$p(\boldsymbol{\beta}|\boldsymbol{d}) = \frac{p(\boldsymbol{d}|\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\boldsymbol{d})} \tag{7.6}$$

Where $p(\boldsymbol{\beta})$ is the prior distribution, $p(\boldsymbol{d})$ is the evidence and $p(\boldsymbol{d}|\boldsymbol{\beta})$ the likelihood distribution. The goal is to find the highest probability of the posterior distribution $p(\boldsymbol{\beta}|\boldsymbol{d})$. As this is a multivariate problem the posterior distribution can not be simply computed. One can compute single points of the multivariate probability distribution but not its complete space. Therefore the field inversion paradigm becomes an optimisation process of finding the highest probability in the multivariate posterior probability distribution, the maximum a posteriori estimate (MAP).

When Gaussian distributions are assumed for the distributions, equation 7.6 can be rewritten as:

$$\beta_{map} = \arg\min_{\beta} \frac{1}{2}\Big[\big(\boldsymbol{d}-h(\beta)\big)^T {\boldsymbol{C_m}}^{-1}\big(\boldsymbol{d}-h(\beta)\big) + \big(\beta-\beta_{prior}\big)^T {\boldsymbol{C_\beta}}^{-1}\big(\beta-\beta_{prior}\big)\Big] \tag{7.7}$$

Where $\boldsymbol{C_m}$ and $\boldsymbol{C}_\beta$ are the observational and prior covariance and β_{map} is the maximum a-posteriori estimate for β. Finding this value and thus minimising the equation is the goal of the field inversion phase.

Therefore equation 7.7 is going to be the cost function in the optimisation process.

One advantage of using the Bayesian approach is that the confidence in the data and prior corrective terms can be expressed and used in determining the values for the corrective field. As well the posterior covariance matrix of the MAP solution can be calculated which is the inverse of the Hessian of the objective function at the MAP point. This estimate of the uncertainty of the MAP solution is an important variable as it expresses the confidence in the correction field. But it can be used in the training of a predictive model of $\beta(\eta)$ in the machine learning process, as well. The process described above is almost identical to a normal least squares method with an added Tikhonov regularisation term.

The observational covariance matrix can be built from the data. In this research, the data points are assumed to be independent of each other. Therefore only terms on the diagonal are to be expected. Multiple snapshots of the flow field are taken using the PIV technique. The mean of the flow is used as reference data which is determined by the average of the snapshots. This variance of the multiple snapshots is used for the observational matrix and is calculated as shown in equation 7.8. The variance is thus the mean of the velocity fluctuations in the PIV flow field.

$$C_m = \mathbb{E}\left[(d_i - \mathbb{E}[d_i])^2\right] \tag{7.8}$$

The prior covariance can be estimated per case. An initial guess for the variance is used to sample multiple correction fields using a Gaussian distribution. These samples of the correction field are then propagated through the simulation. From these results, an indication can be formed of the effect of the variance on the probability distribution of the resulting flow field. The reference data should lay within the $\pm 2\sigma$ limits resulting from the propagated corrective term samples. When this is not the case or the variance is too large, the prior co-variance should be adjusted.

7.4. Optimisation Approach

The key obstacle, as might already be apparent, is that the optimisation problem is highly dimensional. The β is a correction field with the same dimensions as the other flow field variables. For higher Reynolds numbers the number of cells can go well into the millions.

Ordinary gradient based optimisation algorithms are most commonly used. In equation 7.9 the most simple gradient descent is shown. Here a step is taken every iteration in the direction of a negative slope. The variable α, the step size, can be determined every step to optimise this process, called a line search.

$$\beta_i^{k+1} = \beta_i^k + \alpha^k \frac{\delta J^k}{\delta \beta_i} \tag{7.9}$$

The problem with these methods is that they are based on knowing the gradient. Applying simple finite difference methods is simply too computationally demanding for this application.

The most used approach in literature is the adjoint approach, based on Lagrange multipliers [31][18][27][5]. This methodology is used to determine gradients of the cost function with respect to β. This can then be used in any gradient-based optimisation algorithm. The adjoint methodology is a fairly computationally efficient method to determine gradients. It uses an equal amount of equations as there are already in the model. Solving for the gradients is thus in the same order as solving for the flow variables, time-wise.

A problem with a gradient descent algorithm is that it will find a local optimum. There is no guarantee the best solution is found. Therefore the initial guess becomes important.

A different other approach is the usage of genetic algorithms. This is an analogue to natural selection. This approach has been used in literature [36][35]. With the use of this approach, the chance of finding a global optimum is much higher. Also, there is no necessity of determining a gradient. However, the convergence is as a result much slower compared to gradient-based methods. Therefore it is regarded as not the best approach for this problem.

Another approach taken in literature is ensemble Kalman filters [11]. With this method, no gradient has to be derived which makes it applicable for high dimensional problems. The problem is approached in a statistical sense. But again for high dimensional problem, a large number of simulations is needed which makes it less suitable than the adjoint methodology.

In this research the work of van Korlaar [31] on the adjoint implementation in OpenFoam will be continued. The adjoint methodology will be discussed in chapter 8. This method of determining the gradients will be used in combination with a simple gradient descent method as shown in equation 7.9.

A line search algorithm is used to find a suitable step size per iteration. The line search implementation from Scipy in Python is used in this research. In this implementation, an inexact line search is performed which uses the Wolfe conditions to find an acceptable step size that reduces the objective function sufficiently.

A schematic overview of the field inversion process is shown in figure 7.3. The loop in teal shows the gradient-based optimisation process. In orange, the line search algorithm is shown. The blocks in cyan are performed using OpenFoam. After a line search, the cost function and its gradient do not have to be calculated again as these will be equal to the last iteration of the line search. With every optimisation iteration, the absolute sum of the gradient field is calculated. When this value is sufficiently low it is assumed a local optimum is found. The initial guess is thus very important to find an optimum which is the global optimum or close to it. However, due to possible stability issues, it is more important to stay close to the initial model. Therefore the initial guess will be a corrective field that does not influence the model, in this case, a uniform field of one.

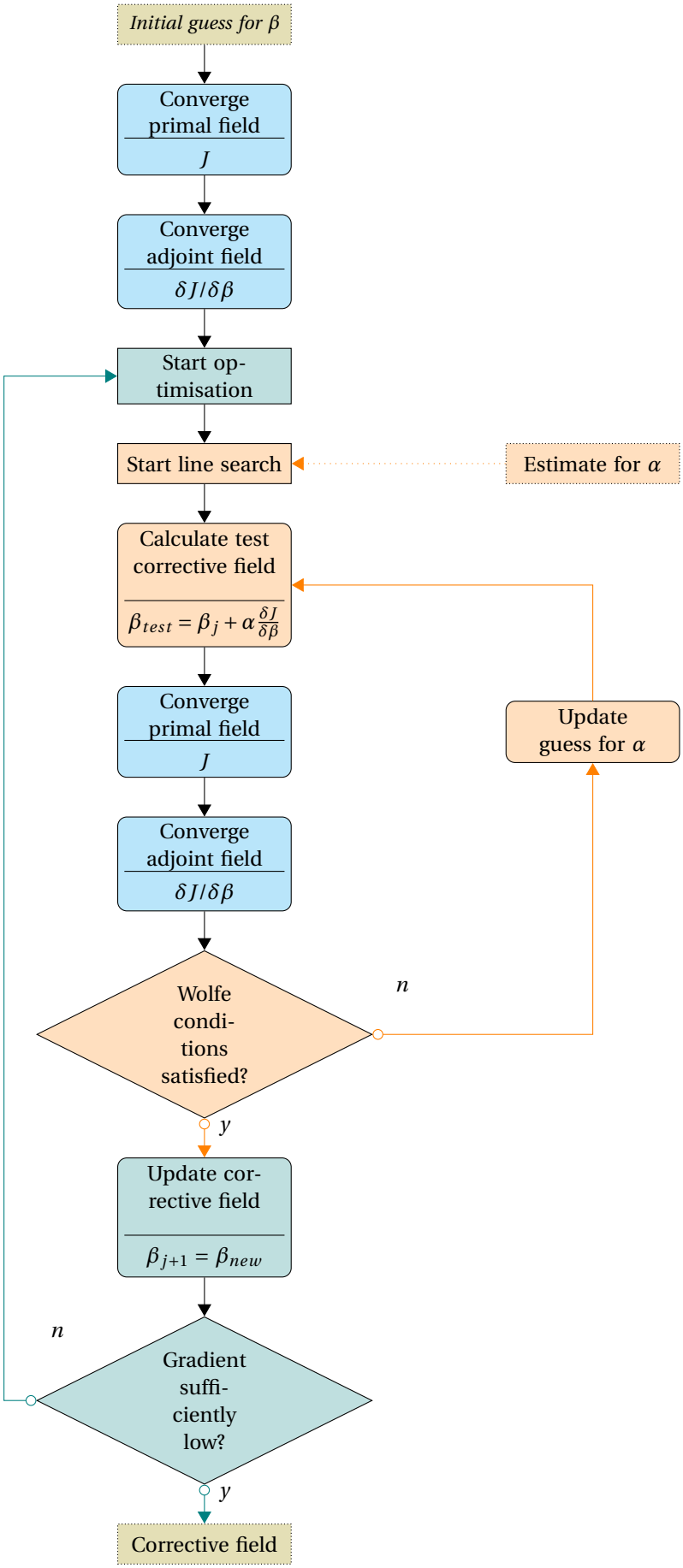


Figure 7.3: Schematic overview of the field inversion process including the gradient based optimisation and line search.

8

Adjoint

As discussed in chapter 7 the adjoint methodology, as discussed in literature [5][31][18][27], is the preferred approach. In this chapter, the steps in the derivation are shown and explained. First, the derivation is roughly shown for the discrete adjoint and the several steps needed to take are explained. After the continuous adjoint will be derived in detail, following van Korlaar [31]. The continuous adjoint is ultimately derived to be used in Open Foam, which makes the implementation a lot more trivial.

8.1. Discrete adjoint

In chapter 7 the cost function has been defined. This has been rewritten in equation 8.1, although in a simpler form. The goal is to find the gradient of the cost function J with respect to β, to be used in the gradient-based optimisation.

$$J(\beta) = \log p(\beta|d) \tag{8.1}$$

The first approach would be taking the derivative of the cost function to β directly. The cost function J is basically a mapping of the parameter input space to a scalar value; $J(\beta) : \mathbb{R}^M \rightarrow \mathbb{R}$. The optimisation can be written down as finding the values for β which give the minimum value for J:

$$\beta_{MAP} = \arg\min_{\beta} J(\beta) \tag{8.2}$$

The problem here is that the cost function becomes a very expensive and complex equation where the simulation code has to be run in order to find the minimum value. With a large number of variables in CFD simulations, this approach is practically unfeasible.

A second approach would be to set the solution, U, of the PDE's as a constraint. The optimisation problem then becomes a constraint optimisation problem.

$$J(U(\beta), \beta) = \log p(\beta|d) \tag{8.3}$$

Here the function J becomes a mapping of the degrees of freedom of β times the degrees of freedom of U; $J : \mathbb{R}^Q \times \mathbb{R}^m \rightarrow \mathbb{R}$.

The optimisation problem then becomes:

$$\beta_{MAP} = \arg\min_{\beta} J(U, \beta) \tag{8.4}$$

$$\mathscr{R}(U, \beta) = 0 \tag{8.5}$$

This constraints the solution to satisfy the governing equations of the system. The cost function can now become much simpler as it does not have to include the governing equations to calculate the velocity field anymore. An example of the cost function is a simple least-squares formulation.

When using this system of equations to determine the gradient of the cost function with respect to β we get:

$$\frac{dJ}{d\beta}\Big|_{\beta_0} = \frac{\partial J}{\partial U}\Big|_{\beta_0}\frac{\partial U}{\partial \beta}\Big|_{\beta_0} + \frac{\partial J}{\partial \beta}\Big|_{\beta_0} \tag{8.6}$$

$$\frac{d\mathscr{R}}{d\beta}\Big|_{\beta_0} = \frac{\partial \mathscr{R}}{\partial U}\Big|_{\beta_0}\frac{\partial U}{\partial \beta}\Big|_{\beta_0} + \frac{\partial \mathscr{R}}{\partial \beta}\Big|_{\beta_0} = 0 \tag{8.7}$$

In these equations, the partial derivatives with respect to β can be solved by hand, except for the gradient of the model variables with respect to β. Because the constraint is used the cost function has become very simple, deriving it with respect to β becomes trivial. The correction factor β is simply used as a multiplier to one of the terms in the governing equation. Therefore the gradient of the governing equation with respect to β is also fairly trivial. The derivatives with respect to the model variables, although tedious, can be worked out by hand. Again, the derivative of the cost function is rather trivial. The derivative of the governing equations with respect to the model variables will form a new set of governing equations.

This means we get a linear system of equations for $\frac{\partial U}{\partial \beta}\Big|_{\beta_0}$ from the constraint formulation.

Although the partial derivatives can be solved by hand, computing them is still rather costly, as can be seen from the dimensional analysis shown below.

$$\frac{\partial \mathscr{R}}{\partial U}\Big|_{\beta_0} \in \mathbb{R}^{Q\times Q} \tag{8.8}$$

$$\frac{\partial \mathscr{R}}{\partial \beta}\Big|_{\beta_0} \in \mathbb{R}^{Q\times M} \tag{8.9}$$

$$\frac{\partial U}{\partial \beta}\Big|_{\beta_0} \in \mathbb{R}^{Q\times M} \tag{8.10}$$

As can be seen, the problem has to be solved M, the size of the β field, times to solve the system. The size of β can be very large in this optimisation problem so this can be very expensive.

To get around this problem the adjoint methodology can be used. Here Lagrange multipliers are used to rewrite the problem as:

$$L(U;\beta) := J(U;\beta) + \psi\mathscr{R}(U;\beta) \tag{8.11}$$

Here the constraint is multiplied by the Lagrange multiplier and added to the objective function to form the Lagrange equation. It still has to be true that $\mathscr{R}(U;\beta) = 0$ for all β's so therefore taking the derivative of this term will also have to be zero. But if we keep this term and divide it into partial differentials we get:

$$\frac{dL}{d\beta} = \frac{dJ}{d\beta} + \psi\frac{d\mathscr{R}}{d\beta} \tag{8.12}$$

$$\frac{dL}{d\beta} = \frac{dJ}{d\beta} + 0 \tag{8.13}$$

$$\frac{dJ}{d\beta} = \frac{\partial J}{\partial \beta} + \frac{\partial J}{\partial U}\frac{\partial U}{\partial \beta} + \psi\Big(\frac{\partial \mathscr{R}}{\partial \beta} + \frac{\partial \mathscr{R}}{\partial U}\frac{\partial U}{\partial \beta}\Big) \tag{8.14}$$

Now as discussed earlier, the term $\frac{\partial u}{\partial \beta}$ is difficult to determine. It is actually almost analogous to finding the gradient of the cost function, which is what we are trying to do here. Therefore the terms including this term are grouped together:

$$\frac{dL}{d\beta} = \frac{\partial J}{\partial \beta} + \psi\frac{\partial \mathscr{R}}{\partial \beta} + \Big(\frac{\partial J}{\partial U} + \psi\frac{\partial \mathscr{R}}{\partial U}\Big)\frac{\partial U}{\partial \beta} \tag{8.15}$$

Now as the derivative $\frac{\partial U}{\partial \beta}$ is hard to determine, a value for ψ, the Lagrange variable, can be found so that $\Big(\frac{\partial J}{\partial U} + \psi\frac{\partial \mathscr{R}}{\partial U}\Big) = 0$. This means that this part of the equation drops out and $\frac{\partial U}{\partial \beta}$ does not have to be solved anymore.

We are thus then left with this system of equations:

$$\frac{\partial \mathscr{R}}{\partial U}^T\psi = -\frac{\partial J}{\partial U} \tag{8.16}$$

$$\frac{dL}{d\beta} = \frac{\partial J}{\partial \beta} + \psi\frac{\partial \mathscr{R}}{\partial \beta} \tag{8.17}$$

All the derivatives that can be written out by hand are simply computed. Following the adjoint method, the only additional cost of determining the gradients with respect to β is solving for the adjoint variable. This computation is very similar to a normal iteration in the CFD code as it the equations have similar terms as the starting governing equations. This however is a great improvement in terms of computational cost with respect to other methods of calculating the gradient.

The Lagrange variables ψ here is called the adjoint variables and will be further referred to as such.

8.2. Continuous Adjoint

In the previous section, the adjoint methodology has been shown and explained. A set of equations is formed from the derivation which needs to be solved to compute the adjoint variables, analogous to finding Lagrangian multipliers. To determine the gradient of the cost function with respect to the variable of interest then becomes easy and can simply be computed. However, the derivation is approached in a discrete sense. As the set of equations used to determine the adjoint variables are based on a RANS turbulence model, the resulting equations can be solved in a similar fashion. Therefore OpenFoam will be used as a platform for the implementation. OpenFoam works by defining the governing equation in continuous form. The equations will be discretised by the software in order to solve them numerically. Therefore the governing equations for the adjoint variables also have to be written in continuous form. The detailed derivation will be given in this section.

The methodology is equal to the discrete case, only now we define a fixed domain Ω in which we solve the equations. The equation becomes:

$$\frac{dL}{d\beta} = \frac{dJ}{d\beta} + \int_{\Omega} \psi \frac{dR}{d\beta} d\Omega \tag{8.18}$$

Where ψ and R are linear combinations of the adjoint variables with the corresponding basic governing equations.

$$\frac{dL}{d\beta} = \frac{\partial J}{\partial \beta} + \frac{\partial J}{\partial U}\frac{\partial U}{\partial \beta} + \int_{\Omega} \psi \frac{\partial R}{\partial \beta} d\Omega + \int_{\Omega} \psi \frac{\partial R}{\partial U}\frac{\partial U}{\partial \beta} d\Omega \tag{8.19}$$

After applying the same trick as before with discrete adjoint, setting the terms in front of the model variable derivative with respect to β to zero, we get the following system of equations:

$$\frac{dL}{d\beta} = \frac{\partial J}{\partial \beta} + \int_{\Omega} \psi \frac{\partial R}{\partial \beta} d\Omega \tag{8.20}$$

$$0 = \frac{\partial J}{\partial U} + \int_{\Omega} \psi \frac{\partial R}{\partial U} d\Omega \tag{8.21}$$

In this research, the $k-\omega$ RANS turbulence model will be used, in the equations given as R. Equation 8.21 will form the new governing equations, from now on referred to as the adjoint solver. As R denotes the $k-\omega$ RANS turbulence model, it now becomes clear that the terms in the adjoint solver will be very similar. The equations will form a linear combination of the adjoint variables multiplied by the derivative of the RANS turbulence model equations with respect to the flow variables U; in this research consisting of u, p, k and ω. The detailed derivation of these equations is shown in appendix A.

The resulting equation for the gradient is shown in equation 8.22. Here it becomes clear that the gradient of the cost function is dependent on the production term. Therefore, in order to get sufficiently smooth gradients, a smooth mesh is necessary, as at mesh refinement transitions the velocity gradients can be quite high. When sharp refinement jumps are present in the mesh, discontinuities can form in the resulting cost function gradients.

$$\frac{dL}{d\beta} = \frac{1}{\sigma_{\beta}^2}(\beta - \beta_0) - \int_{\Omega} \omega_a \gamma P d\Omega \tag{8.22}$$

In equation 8.23 till 8.26 the governing equations of the adjoint solver are shown. As they are derived from the RANS k-omega model equations they have a similar form. Note that only the cost function derivative, the first term of equation 8.21, is present in the adjoint momentum equation. This is the case because the cost function only has a velocity term in it, and lacks the other flow variables.

Something interesting that can be noted is that when comparing the adjoint equation to the original turbulence model equations some of the terms appear to have switched signs. To be more precise; the term with an odd order spatial derivative have switched signs while the terms of an even order spatial derivative kept the same sign. Later it will be shown that this causes the adjoint variables to flow upstream with respect to the primal variables.

$$R_i^{u_a} = \frac{\partial J}{\partial u_i} - \frac{\partial (u_{a,i} u_j)}{\partial x_j} - \frac{\partial u_{a,i}}{\partial x_j} u_i - \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial u_{a,i}}{\partial x_j} + \frac{\partial u_{a,j}}{\partial x_i}\right)\right) + \frac{\partial p_a}{\partial x_j} - k\frac{\partial k_a}{\partial x_j} - \omega\frac{\partial \omega_a}{\partial x_j} - \frac{2}{3}\frac{\partial (k_a k)}{\partial x_j} - \frac{2}{3}\gamma\frac{\partial (\omega_a \omega)}{\partial x_j} + 2\frac{\partial}{\partial x_j}\left((k_a \nu_t + \omega_a \gamma \beta)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) \tag{8.23}$$

$$R^{p_a} = -\frac{\partial u_{a,i}}{\partial x_i} = 0 \tag{8.24}$$

$$R^{k_a} = -\frac{\partial (k_a u_j)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_k \nu_t + \nu)\frac{\partial k_a}{\partial x_j}\right) + \frac{\partial k_a}{\partial x_j}\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} + \frac{\partial \omega_a}{\partial x_j}\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j} + \frac{1}{\omega}\frac{\partial u_{a,i}}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{k_a}{\omega}P + k_a\frac{5}{3}\frac{\partial u_j}{\partial x_j} + k_a C_\mu \omega = 0 \tag{8.25}$$

$$R^{\omega_a} = -\frac{\partial (\omega_a u_j)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega_a}{\partial x_j}\right) - \frac{\partial \omega_a}{\partial x_j}\sigma_\omega\frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} - \frac{\partial k_a}{\partial x_j}\sigma_k\frac{k}{\omega^2}\frac{\partial k}{\partial x_j} - \frac{k}{\omega^2}\frac{\partial u_{a,i}}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + k_a\frac{k}{\omega^2}P + \left(1 + \gamma\frac{2}{3}\right)\omega_a\frac{\partial u_j}{\partial x_j} + 2\omega_a\alpha\omega + C_\mu k_a k = 0 \tag{8.26}$$

8.3. Data Mapping in the Cost Function

As discussed in the previous section the continuous adjoint formulation is used during this research. In this section, the cost function will be defined. A problem that arises is that the reference data is a discrete data set. As well, the discrete data points will most probably not coincide with the grid of the simulations. Next to that, the experimental data does not cover the full CFD domain. How the different data is handled in OpenFoam will be discussed in this section.

8.3.1. New Cost Function

The cost function in its continuous form, as defined in 7.3, can be written as:

$$J_\Omega = \int_\Omega \frac{1}{2C_m}\Big(h(\beta) - d\Big)^2 d\Omega + \int_\Omega \frac{1}{2C_\beta}\Big(\beta - \beta_0\Big)^2 d\Omega \tag{8.27}$$

Where $h(\beta)$ is the simulation code corrected by β which outputs the velocity field. d denotes the experimental velocity field data. The squared difference between the velocity field from the simulation and the reference data is integrated over the domain. The second term is the regularisation term, originating from Bayes theorem. Here C_m is the observational co-variance matrix and C_β the prior co-variance matrix.

The experimental data, measured by means of PIV, consists of a 2D mesh with a certain velocity known value per square. The mesh, however, is very different compared to the mesh used in the simulations. To be able to compare the two flow fields the values at the cell centres are used. While in OpenFoam the equations are written in their continuous form, the solver still uses a numerical approach to discretise the equations. As OpenFoam is a finite volume method solver, the values at the cell centres will always be available. Therefore the terms involving the cost function J are the only terms written in their discrete form. This forms no problem as no numerical approaches are necessary for these terms.

As the data points are only known in discrete places the cost function is rewritten as:

$$J_\Omega = \int_\Omega \frac{1}{2C_m}\delta(x - x_i)\Big(h(\beta)_i - d_i\Big)^2 d\Omega + \int_\Omega \frac{1}{2C_\beta}\delta(x - x_m)\Big(\beta_m - \beta_{0,m}\Big)^2 d\Omega \tag{8.28}$$

Which can naturally be written in its discrete form as:

$$J_\Omega = \sum_{i=0}^{N} \frac{1}{2C_m}\Big(h(\beta)_i - d_i\Big)^2 + \sum_{m=0}^{M} \frac{1}{2C_\beta}\Big(\beta_m - \beta_{0,m}\Big)^2 \tag{8.29}$$

The Dirac delta is used in the continuous form in order to only use the data on the specific location while integrating. When transitioning to the discrete form the Dirac delta drops out and no dependency on the mesh volume appears.

As discussed, in this case, the reference data d and the simulation results h can not be subtracted that easily as they live on different grids. In this research it is chosen to map the CFD field h_j to the grid of the PIV data by means of a transformation matrix $H_{i,j}$. In this direction, the true experimental data field is maintained because no interpolation of the data is needed.

The cost function should thus be written as:

$$J_\Omega = \sum_{i=0}^{N} \frac{1}{2C_{m,i}}\Big(H_{i,j}h(\beta)_j - d_i\Big)^2 + \sum_{m=0}^{M} \frac{1}{2C_{\beta,m}}\Big(\beta_m - \beta_{0,m}\Big)^2 \tag{8.30}$$

Where the transformation matrix $H_{i,j}$ is a mapping from the CFD grid to the PIV grid. Note that the second term in the equation only lives on the CFD grid as it only a regularisation term.

In the adjoint equations the cost function is used in a differential with respect to the flow variables. These can be now simply written as:

$$\frac{\partial J_\Omega}{\partial u_k} = \sum_{i=0}^{N} \frac{\partial}{\partial u_k}\frac{1}{2C_{m,i}}\Big(H_{i,j}h(\beta)_j - d_i\Big)^2 \tag{8.31}$$

$$\frac{\partial J_\Omega}{\partial u_k} = \frac{1}{C_{m,i}}\Big(H_{i,k}h(\beta)_k - d_i\Big)H_{i,k} \tag{8.32}$$

and

$$\frac{\partial J_\Omega}{\partial \beta_n} = \sum_{m=0}^{M} \frac{\partial}{\partial \beta_n}\frac{1}{2C_{\beta,m}}\Big(\beta_m - \beta_{0,m}\Big)^2 \tag{8.33}$$

$$\frac{\partial J_\Omega}{\partial \beta_n} = \frac{1}{C_{\beta,n}}\Big(\beta_n - \beta_{0,n}\Big) \tag{8.34}$$

Where the sum sign naturally disappears due to the extra transformation matrix which appears due to differentiation. But an other reason for the sum to disappear is that the all the terms are zero except when $j = k$ and $m = n$.

In OpenFoam a finite volume method is used. This means that the PDE's are written and solved in their integral form. The same has been done for the cost function terms. To include the cost function in this volume integral we can simply divide by the cell volume, as the terms are not volume dependent. Therefore in OpenFoam, the cost function derivative is included in the volume integral:

$$0 = \int_\Omega \Big(\frac{1}{V_\Omega}\frac{\partial J_\Omega}{\partial u} + \psi\frac{\partial R}{\partial u}\Big)d\Omega \tag{8.35}$$

$$\frac{\partial J}{\partial u_k} = \frac{1}{V_\Omega}\frac{1}{2C_{m,i}}\Big(H_{i,k}h(\beta)_k - d_i\Big)H_{i,k} \tag{8.36}$$

Equation 8.36 is thus the term appearing in the adjoint momentum equation where it acts as a source term. For this reason, this term will from on be further referenced as the source term in the adjoint momentum equation. Something interesting that can be deduced from equation 8.36 is that this source term is completely linear. Therefore the effect of the source term from all the data points combined is a linear combination of the source terms of single points of data. Furthermore, as all the equations in the solver are linear, the computed gradient of the cost function with respect to the corrective field, is also a linear combination of all the single reference data points. In other words, when determining this sensitivity for all the reference data points separately and adding them together afterward will give the same result as using the complete reference data set and calculating the sensitivity in one go.

8.3.2. Data Mapping

What might be odd is that the transformation matrix is applied to the velocity field of the simulation. By doing this the source term $\partial J/\partial u_k$ in the adjoint momentum equation becomes more complex. When the transformation is simply applied to the experimental velocity field the equation remains much simpler. However, the experimental data is regarded as the ground truth. The goal is to improve the simulation to get the simulation velocity field closer to the experimental data. For this reason, it makes sense to transform the simulation data to the grid used by the experiments. This way the experimental data remains unaltered.

An added benefit of this method is that when there is no reference data available in a certain portion of the domain, the contribution to the cost function becomes zero. The transformation matrix interpolates the simulation grid to the grid of the experimental data. However when no data points are present in the vicinity of a cell the values for the transformation remain zero. The transformation matrix thus makes sure only the section where reference data is available is used for the cost function.

To get the transformation matrix, an implementation of an interpolation algorithm is used in OpenFoam. This algorithm uses both the values at the cell centres and the values at the cell vertices. Therefore, the cell centre values have to be interpolated by OpenFoam to get the cell vertex values as well.

First, the algorithm searches for the cell in the simulation mesh where the experimental data point is located in. This cell is divided into prisms using the cell centre point and the vertices of the cell. The prisms all have one of their vertices at the cell centre point. A hexahedron will thus be divided into twelve prisms. Then the prism which encapsulates the experimental data point is chosen. The barycentric weights of the prism vertices to the PIV data point are calculated for the interpolation. The weights times their value at the respective prism vertex will be summed together to get the interpolated simulation value at the experimental data point.

Because the OpenFoam implementation uses the values at mesh vertices as well, equation 8.32 will be split in a cell centre and a cell vertex variant. Afterward, when the source terms are calculated, the cell vertex variant will be interpolated to cell centres and added to the cell centres source terms.

8.4. Boundary Conditions

From the derivation of the adjoint solver, next to volume integrals, also boundary integrals followed. These have to be set to zero while solving for the adjoint variables. Because of this new boundary conditions should be formed of which the derivation is shown in appendix A.3.

For the inlet and on walls, where Dirichlet boundary conditions are applied on the primal velocity and turbulence variables, zero Dirichlet boundary conditions have to be applied to their adjoint counterparts. Where zero Neumann boundary conditions are applied to the primal pressure, also zero Neumann boundary conditions have to be applied to the adjoint pressure.

However for the outlet and the slip wall the boundary conditions become a lot more complex. The equations which have to be satisfied at the outlet are shown below:

$$p_a = u_{a,n}u_n + u_{a,j}u_j + 2(\nu+\nu_t)n_j\frac{\partial u_{a,n}}{\partial x_j} + \frac{5}{3}k_a k + \left(1+\frac{2}{3}\gamma\right)\omega_a\omega \tag{8.37}$$

$$u_{a,t} = -\frac{1}{u_{a_n}}\left[(\nu+\nu_t)\left(n_j\frac{\partial u_{a,t}}{\partial x_j} + t_i\frac{\partial u_{a,n}}{\partial x_i}\right) - 2(k_a\nu_t + 2\omega_a\gamma\beta)t_i\left(\frac{\partial u_n}{\partial x_i}\right)\right] \tag{8.38}$$

$$k_a = \frac{1}{u_{a_n}}\left[u_{a,i}\frac{1}{\omega}\frac{\partial u_n}{\partial x_i} - \frac{\partial k_a}{\partial x_j}(\sigma_k\nu_t+\nu)n_j\right] \tag{8.39}$$

$$\omega_a = -\frac{1}{u_{a_n}}\left[\frac{\partial \omega_a}{\partial x_j}(\sigma_\omega\nu_t+\nu)n_j + u_{a,i}\frac{k}{\omega^2}\frac{\partial u_n}{\partial x_i}\right] \tag{8.40}$$

At the slip wall the boundary conditions for the adjoint velocity is zero Dirichlet and for adjoint turbulence values zero Neumann. The boundary condition for the adjoint pressure is equal to the outlet boundary condition.

In practice equation 8.37 to 8.40 are all zero at the outlet. The variables are only driven by the source term as the adjoint fields will be initialised as zero fields. Also at the boundaries, there are no driving conditions as all are zero Dirichlet or Neumann. Also, as earlier noted, all the odd order spatial derivatives in the adjoint equations have switched signs, causing a reversed flow direction. If the outlet is thus sufficiently far away from the source terms, or in practice, far enough downstream from the reference data, the effect will be zero thus the internal field at the outlet has no direct driver.

If we look at equation 8.38 we see that there is a dependency on the adjoint velocity gradient and on the adjoint turbulence values. The boundary conditions at the wall for the adjoint velocity are zero Dirichlet. So only if the adjoint turbulence values take a non-zero value this equation will be non-zero. However, if the adjoint turbulence values remain zero, equation 8.38 could be simply reduced to a zero Dirichlet boundary condition.

Equations 8.39 and 8.40 are very similar and when looking at them a dependency on the adjoint velocity can be seen. The other term is a dependency on its own gradient. Therefore, if the adjoint velocity does not take on a value at the boundary, the adjoint turbulence variables will also be just zero at the outlet boundary.

Again we can note that the field is initialised as zero fields and the boundary conditions on the neighbouring boundaries are zero Dirichlet or Neumann. Therefore it can be reasoned that the boundary conditions for the adjoint velocity and the adjoint turbulence values at the outlet could be replaced by a zero Dirichlet boundary condition.

When looking at equation 8.37 it simply follows this could also be a zero Dirichlet boundary condition. Again, only with the assumption, the reference data is sufficiently away from the outlet boundary.

9

PIV Experiment and Data

In this chapter, the experiments performed by Deltares are described. The test setup used will be explained and also the possible problems of the test setup will be discussed. Next, the data acquisition will be described and finally, the processing of this data will be discussed.

9.1. Experimental Set Up

As mentioned before, particle image velocimetry data will be used as reference data. For this research, Delatares has performed a PIV experiment in their experimental facilities. A scale model was used to recreate the flow features needed. In figure 9.1 a schematic layout of the experiment set up is shown.

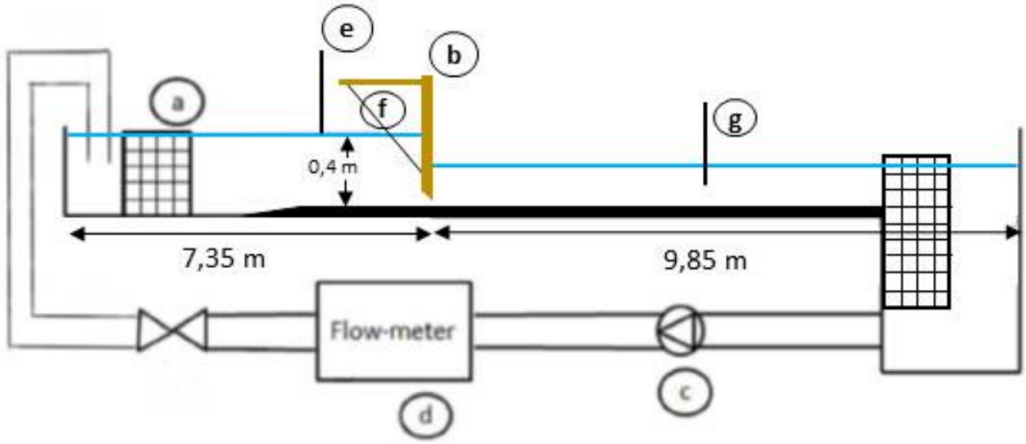


Figure 9.1: Schematic layout of the experimental set up. a) Flow straightening device. b) Underflow gate. c) Pump. d) Flow meter. e) Level staff. f) Wires. g) Wave gauge. [1]

The water, which is being pumped around in a closed system, enters the channel vertically to reduce its effect. Right after it passes through a flow straightening device. The channel after is fairly long in order to get a uniform stream. At 6.5 m downstream of the flow straightening device the flow hits a ramp that raises the floor level. After the slope the gate is positioned, 1.32 m downstream. The gate was adjustable in height so that multiple gate openings could be tested. The tip of the gate has a rounded shape so that the circle is flush with the upstream side and intersects the downstream side at an angle of 70 degrees. A level staff, located at around 1.18 m upstream, is used to set the water level at 0.4 m. The water flows downstream at around 12 m into a basin from where it is pumped to the front. A flow meter is placed behind the pump to measure the discharge. The sides of the channel are all made of acryllic glass. A closeup photo of the experimental setup is shown in figure 9.2.

Figure 9.2: Close up photo of the underflow gate in the channel. [1]

9.2. Caveats of the Experiment

As with every experiment, it is important to understand the differences with the actual problem you are trying to replicate. This is especially important for this research as the data is going to be used as reference data with the goal of improving a turbulence model.

The standard problem is that the geometry is not exactly the same. It is very difficult to get the exact same geometry in the simulation as in the experiment. The weir is replicated as exact as possible but there will always be some production errors in reality. Also, due to simulation constraints, not the whole experimental setup is replicated in CFD. This would simply be very inefficient. The differences however are estimated small, and it is probable other aspects have a far bigger impact. Another difference between the current simulation setup and the experimental setup is the 3D effects. In the scope of this research, only a semi 2D simulation has been used. As discussed in chapter 4.3 3D effects will have an effect on the actual shape of the jet. In the experiment, obviously, a free surface was present between the water and the air. As discussed in chapter 10.2, to simplify the problem only the water will be simulated in the CFD simulations. One of the biggest problems however was the water pump. Its discharge was not constant during the experiment as it showed an oscillatory behaviour. This adds extra turbulence to the flow as well as causes some extra oscillations to the jet. Next to this, the turbulence values could not be accurately measured at the inlet. Therefore setting the turbulence intensity for the CFD simulations has been a guess.

There are quite a few unknowns which make it difficult to properly compare the experimental setup to the simulation setup. However, the effects are deemed small enough to still get a use full and meaning full correction field in the end. The jet is insensitive enough to the minor differences in geometry. Also, the 3D effects are small, and in possible future research will be taken into account. The discharge oscillation is a problem, however, this is a part of a bigger problem of comparing experimental data to a CFD simulation that uses a RANS model, which will be discussed later. The problem of not accurately knowing the turbulence values of the flow upstream of the weir can possibly be corrected by the field inversion paradigm itself.

9.3. Data Acquisition

To get a proper detailed flow field, four capturing frames are setup. One frame is set up in front of the weir and three behind the weir. A fifth, smaller, plane is placed just after the gate opening to capture the flow in detail. A schematic sketch of the five planes is shown in figure 9.3.

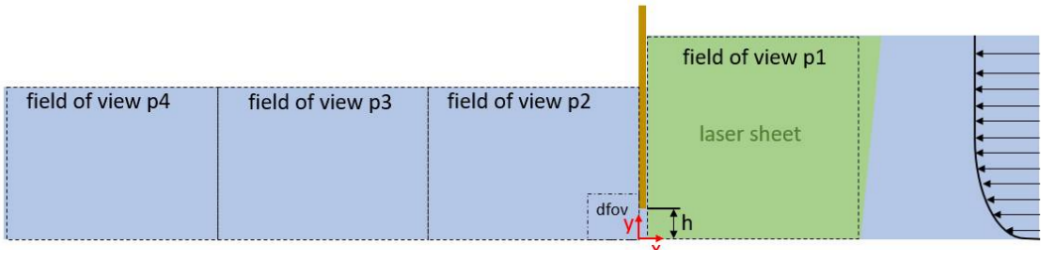


Figure 9.3: Sketch of the five different PIV field of view planes. [1]

As lasers are being used to light up the particles in the flow, reflections are a problem. Therefore close to walls the data from a PIV experiment is generally unreliable. Boundary layer data is thus not available. While this is unfortunate, the interesting area is the shear layer of the jet which will be fully visible.

In a PIV experiment, the flow field is captured multiple times during a set amount of time. Afterward, these snapshots of the flow field are time-averaged to get an image of the mean flow field. In the research of Deltares, unsteady simulations, URANS and LES, have been used. As comparing two instantaneous flow fields with each other is of no interest, the time-averaged flow fields are used for comparison.

A problem however is that during this research a RANS model has been used. This model simulates the mean flow and is thus inherently steady. There is a difference between a time-averaged unsteady flow field and a simulated mean flow. This is the problem with the RANS approach. This is no problem, though, as this difference is sought to be overcome.

As the time average is taken from the experimental data, also the variance can be calculated. This can be used in the field inversion process as discussed in chapter 7.3. The mean and variance are calculated using equations 9.1 and 7.8. The variance is analogous to the Reynolds stresses in the flow field. However the variance can also be influenced by measurement induced errors and represents thus not only the velocity fluctuations due to a turbulent flow.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{9.1}$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2 \tag{9.2}$$

The variance of the data will be used in the adjoint solver as to how well the data point is trusted. For experimental data this can be quite important. Due to measurement errors, sharp jumps can occur in the resulting velocity field. These discontinuities can form problems with the stability of the adjoint solver. The adjoint solver tries to find a direction the corrective field should be changed in to match the experimental velocity field. When the experimental velocity field is not physical this will cause problems. Therefore dampening the effect of these erroneous data points increases the stability and performance of the paradigm.

10

Flow Cases

During the thesis, several flow cases were used to test and improve upon the paradigm and adjoint solver. In previous work, [18][31] principal aerodynamic flow cases were used with readily available high fidelity simulation data. These flow cases are suitable for testing as they generally are off low Reynolds number and simplistic in terms of boundary conditions. Also, these flow cases are well understood and the setups and meshes are carefully crafted and tested.

Therefore in order to further extend the paradigm these flow cases have been used as a test platform. After the development phase, the actual test flow case was set up. The simulations by Deltares [1] have been performed in Star CCM+. This paradigm is however developed in OpenFoam. Also, the flow case had to be simplified, as, for example, the implementation of field inversion in OpenFoam can not handle complex boundary conditions yet.

In this chapter, the flow cases used for developing and testing the paradigm will be discussed. First, the principal aerodynamic flow cases will be discussed and at last, the underflow weir case in OpenFoam will be explained.

10.1. Principal Aerodynamic Flow Cases

In order to order to develop the paradigm in an efficient manner, principal aerodynamic flow cases will be used. These flow cases have been well studied so high fidelity data is readily available for the inversion process. However, the main advantage of these cases over the underflow weir case is that they are simple and fast to solve. The choice has been made to select one of the flow cases also used in previous research [31] so to be able to compare and verify the results. The same settings and case setup will be used. These cases generally have a low Reynolds number and a relatively low cell count for RANS simulation setups, resulting in it being computationally light. Also, the boundary conditions used are principal and simple.

The chosen flow case is the periodic hill. This is one of the principal aerodynamic flow problems as it is simple and includes some key phenomena in aerodynamics. First flow separation occurs just after the crest of the hill on a curved surface. After the flow re-attaches on the flat section between the hills. These flow features have been well studied as they are often relevant in practical flow cases in engineering. For the inversion phase, high fidelity data is used from the research from Breuer et al. [2]. In their research, they performed a direct numerical simulation (DNS) for multiple Reynolds numbers. For this research, the case with a Reynolds number of 5600 is chosen.

The reason why this flow case has been chosen as the first testbed is because of the simple boundary conditions. The upper and lower walls are treated as physical walls. Because of the relatively fine mesh and low Reynolds number the y^+ values are small enough so that extra wall functions are not necessary. Cyclic boundary conditions are imposed on the inlet and outlet. A momentum source at the inlet is used to drive the flow. A careful mesh convergence study has been performed in earlier research [31]. The mesh settings have been copied in order to be able to properly compare the results.

10.2. Underflow Weir

The previously described cases have been used to test the paradigm and built on its capability step by step. However, the actual important test case is going to be the underflow weir. Deltares has performed PIV experi-

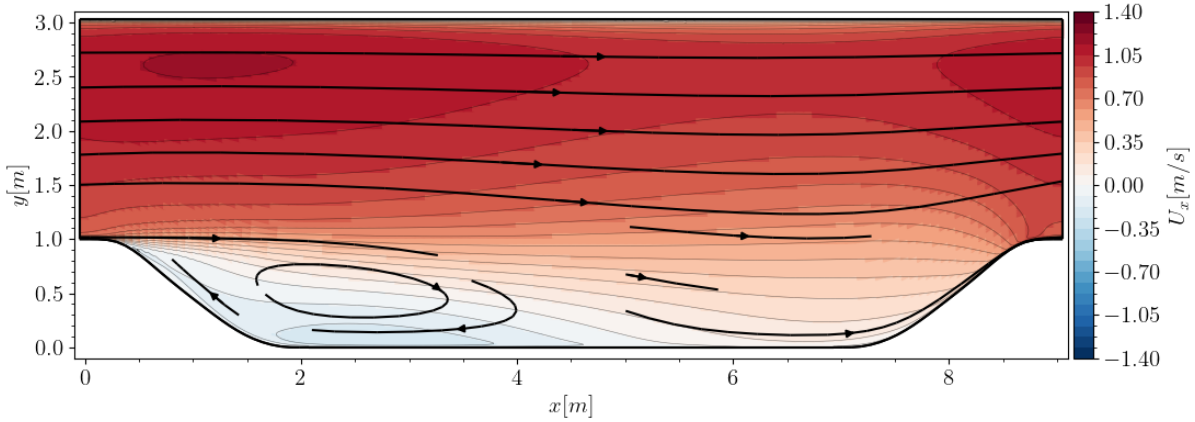


Figure 10.1: Periodic Hill flow field with a Reynolds number of 5600, computed by using a RANS k-ω model.

ments and CFD simulations of different levels of fidelity [1]. They also tested for different weir gate openings, namely an opening of 2cm, 6cm, and 15cm. Therefore a lot of data is available. This case is thus ideal to test if the paradigm is capable of correcting the turbulence model and consequently improving the output flow field.

For this research, the gate openings of 2cm and 6cm are going to be used to test the paradigm. For the 6cm gate opening case the most reference data was present while for the case with the 2cm gate opening the steady simulations were most stable.

PIV data will be used as reference data to improve the standard k-omega model to better present the reality. For this reason, the simulations should represent the experiment as close as possible. The choices made in the simulation setup will be discussed below. First, the domain and the boundary conditions will be discussed. After that, the mesh will be explained. Finally, the turbulence values, inlet, and initial conditions will be discussed.

10.2.1. Domain and Boundary Conditions

The case resembles a river that encounters an underflow weir. Weirs are generally used to maintain the water levels upstream, so there will be a water level difference between upstream and downstream of the weir. In order to simulate physics as close to the truth as possible, Deltares used a multi-flow simulation, simulating the air and the water at the same time separated by a free surface. In figure 10.2 a schematic overview of the boundary conditions used is shown. The water is pushed through the weir because of the difference in hydrostatic pressure caused by the water level difference between upstream and downstream of the weir.

The problem, however, with these chosen boundary conditions and using the multi-phase flow is that the case becomes rather complex. Extending the adjoint solver and the paradigm of field inversion to be capable of handling such a case might be the next step but is out of the scope of this research. Therefore the case had to be simplified to work within the capabilities. Another problem was that the case was set up in the CFD solver Star CCM+. This CFD solver, being a commercial tool, does not yet allow the features needed to implement a new solver. Therefore the CFD solver OpenFoam was chosen and thus the case had to be set up in that environment.

The domain is chosen such that only the water is simulated. Therefore the height of the domain is chosen to be the height of the water level. Close to the gate the water level is changing slightly however this is not included in the OpenFoam case setup. The domain height is kept constant upstream and constant downstream of the weir, where upstream the domain has a greater height than downstream. When checking the actual height difference in the experiments and in the simulation performed by Deltares [1] this shows to be a reasonable assumption as the Froude number and thus the height difference is low.

The length of the upstream and downstream section was chosen to be equal to the simulations of Deltares [1] in order to be able to use their simulation as reference and baseline.

To further reduce the complexity of the simulation case setup the decision has been made to go for a semi 2D setup. The width of the domain is set to the base cell size so that the number of cells in this direction equals one. In zones where a refinement is used the number of cells will be increased.

As in the OpenFoam setup, only the water is simulated and the air is left out, the choice of boundary

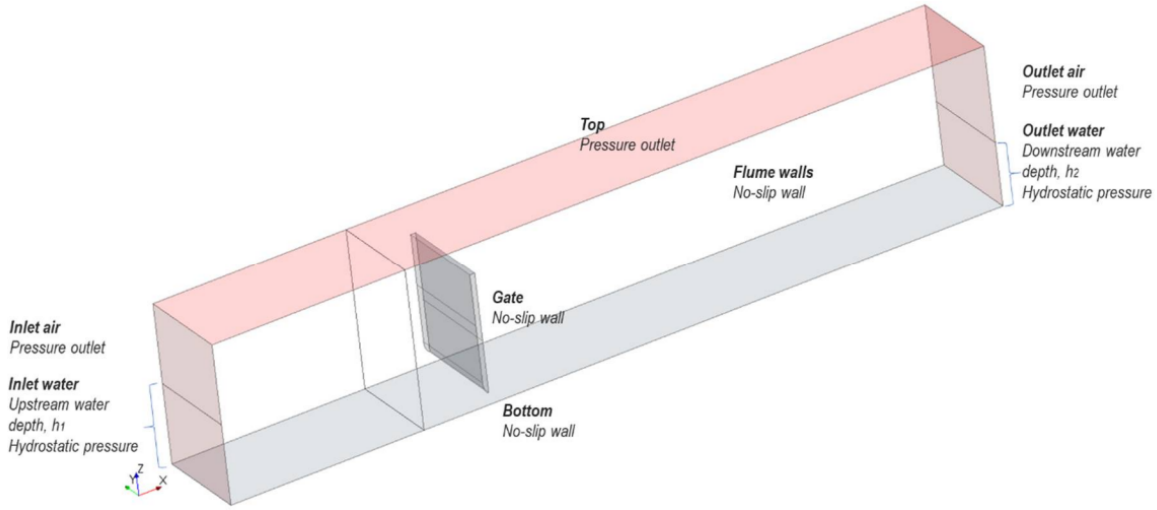


Figure 10.2: Schematic representation of the domain boundary conditions used by Deltares [1].

conditions on the top boundaries will be different compared to the one used by Deltares. In this research, the top boundaries have a slip boundary condition. It is reasonable to assume this as in reality the water will not feel any wall resistance from the free surface. One problem is the pressure in this case. However from the simulations performed by Deltares [1] the Froude number is concluded to be very small and thus few surface disturbances will be present. Therefore no real problems are to be expected using the slip boundary condition because the flow field should be calm in this region.

The bottom and the weir boundaries have a normal no-slip wall condition set to them. In order to make the case semi 2D the front and back boundary have a symmetry boundary condition.

Another difference in the water-only simulation is that hydrostatic pressure can not be used to drive the flow. For this reason, a velocity inlet is chosen at the start of the domain. The outlet is set to a pressure outlet. A problem that occurs with using a constant velocity inlet is that for the cell at the corner of the inlet and bottom boundary impossible boundary conditions are imposed. At the inlet, a boundary of the cell a constant velocity is imposed. However, at the bottom boundary, the velocity should be zero due to the no-slip boundary condition. A normal CFD simulation will be capable of handling this though and normally no problems will occur, although better solutions are available. The problem is that still, some artifacts will be visible in the flow solution due to these unnatural boundary conditions. High-velocity gradients will be present at the corner in order to cope with the set constraints. As the resulting flow field of the base simulation will be used by the adjoint solver, the high-velocity gradients present at the corner will cause instabilities, and using a simple constant velocity inlet is thus not possible.

To get around this problem a non-constant velocity field is used as a constraint at the inlet which includes a boundary layer. To get this velocity field at the inlet the domain is extended forward. The start of the domain will have a slip boundary condition on the bottom. After that, the no-slip boundary condition is applied to form a boundary layer. This way the flow has some time to form to both constraints. The domain is extended enough to form an established boundary layer at the initial inlet. A mapping of the flow field at this plane is used as an inlet boundary condition for the rest of the simulations.

A schematic overview of the domain with a 6cm gate opening is shown in figure 10.3. For the case with the 2cm gate opening the hydraulic jump is slightly higher. Therefore the downstream domain height is decreased from 0.32m to 0.25m, which is in agreement with the PIV experiments and the multi-flow simulation performed by Deltares [1].

10.2.2. Mesh

A few considerations had to be made for the mesh. First of all, again to simplify the simulation, no wall conditions can be applied to the no-slip boundaries. The main reason this simplification is needed is that the adjoint solver is not capable of using complex wall models. Research has been done in implementing wall

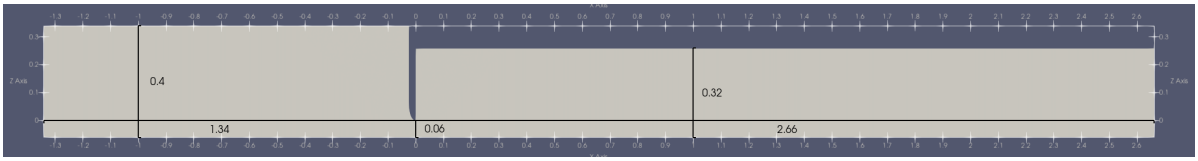


Figure 10.3: Schematic representation of the underflow weir domain in OpenFoam.

functions for the standard RANS turbulence models [40], however, this is out of the scope of this research.

A second consideration is that the mesh needs to be as smooth as possible. A not smooth mesh or jumps in refinement over a small number of cells will cause high gradients in velocity. This is not directly a problem for a normal CFD solver however it can be for the adjoint solver as with high gradients it can become unstable.

The mesh has been created using OpenFoam. Initially, the blockMesh utility has been used. With this mesher, a smooth background mesh has been created. The base cell size is 0.01m. Towards the bottom wall and the location of the weir, the mesh is refined in a smooth manner. Around the tip of the weir, the cell size is 0.002m. After this, the snappyHexMesh utility from OpenFoam has been used to place the weir gate and adapt the mesh accordingly. Also, the prism layers around the weir have been created using this utility. Because of the weir gate, there are some jumps in the mesh cell size, however, these are still small. In figure 10.4 the cell volumes are shown in the simulation domain. Here the smooth grading towards the refined areas can be seen. In figure 10.5 the cell volumes around the weir gate opening are shown. Due to the added weir geometry, the mesh is not perfectly smooth around it.

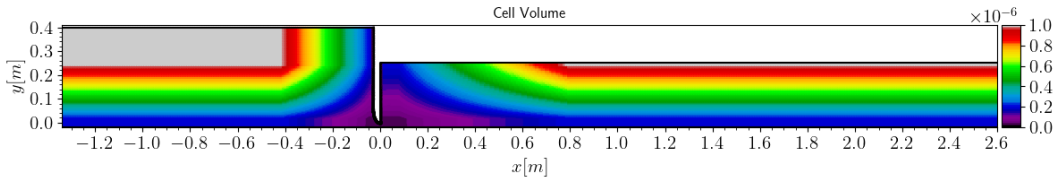


Figure 10.4: Cell volume in the domain.

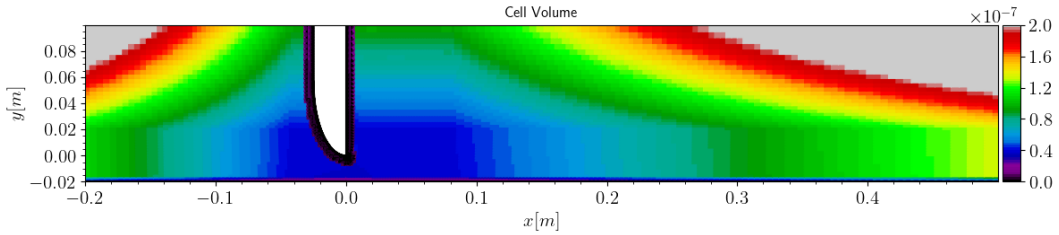


Figure 10.5: Cell volumes around the weir gate opening.

In figure 10.6 the mesh is shown in the area of the gate opening. The prism layers around the weir and at the bottom wall are clearly visible. Also, some refinement is visible around the weirs tip. Instead of a smooth transition, the cells are divided into eight. At the very tip, another refinement is applied. While the whole mesh has only one cell in the span-wise direction, due to these refinements there are locally two or four cells in the span-wise direction.

To test the capabilities of the adjoint solver a different mesh is generated. While for the first mesh the focus was on making it as smooth as possible, this is not always possible in more complex cases. Therefore, instead of a smooth transition in refinement, refinement zones are used. Around the weir gate opening several refinement zones have been applied, each going down just one step in refinement. In figure 10.7 the cell volumes are shown for the mesh. As shown there are jumps in the cell volume, each of a factor of eight. This type of mesh is more commonly used for more complex geometries.

10.2.3. Turbulence Parameters and Inlet Conditions

The turbulence parameters are derived from the simulation setup by Deltares [1]. The dynamic viscosity and density of water at the temperature of the water during the experiments are used to calculate the kinematic

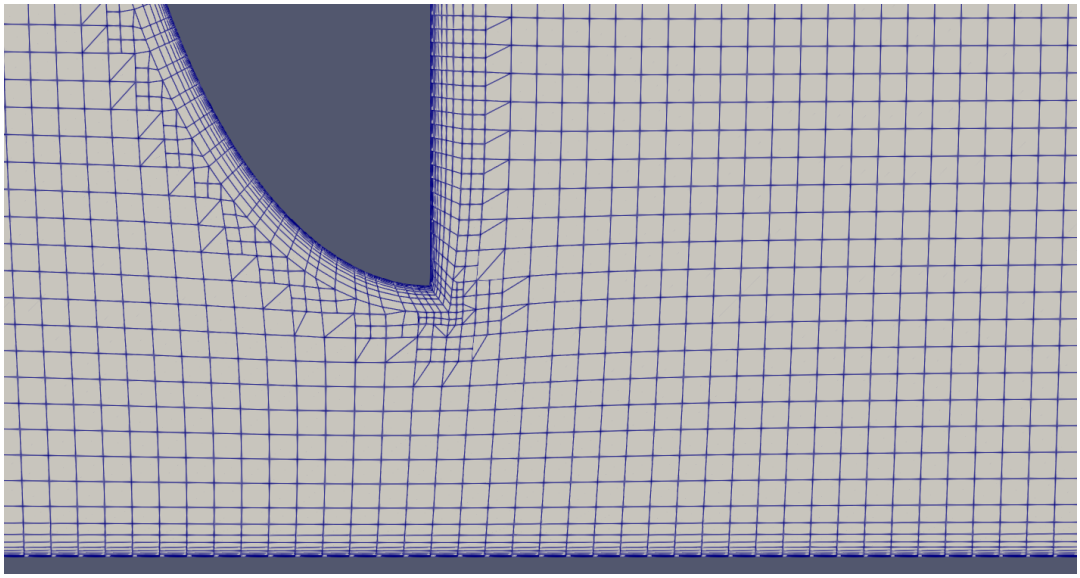


Figure 10.6: Snap shot of the mesh around the weir gate opening.

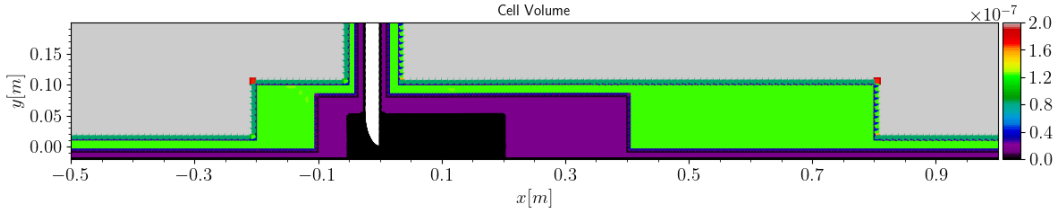


Figure 10.7: Cell volumes for the non uniform mesh with refinement zones.

viscosity needed by OpenFoam.

Furthermore, the inlet turbulence values are also derived from the simulations of Deltares. Star uses a different set of input variables to OpenFoam so they have to be calculated. The turbulence intensity, I, is set to 1% and the turbulent viscosity ratio, μ_t/μ to 10. In OpenFoam a freestream velocity, U, of 0.1572 m/s is used for the case with the 6cm gate opening and 0.0698 m/s is used for the 2cm gate opening. For setting the velocities at the inlet in the simulations the mass flow is measured just after the pump is used. This is then validated by comparing the flow velocities upstream of the gate. Using the velocity at the gate opening for comparison should be avoided as this can be heavily influenced by the physics around the weir's walls and the flow field downstream.

OpenFoam needs a turbulent kinetic energy, k, and specific turbulent dissipation rate, ω, as input. To calculate these variables the following equations are used:

$$k = \frac{3}{2}(UI)^2 \tag{10.1}$$

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{10.2}$$

$$\nu = \frac{\mu}{\rho} \tag{10.3}$$

$$\epsilon = \frac{C_\mu k^2}{\nu \frac{\mu_t}{\mu}} \tag{10.4}$$

$$\omega = \frac{\epsilon}{C_\mu k} \tag{10.5}$$

Where the constant, C_μ, is a model constant used in the RANS turbulence models and has a value of 0.09. The resulting values are shown in table 10.1.

Variable	Value 6cm	Value 2cm	Unit
ν	8.9088E-7	8.9088E-7	m/s^2
k	3.7068E-6	1.8283E-5	m^2/s^2
ω	0.41608	2.0523	$1/s$

Table 10.1: Turbulence parameters used for the two underflow weir cases in OpenFoam.

10.2.4. Stability of cases

As mentioned before, the gate opening of 2cm was chosen because it displayed less unsteadiness in the experiments and simulations performed by Deltares. Deltares used several models of different fidelity, however, all of them were transient. In this research, a steady model is used which could cause issues if the flow is too unsteady. RANS is inherently steady and thus solves for the mean flow field. If the actual flow field is completely transient a RANS turbulence model will not be capable of finding a solution.

This problem was observed with the 6cm gate opening case. With the chosen kinematic viscosity, which influences the Reynolds number, the RANS turbulence model could not find a well-converged solution. Behind the weir opening the flow field was fluctuating and showed a clear transient behaviour. This resulted in the residuals staying rather high and showing an oscillating behaviour, as shown in figure 10.8.

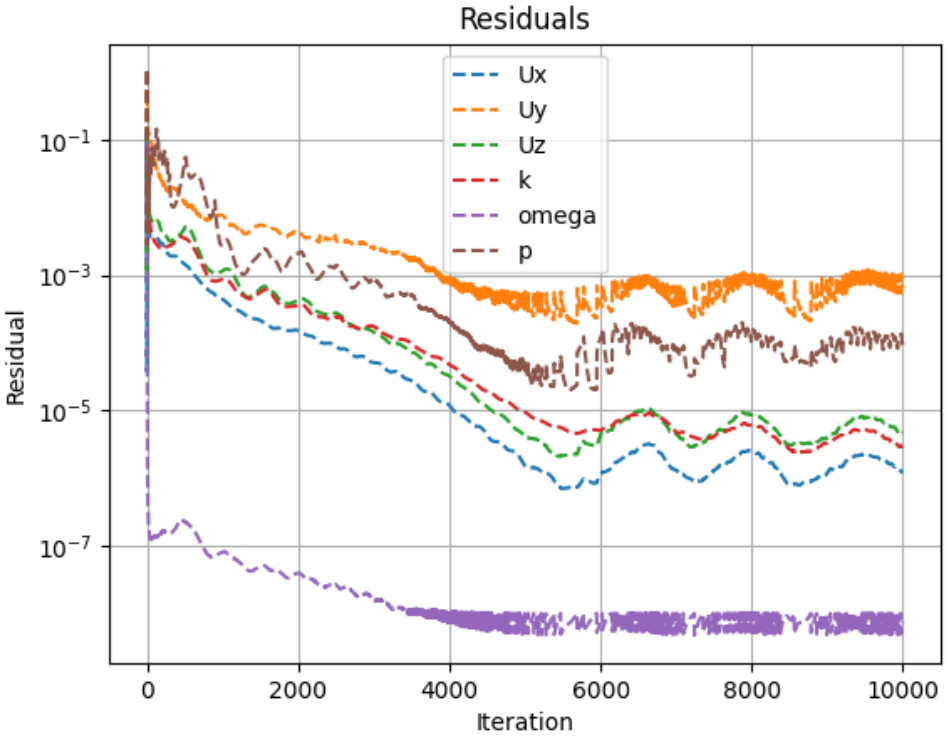


Figure 10.8: The residuals of the solver plotted over the iterations for the 6cm gate opening case.

These residuals are still too high for using the solution as an input for the adjoint solver. The adjoint solver is less stable and therefore a higher chance of divergence will be present.

To still be able to use the 6cm case a sensitivity analysis was performed on the several input variables. The kinematic viscosity was found to have a high sensitivity to the resulting flow field and stability of the case. Initially, a value was chosen that matches the value used by Deltares [1]. However after increasing the value by about 20% to 1.1385E-6 m/s^2, which resembles a temperature decrease of 10°C, from 25°C to 15°C, a

steady solution was found. Increasing the kinematic viscosity has a stabilising effect on the flow as there will be more mixing and higher shear stresses.

The newly found kinematic viscosity will be used for this case in the rest of the Research. However, the stability of the case might still be a problem due to high sensitivity to this value. The turbulence model is sought to be corrected by changing the production of ω. This will directly have an effect on the turbulent viscosity, which can influence the stability negatively.

The case with the 2cm gate opening showed a steady flow field with the initially chosen values so these will stay unaltered.

The newly found turbulence values are shown in table 10.2.

Variable	Value 6cm	Value 2cm	Unit
ν	1.1385E-6	8.9088E-7	m/s^2
k	9.2669E-5	1.8283E-5	m^2/s^2
ω	8.1396	2.0523	$1/s$

Table 10.2: Changed turbulence parameters used for the two underflow weir cases in OpenFoam.

10.2.5. Initial Flow Field

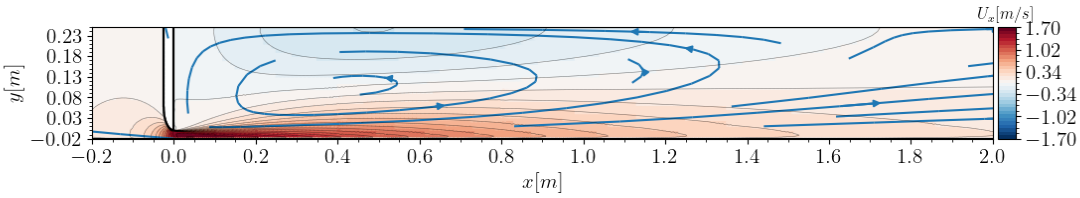
It is important to understand the results from using the baseline turbulence model. In this section, the initial flow field will be analysed and the differences with the reference data will be visualised and discussed. Both flow cases with the different gate openings show a fairly familiar flow field although some differences are present.

2cm Gate Opening The 2cm gate opening case has been the most stable case and did not show the same transient behaviour as found with the 6cm case. In figure 10.9 different field variables are presented. The turbulence features form a small plume originating from the tip of the weir, centred around the high gradients in velocity, as shown in figure 10.9c and 10.9d. This indicates not a lot of turbulence is present in the flow and it is quickly dissipated. The jet formed by the weir is clearly visible in figure 10.9a. Due to the small gate opening the water is accelerated to relatively high values of 1.7m/s compared to the upstream velocity of just 0.07m/s. Above the jet a re-circulation zone is present. Centred on this re-circulation zone above the jet the turbulent viscosity is highest, as shown in figure 10.9b.

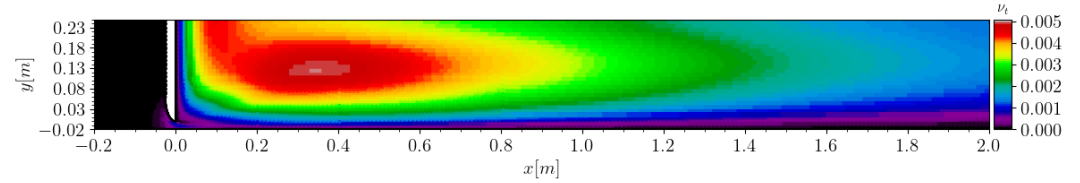
In figure 10.10 the results from the CFD simulation using the baseline model are compared with the experimental data. Here the width of on-grid block equals 0.5m/s as the velocities are scaled by a factor of 0.1. It can be clearly seen that at the gate the experimental data suggest the flow velocity is lower than predicted in the simulations. However moving downstream the opposite is clearly visible, where the velocities are under predicted in the simulations. Following the discussion, in section 10.2.3 the velocity profiles at the inlet are heavily influenced by the flow downstream and how the walls are modelled.

Next to the difference in velocity in the jet itself, a large discrepancy is visible in the shear layer between the jet and the free stream. In the simulation, the jet is much wider compared to the experimental data, which may be the reason why the maximum velocity in the jet is lower. Also, the height where the re-circulation starts is higher in the simulations. Also, the amount of re-circulation is under-predicted in the simulations. Downstream the velocity magnitudes are lower in the simulation, indicating that probably too much viscosity is present in the flow. As a result, the approach for correcting the turbulence model suggested in this research might be capable of improving the simulation results.

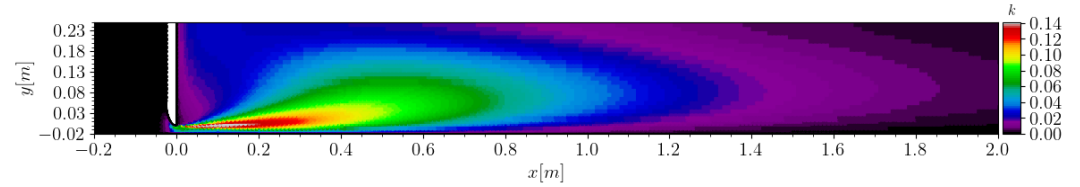
6cm Gate Opening In figure 10.11 several field variables are shown for the 6cm case. In figure 10.11a it is shown how the flow field behind the weir forms a wall-bounded jet. Velocities up to 1.4m/s are observed while the velocity upstream of the weir only has the magnitude of 0.15m/s. The core of the jet spans almost a length of 0.5m behind the weir, much longer than in the 2cm case. Above the jet a re-circulation zone is visible. At the location of high-velocity gradient, just behind the weir, the turbulent kinetic energy is sharply increasing and forms a sort of plume downstream centred around the shear layer between the jet and the free stream, as shown in figure 10.11c. Also in this region, it can be seen that the specific turbulent dissipation rate goes up, as can be seen in figure 10.11d, which is to be expected. As a result, turbulent viscosity has a maximum just behind the re-circulation zone, where due to the introduced turbulence in the flow the viscosity is also



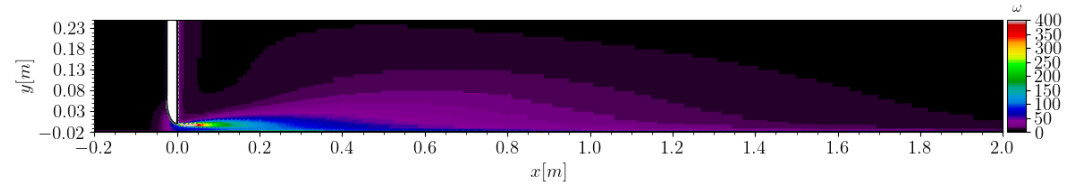
(a) Velocity in the x-direction



(b) Turbulent viscosity



(c) Turbulent kinetic energy



(d) Specific turbulent dissipation rate

Figure 10.9: Velocity and turbulent field variables plotted for the underflow weir case with a 2cm gate opening.

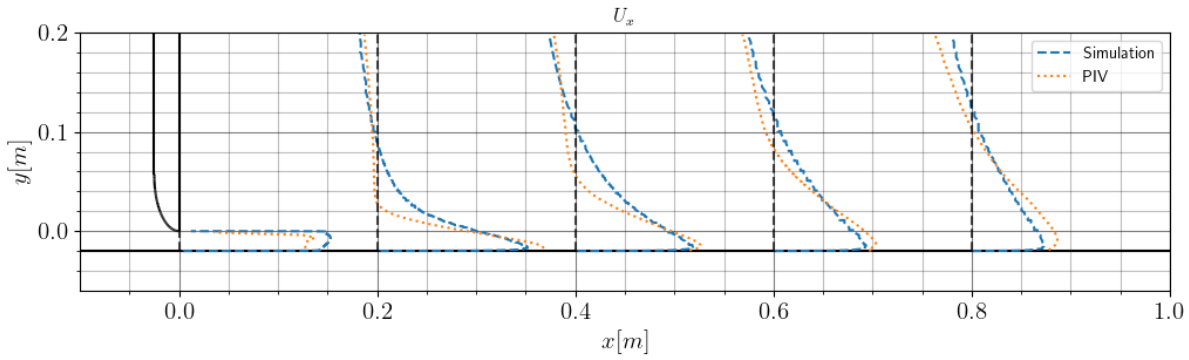
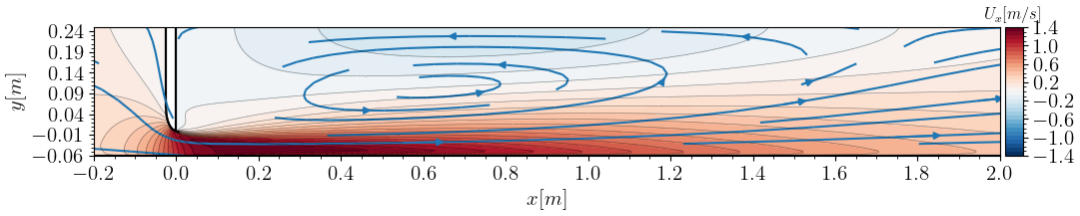


Figure 10.10: Comparison of the x-velocity field between the experimental data and the simulation results for the 2cm gate opening. The velocity profiles are scaled by a factor of 0.1, which means that the width of one minor grid block resembles a velocity of 0.5m/s.

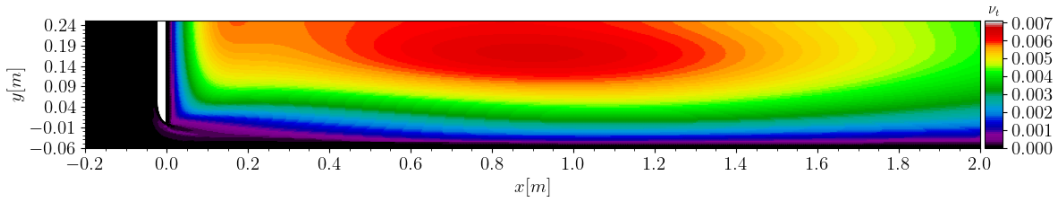
increased. The plume of the turbulent kinetic energy is much more extended and diffused downstream. Where for the 2cm case the plume was of a larger magnitude it dissipated much quicker.

When the results from the baseline simulation results are compared with the experimental data, some clear differences are visible. In figure 10.12 the velocity profiles behind the weir are shown. In the core of the jet, it can be seen that the velocities seen in the experimental data are slightly higher. But the biggest difference can be seen in the shear layer of the jet with the free stream. The jet predicted by the baseline simulation is slightly wider initially. However, after 0.6m downstream the jet as seen in the experiment is slightly larger. Also, something worth noting is that the entrainment is of a higher magnitude. In the re-circulation zone, the negative velocities are in general higher in the experiment than seen in the simulations.

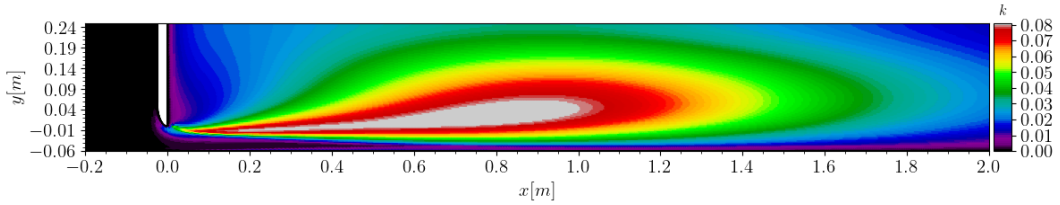
The biggest difference can thus be seen in the shear layer between the jet and the free stream above. In



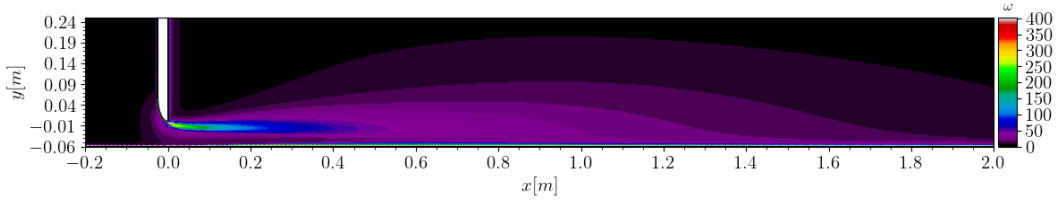
(a) Velocity in the x-direction



(b) Turbulent viscosity



(c) Turbulent kinetic energy



(d) Specific turbulent dissipation rate

Figure 10.11: Velocity and turbulent field variables plotted for the underflow weir case with a 6cm gate opening.

this region also a high turbulent kinetic energy and specific turbulent dissipation rate are seen. Therefore, as the production of the latter is going to be corrected, this shows to be an effective approach for this problem.

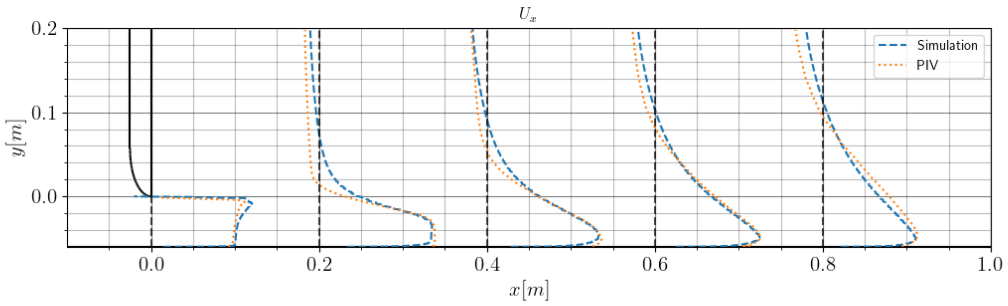


Figure 10.12: Comparison of the x-velocity field between the experimental data and the simulation results for the 6cm gate opening. The velocity profiles are scaled by a factor of 0.1, which means that the width of one minor grid block resembles a velocity of 0.5m/s.

IV

Results

11

Results Adjoint

In this chapter, the results of the new adjoint solver will be discussed. Compared to previous work the solver has been improved upon in several ways. Some minor bugs were present in the solver which was used as a reference. Also, the cost function has been rewritten to be able to cope with reference data living on a different grid than the simulations. These changes were verified on the principal cases and the results will be presented here. After that, the power of the new cost function will be shown by comparing different results. Finally, the results of the adjoint solver applied on the underflow weir case will be discussed. As mentioned earlier some interesting features of the adjoint will be shown and explained.

11.1. Verfication of the Adjoint Solver

As several changes were made to the adjoint solver, verification is necessary to test the validity of the results. The adjoint solver ultimately determines the sensitivity of the cost function to the corrective field. This can also be done in a different way than using an adjoint method. Instead, a simple finite difference method can be used to compute the gradient of the cost function with respect to the corrective field. The reason the adjoint methodology is chosen over a simple finite difference method is that it is superior in computational time. However, due to the simplicity of the finite difference method, this can be used to verify the gradients computed by the adjoint solver.

A simple finite difference method calculates the gradient with equation 11.1. As may be already apparent from the equation, the finite difference method determines a single gradient value. This means that the sensitivity of the cost function is determined with respect to a single change in the corrective field. In contrast, the adjoint solver computes the complete gradient field in one go. For the finite difference method to compute this field, it has to be run as many times as there are cells in the domain. Every simulation run, a single value of the corrective field is changed to determine the sensitivity to that single term. With simulations of close to a million cells, it is easy to conclude the method becomes unfeasible.

$$\frac{\delta J}{\delta \beta_i} = \frac{h(\beta_i + \Delta\beta) - h(\beta_i)}{\Delta\beta} \tag{11.1}$$

In this case, we use the finite difference method to compute the sensitivity along a line in the domain. Although still very computationally expensive, this is an easy and robust method to check for sensitivities. The problem however is the accuracy of the simulations. For every point on the line, a single value in the complete corrective field is changed. This has a small effect on the solution. However, when only a small window of reference data is used the cost function gets a very small value. This causes the sensitivity to be very low as well. To get to this accuracy the simulation residuals have to be very low as well, to the point, it is not feasible anymore. For this reason, when the amount of reference data points becomes a lot smaller than the number of cells in the simulation domain, the finite difference method will not give correct results anymore.

11.1.1. Periodic Hill

For the periodic hill case, two verification tests have been performed. First, the adjoint solver is tested with full-field reference data. Then just a small window of reference data is used around the crest of the hill, as around this region the simulation has the most problems.

So first the adjoint solver is run with full-field reference data. In figure 11.1 the source term $\partial J/\partial u$ is shown. The source term is the driver for the adjoint solver as all the boundary and initial conditions are zero. It clearly shows how the differences between the simulation results and the reference data are seen by the adjoint method.

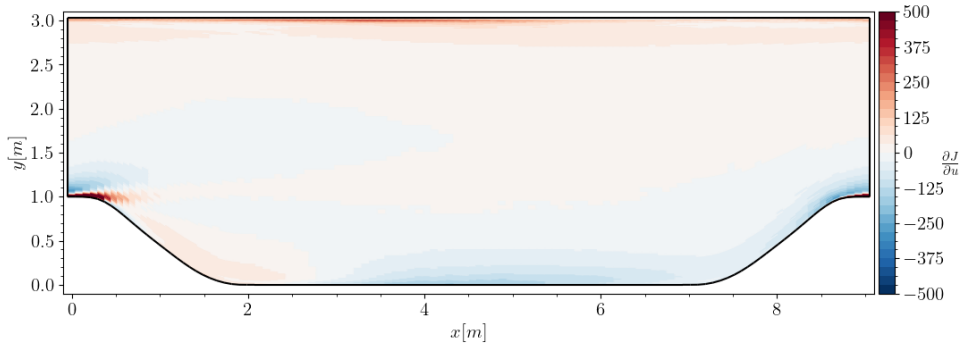


Figure 11.1: The source term of the adjoint momentum equation shown for the periodic hill case with full field reference data.

This source term results in a gradient field which shows the sensitivity of the cost function to a local change in the corrective field. In figure 11.2 this gradient field is shown for this case. A clear plume is visible originating from the crest of the hill, suggesting the model should be corrected in this area. Also in this figure, the line is shown along which the finite difference method is applied.

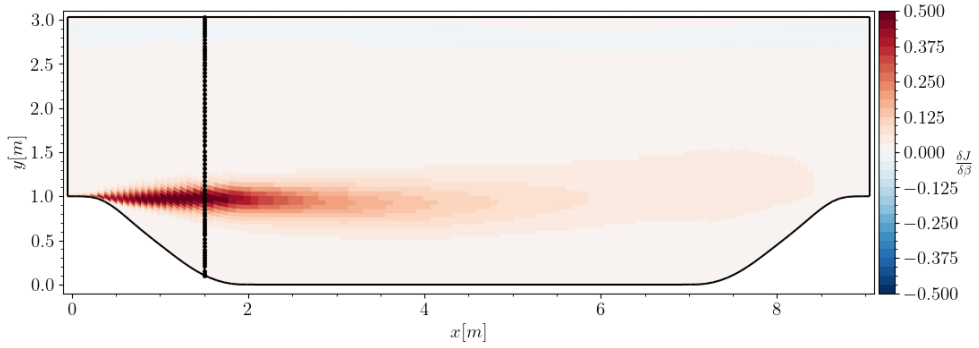


Figure 11.2: The gradient field as predicted by the adjoint solver using full field reference data.

The comparison between the output of the finite difference method and the adjoint solver is shown in figure 11.3. Here it can be seen that the adjoint solver predicts the location of the sensitivity correct. Also, the direction, which is most important is correctly predicted. However, the magnitude is not predicted correctly.

To also verify the solver working with just limited reference data, the same methodology is applied again, however now with just a window of reference data. As well the reference data is thinned out, meaning that next to giving the adjoint solver just a window, also the reference data mesh is much coarser. The resulting source term, driving the solver, is shown in figure 11.4. It can be clearly seen that the magnitude of the field is lower due to the coarser reference data. It can even be noted that in the higher refined areas zero lines are

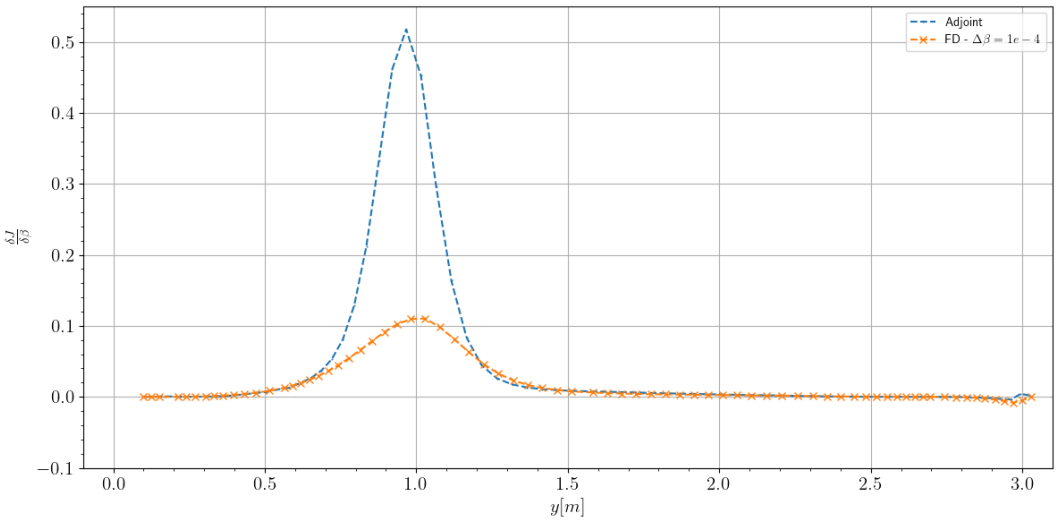


Figure 11.3: Comparison of the adjoint solver with the finite difference method using full field reference data.

visible. At these points, there is no reference data so there will be no source term as a result. The same is the case for the cells outside the window with data, where the source term also becomes zero.

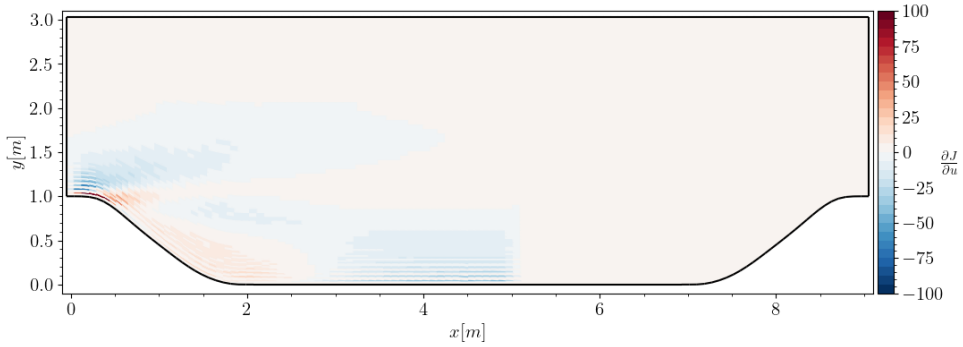


Figure 11.4: The source term of the adjoint momentum equation shown for the periodic hill case with limited reference data.

The computed gradient field for the case with limited data is shown in figure 11.5. As can be seen, is the field very comparable to when full-field reference data was used. This makes sense as the section of the domain with the highest contribution to the source term has been used. The magnitude has dropped significantly though. However, this can be easily explained with the limited amount of reference data. The cost functions value is simply a lot smaller.

The finite difference method predicts also in this case a lower gradient than the adjoint solver computes, as shown in figure 11.6. Also, the shape is slightly different, where the finite difference method predicts a wider plume. The location however is again predicted correctly.

Also, practically the results from the adjoint solver make sense. In figure 11.2 it can be seen that the adjoint solver predicts a plume originating from the crest of the hill. It thus suggests that in order to lower the cost function the corrective field has to be decreased in this area. The corrective field corrects the production term in the ω-equation of the turbulence model. A lower value in the corrective field will thus result in less production of the specific turbulent dissipation rate. Due to less production, ω will be lower in the suggested area which spreads downstream. This results in less dissipation of the turbulent kinetic energy which will

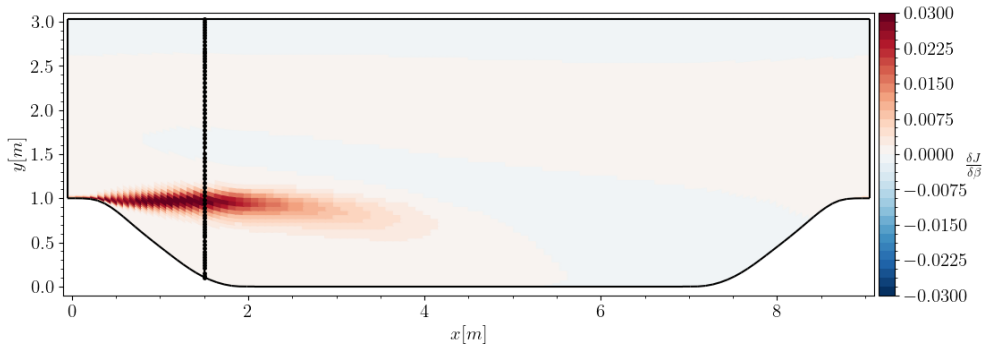


Figure 11.5: The gradient field as predicted by the adjoint solver using limited reference data.

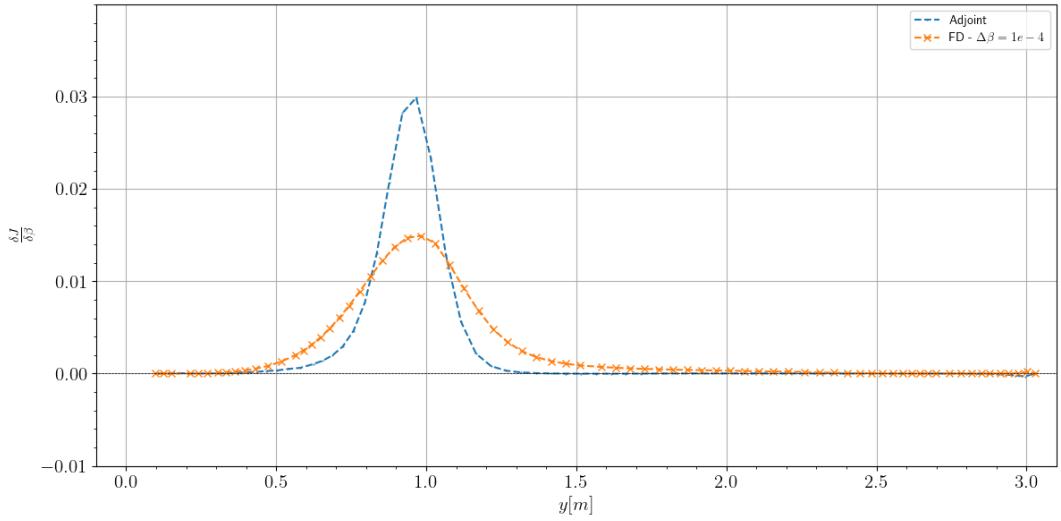


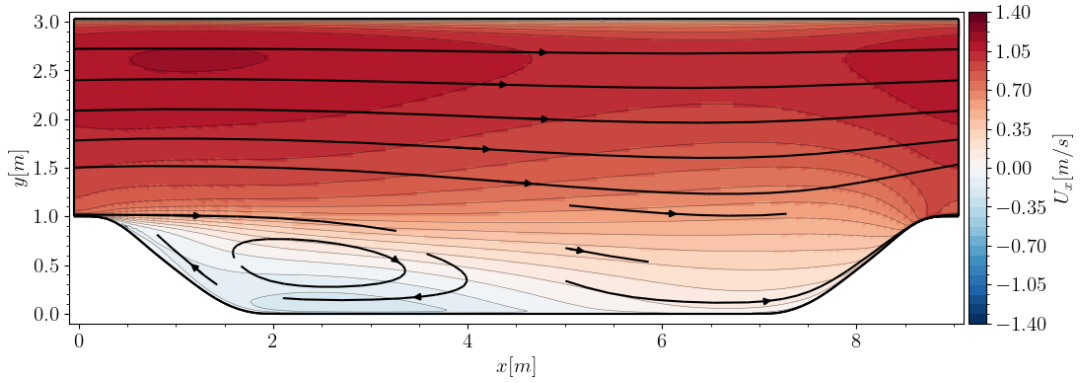
Figure 11.6: Comparison of the adjoint solver with the finite difference method using limited reference data.

thus be higher downstream. With the specific dissipation rate decreasing and the turbulent kinetic energy increasing the adjoint solver thus suggests that in the area of the plume and downstream of it, the turbulent viscosity should be higher than currently predicted by the standard model.

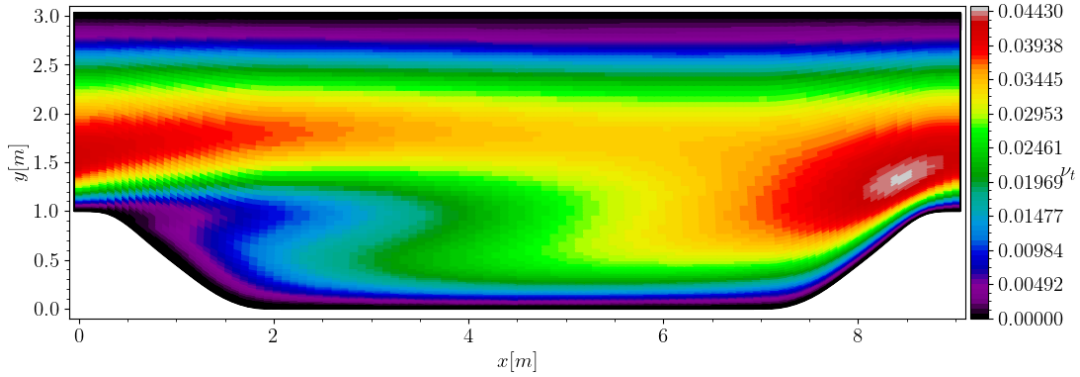
To visualise what this means for the simulation the velocity and turbulent field variables are shown in figure 11.7. It shows that where the adjoint solver suggests reducing the production of ω, already a plume is visible in ω. It thus suggests this plume is wrong in the baseline model and it should be reduced. If we look at figure 11.7b it is clear this plume in ω causes an area of low turbulent viscosity. When looking at the velocity field this is an area of high shear as the velocity gradients are high. The adjoint solver is thus suggesting that the viscosity in this high shear region should be increased. This result is what is to be expected as the $k-\omega$ model is known to under predict the eddy viscosity in shear layers. As a result the separation bubble is over predicted in size and the reattachment point is to far aft.

11.1.2. Underflow Weir

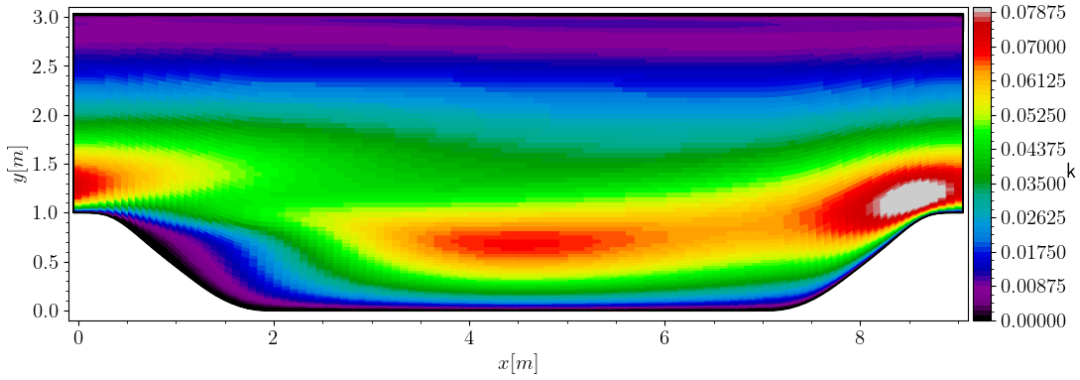
Now it is shown the adjoint solver computes correct gradients on the principal cases, it is also good to check how it performs on the more complex underflow weir case. The mesh is more complex than the principal cases and the data is of less quality. Also, the underflow weir cases are far larger and have a higher Reynolds



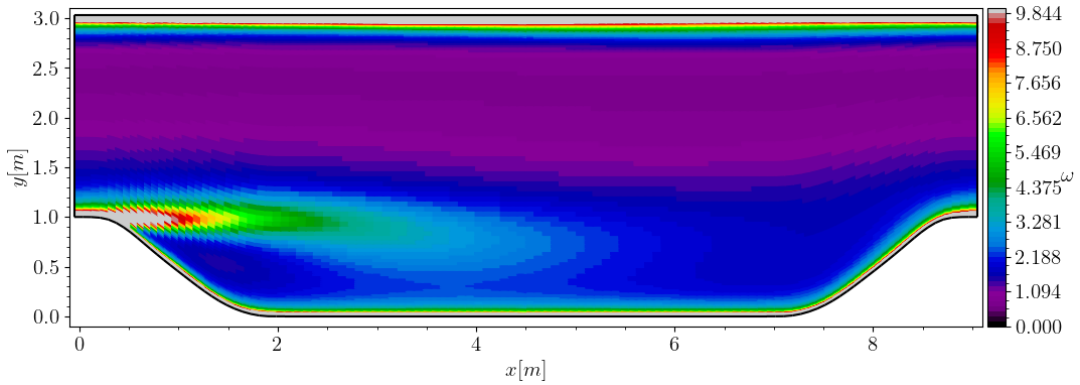
(a) Velocity in the x-direction



(b) Turbulent viscosity



(c) Turbulent kinetic energy



(d) Specific turbulent dissipation rate

Figure 11.7: Velocity and turbulent field variables plotted for the periodic hill case.

number.

Again to verify the computed gradients a finite difference method is applied. However now, with using a finite difference method in this case, a few problems arise that influence the result. First, as the mesh is very fine at the places of interest, changing β in just one cell to check the local sensitivity will not work. The problem is that the change will not affect the cost function enough. Either the simulation has to be converged to unfeasible low residuals or the change has to be made bigger.

A solution for this is to apply a change in β not only in one cell but in a small group of cells centred around the point of interest. The found gradient however is then not exactly equal to the gradient found by the adjoint solver. This effect is especially visible in figure 11.8 where due to the large area of change in β the gradient looks to be more smeared out. In figure 11.9 even the last section of the gradient was not captured at around 0.05m in height. This just shows how difficult it is to generate a cost function gradient with respect to the corrective field with a standard finite difference method. However, as the finite difference method is rather trivial, the results are to be trusted, keeping the caveats in mind.

For the 2cm gate opening case the location of the finite difference line is chosen at 0.05m behind the weir. The gradient is plotted over a vertical line going from the bottom of the domain to the top, see figure 11.8. The location is predicted very well by the adjoint solver. Only the magnitude is different although for both the adjoint and the finite-difference the value is in the same order of magnitude. Also, the gradient found by the finite difference is wider in the vertical direction. Although this can be caused by the way the finite difference method is made to work, as discussed earlier.

For the 6cm case, a different location is chosen to extend the verification. In figure 11.9 the gradients are computed using finite difference and the adjoint solver is shown. The location chosen here is 0.5m downstream of the weir gate. Here the gradients are a lot smaller and some positive sensitivity can be seen as well. The finite difference and adjoint solver agree with each other all most everywhere on the vertical line. Especially in the section with the positive gradient, both methods are in agreement. The negative peak higher up is predicted a bit higher by the finite difference method.

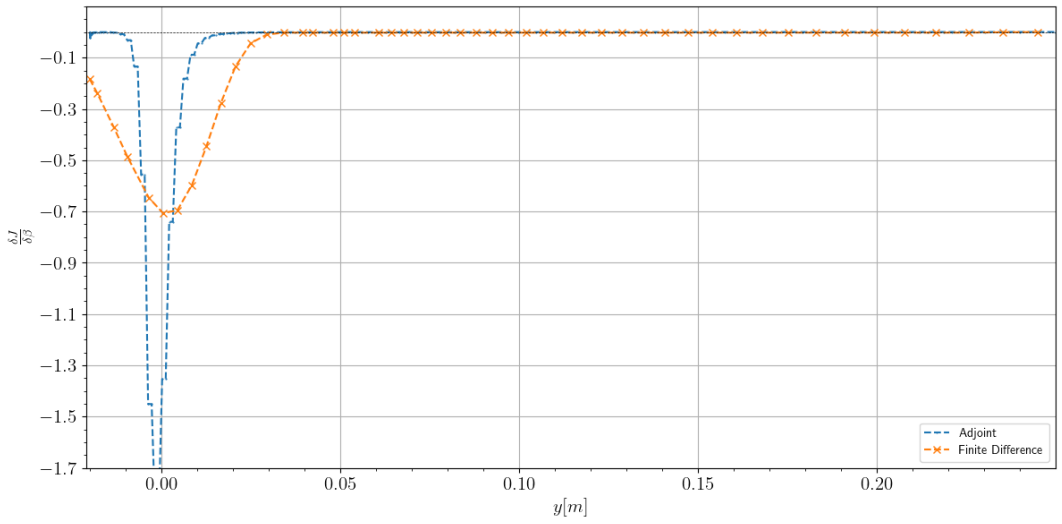


Figure 11.8: Verification of the adjoint solver on the 2cm gate opening underflow weir case at x=0.05m.

11.2. Adjoint Flow Field

In this section, the resulting flow field from the adjoint solver will be discussed. In chapter 8 some references were made to this section. Some interesting features of the adjoint solver were highlighted and in this chapter, they will be shown in practice. Some interesting findings can be made from looking at the different variable fields which give insight into how the solver operates.

To show and discuss the adjoint flow field for the underflow weir the 2cm gate opening case will be used. First, the results of the smooth mesh will be discussed. Afterward, the mesh with refinement zones is going to be discussed and compared to the smooth mesh.

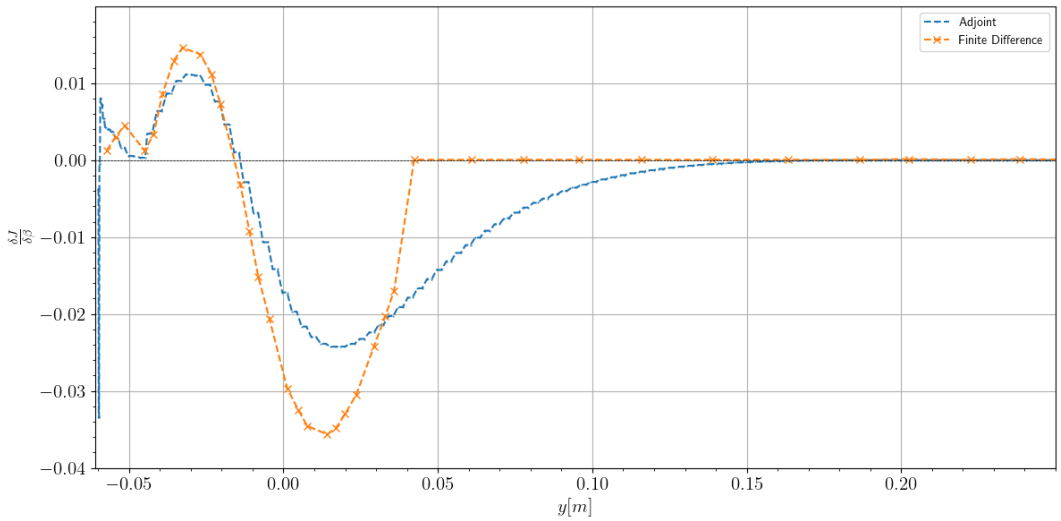


Figure 11.9: Verification of the adjoint solver on the 6cm gate opening underflow weir case at x=0.5m.

11.2.1. Smooth Mesh

In figure 11.10 the source term in the adjoint momentum equation is visualised. This shows how the experimental data drives the adjoint solver. A few interesting observations can be made. The first one can note that this source term field resembles the delta between the simulation data and the experimental data.

Another thing that can be noted is the bands around 0.4m and 0.8m. At these points, the PIV windows were overlapping. In this region twice the amount of data points is present. Therefore either the value of the source term in the cells is doubled or the density of cells with a source term is increased. When the difference between reference data and the simulation is large this effect is also large. However in the field inversion process the corrective field is changed such that the simulation results come closer to the reference data. This effect will as a result be less pronounced. It also makes sense that there seems to be a higher source term in this region as there is simply more data present.

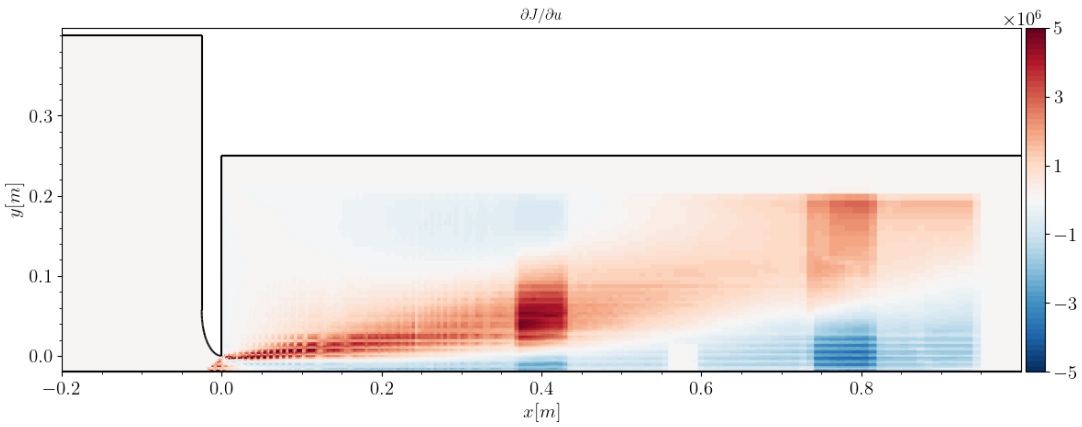


Figure 11.10: Source term, $\partial J/\partial u$, in the adjoint momentum equation visualised for the underflow weir case with a 2cm gate opening.

In figure 11.11 the adjoint variables are shown. Immediately one can see the flow field looks to be reversed. There is no need of showing the end of the domain as the flow field variables are zero there. From the section where the source term in the adjoint momentum equation is introduced the adjoint fields start to be non-zero. The effect of the source thus does not propagate downstream but upstream. Mathematically this also makes sense as the convective term in the adjoint equations, which uses the primal velocity and not the adjoint velocity, is a negative term.

As a result the argument made in section 8.4 is also proven. It was argued that the complex boundary conditions at the outlet could be replaced by simple zero Dirichlet boundary conditions as all the terms in the boundary condition equation were practically zero in this case.

Another thing that can be noted is the high magnitude. This is solely dependent on the source terms and thus on the cost function. When a lot of data is available, as is the case in this research, the adjoint variables will take on high magnitudes. There is also a dependency on the mesh volume, as the inverse is present in the source term in the adjoint momentum equation. When a fine grid is used for the simulations the source term becomes very large.

At the inlet, different boundary conditions have been applied than should be according to the derivation. For the adjoint velocity and turbulence values, the boundary condition should be zero Dirichlet however zero Neumann has been used. Looking at the flow field this change makes sense as the fields are non-zero at the inlet boundary. Mathematically this is not correct though. As a result, the boundary integral, resulting from the derivation of the adjoint equations, along the domain is not zero anymore. If the correct boundary conditions are to be used the location of the inlet should have been moved further upstream. With the current domain, the inlet boundary is too close to the features of importance. When a zero Dirichlet boundary condition is applied the flow field will be affected too much as it will practically hit a wall. To resolve this issue the inlet should thus be moved further upstream. However, when looking closely at the flow field the adjoint variables seem to be dissipating naturally. If zero Neumann boundary conditions are used the flow field is not affected too much. The assumption is then made the adjoint variables will go to zero past the inlet boundary, and thus the boundary integral will be zero when the extended domain is taken into account. The effect of this assumption is very minimal. It can be argued that the adjoint flow field around the weir will not be much different between the extended domain with the correct boundary condition and the current domain with a zero Neumann condition. Because of this, practically this approach is still correct. A big motivator for this approach is reducing the mesh count and thus computational time. The value of interest, the sensitivity of the cost function with respect to the corrective field, is a function of the adjoint omega field and the production field. Around the inlet, the production in the flow will be practically zero. If the adjoint omega field is predicted somewhat wrong due to the used boundary conditions it will have no effect on the cost function sensitivity. The production field will normally have a value around the areas of interest, where the domain is centred around.

These fields are computed to be able to calculate the adjoint gradient field. The sensitivity of the cost function with respect to the corrective field. For the underflow weir case with a 2cm gate opening and using the baseline $k-\omega$ model, this gradient is shown in figure 11.12. This shows where the adjoint solver determined the production of the turbulent specific dissipation rate should be changed.

Around the tip of the weir gate, some sensitivity can be seen. Especially downstream a plume-like shape can be seen. Increasing the corrective term in this region and thus increasing the turbulent specific dissipation rate production should bring the simulation output closer to the experimental data.

By increasing the turbulent specific dissipation rate the eddy viscosity is decreased. This ties in with what we see in the difference between the data and simulations. In the data, the jet is much more pronounced while in the simulations the jet is more washed out. Decreasing the viscosity here locally will reduce the shear stresses in the flow. This could lead to a stronger jet which dissipates less.

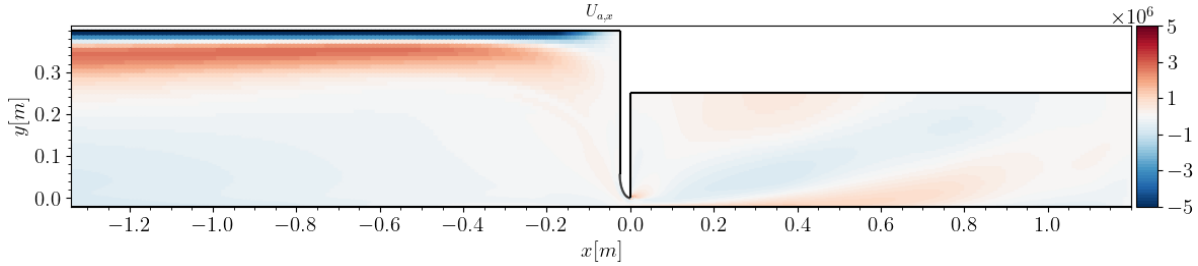
11.2.2. Mesh Refinements

In figure 11.13a the source term in the adjoint momentum equation is visualised for the mesh with refinement zones. Clearly, the effect of the mesh refinement can be seen. In the zones of highest refinement, the magnitude is higher than in sections of the domain with a coarser mesh. The reason behind this is that the source term in OpenFoam is normalised by the cell volume. In the areas where the cell volume is smallest the source term will be highest.

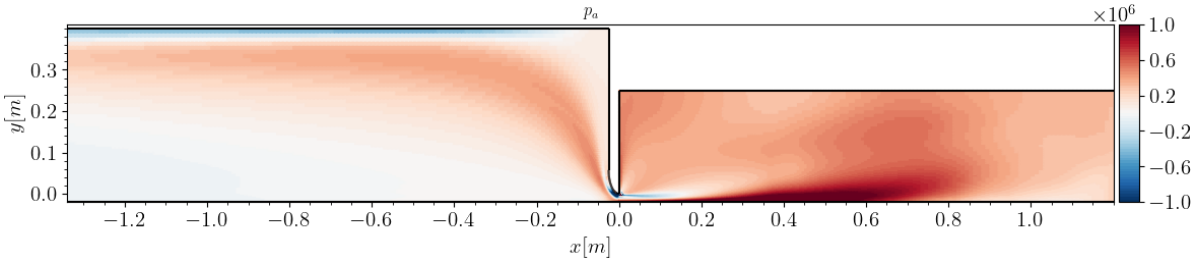
This mesh dependency seems an unwanted feature of the solver. However, when looking at figure 11.13b it becomes clear why this mesh dependency is actually not present. The experimental data lives on a grid differently than the simulation grid. When the reference data lives on a coarser grid than the local simulation grid only a few cells in the simulation will have a source term, as can be seen in figure 11.13b. All equations are linear and thus the solution is a linear combination of the sources. When a coarser grid had been used for the simulation data the cell values would be lower while fewer zero source cells will be present. The linear combination will be the same in both cases.

The same argument can be made when the reference data lives on a finer grid. In that case, the effect of the multiple data points living in one simulation grid cell will be added together to get the final source term value.

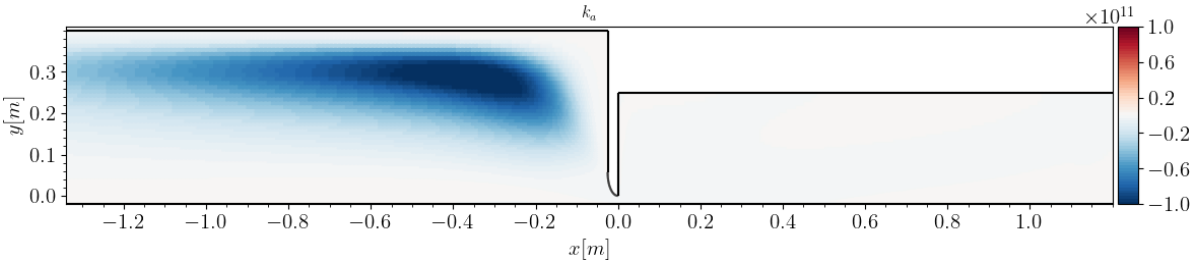
A final comment that can be made is that the reference data is 2D. Around the weirs tip refinement zones have been applied which increase the number of cells in the span-wise direction as well. The simulation



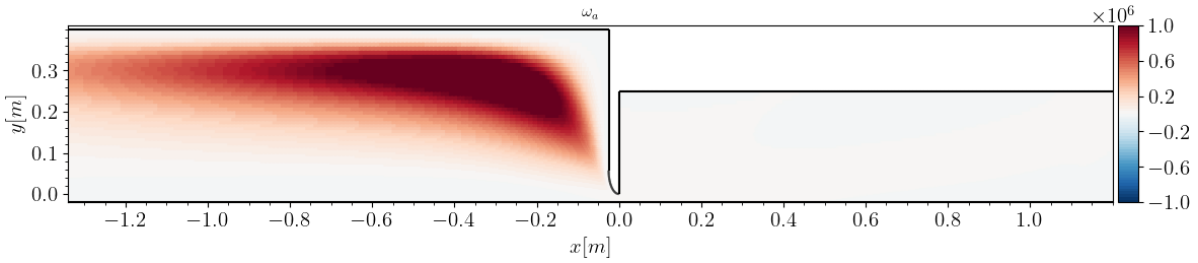
(a) Adjoint velocity in the x-direction



(b) Adjoint pressure.



(c) Adjoint turbulent kinetic energy



(d) Adjoint specific turbulent dissipation rate

Figure 11.11: Adjoint velocity, pressure and turbulent field variables plotted for the underflow weir with a 2cm gate opening.

in this region is thus also not strictly 2D. The data mapping finds a grid cell of the simulation in which the reference data point is located. The same effect we see in figure 11.13b, where the source term is very local, will be present in the span-wise direction.

The resulting adjoint field variables are not that different between the different meshes. Therefore they are not shown for this mesh for the sake of brevity. The resulting adjoint gradient, however, is very different. This field is shown in figure 11.14. A few interesting things can be observed here regarding the adjoint solver.

First of all, a clear mesh dependence is visible. When the mesh goes to a step coarser a jump in the gradient can be seen. This effect is especially visible upstream of the weir. The adjoint solver is thus suggesting to correct the model more in coarse regions of the mesh than in finer regions of the mesh.

The reason for this strong mesh dependency becomes clear when looking at equation 11.2 where the calculation of the adjoint gradient is shown. The volume integral in the second term gives the method a strong mesh dependency. The flow fields resulting from the adjoint solver are all reasonably smooth.

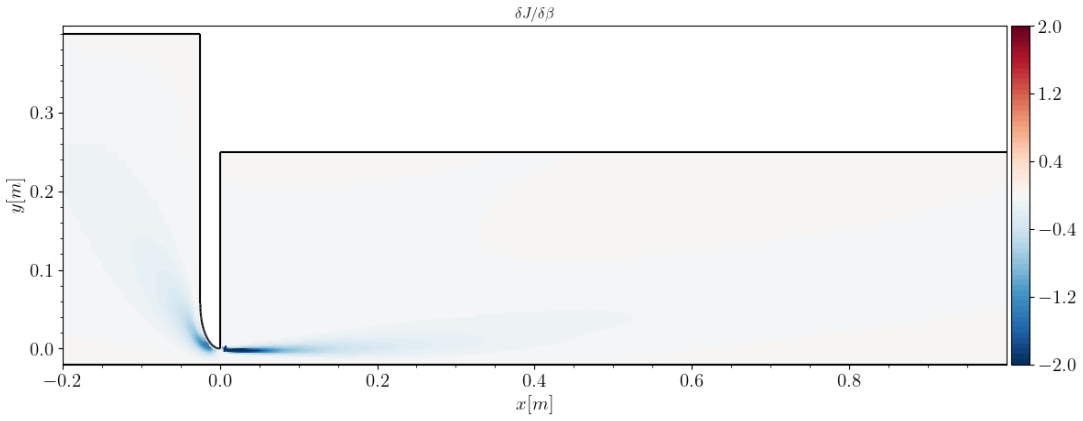
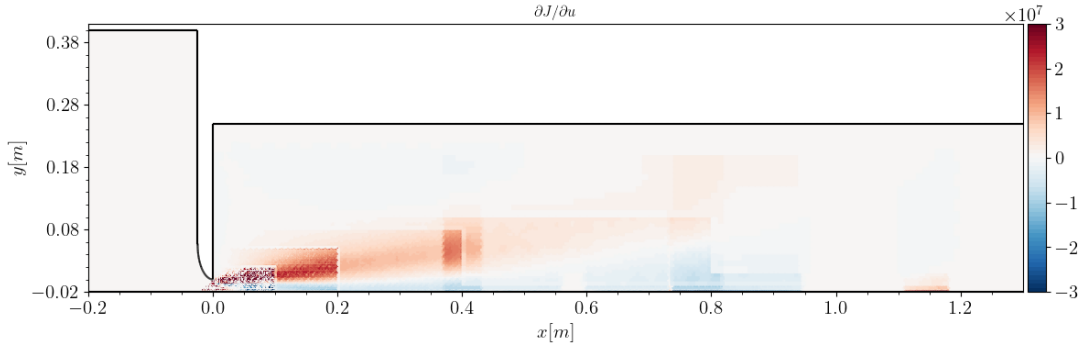
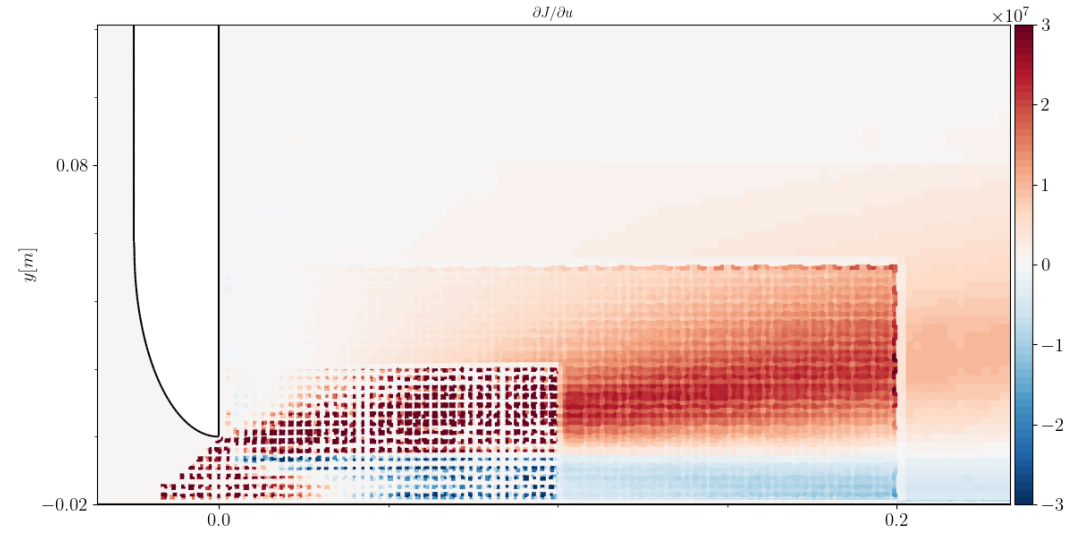


Figure 11.12: Cost function gradient with respect to the corrective field for the under flow weir with a 2cm gate opening.



(a) Source term, $\partial J/\partial u$, in the adjoint momentum equation visualised for the underflow weir case with a 2cm gate opening.



(b) Source term, $\partial J/\partial u$, in the adjoint momentum equation visualised for the underflow weir case with a 2cm gate opening, zoomed in.

$$\frac{dL}{d\beta} = \frac{1}{\sigma_\beta^2}(\beta - \beta_0) - \int_\Omega \omega_a \gamma P d\Omega \tag{11.2}$$

11.3. Computational Cost

As the adjoint equations are very similar to the governing equations used for the simulations the same approach is used to solve them. In this section, the computational cost of the adjoint solver is discussed and

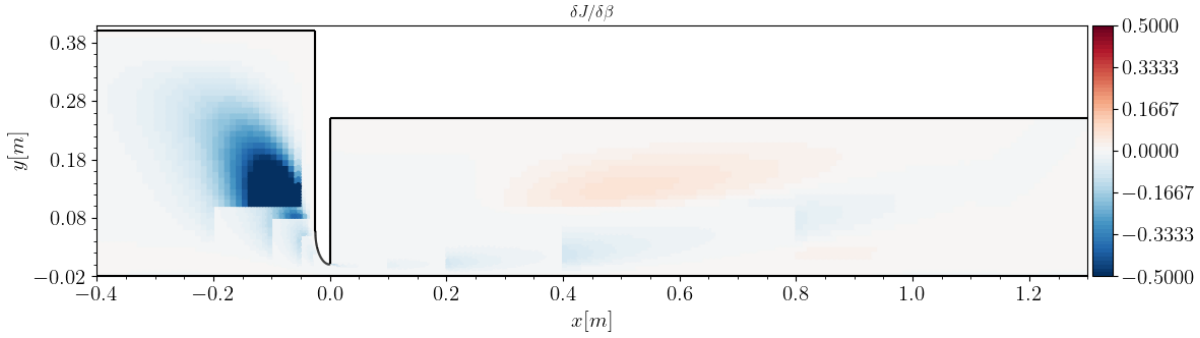


Figure 11.14: Cost function gradient with respect to the corrective field for the under flow weir with a 2cm gate opening.

compared to the cost of a normal CFD simulation.

The computational cost obviously depends on many parameters. From the chosen numerical scheme to the chosen relaxation factors, there is a lot of variables that determine the time it takes for a simulation to converge. Therefore the numbers mentioned here are all based on the underflow weir with a 2cm gate opening case using the smooth mesh. The chosen settings were all based on stability and computational time. This should thus be a good example of the performance in practice.

For this example, the simulations were run in parallel. A normal simulation, converging the primal flow fields, took 2419 iterations which took 418 seconds to converge. When compared to the adjoint solver this took up to 10000 iterations and 2021 seconds to converge. That is almost 5 times slower. Also, the solver was stopped when all the adjoint variables had a residual of $1e^{-5}$ or lower while the normal simulations were stopped when all the variables reached a residual of $1e^{-6}$ or lower. It thus takes a significantly greater time for the adjoint solver to converge. One caveat to this is that the relaxation factor was set to less than half for the adjoint variables compared to the primal variables.

12

Results Field Inversion

In this chapter, the field inversion results from the underflow weir cases will be discussed. First, the 2cm case will be discussed which was a success. Afterwards the 6cm case will be discussed which had less success due to model limitations.

12.1. 2cm Gate Opening

The results from applying the paradigm of field inversion on the first underflow weir case with a gate opening of 2cm will be discussed here. First, looking at the cost function sensitivity computed with the baseline turbulence model the direction of the optimisation process can already be reviewed. A negative sensitivity can be spotted around the weir gate tip, see figure 12.1. This sensitivity propagates upstream and downstream. Especially around the weir, the sensitivity is fairly high. This suggests the production of the turbulent specific dissipation rate should be higher around the weir. An increase in the specific dissipation rate will lower the turbulent kinetic energy in the flow. This combination will in turn lower the eddy viscosity.

The adjoint solver thus suggests that in the region around the tip of the weir too much eddy viscosity is present. Lowering the eddy viscosity at the front face of the tip weir will maybe increase the velocity of the water at the gate opening as fewer shear stresses are present in the high shear flow. In section 10.2 the simulation flow field has been compared to the experimental data. Here a discrepancy in the velocity at the gate opening was found for this case.

Behind the weir, the jet was too much washed out in the simulations. In the experimental data, the jet was much more pronounced and a sharper velocity decrease was found at the edge of the jet. Reducing the eddy viscosity in this region will reduce the shear stresses locally. This in turn might cause less diffusion of the jet.

The gradient-based optimisation process, using a line search algorithm to find the appropriate step size, was able to reduce the cost function value by a factor of two. In figure 12.2 this value is plotted over the iterations. Also in this figure the step size, alpha, is shown. Both are initially sharply decreasing. After two steps just minor improvements were found. Also at this point, the step size became very low. At this point the improvements in the cost function were also minimal and within the error range of the simulations.

The algorithm stopped because after five iterations the cost function sensitivity was found too low. In figure 12.3 the adjoint gradient is shown. The sensitivity around the weir is completely gone, only a small plume is visible originating from the tip of the weir. This shows that, within the capabilities of the paradigm, the optimal corrective field, the MAP solution, is almost found.

The resulting corrective field is shown in figure 12.4. Around the front face of the tip of the weir, the production term of the turbulent specific dissipation rate has been increased by a factor of 1.5. Behind the weir, two distinct traces can be seen. The bottom one is following the core of the jet, while the top trace is located around the shear layer between the jet and the free stream. Apparently, in both locations, an increase in turbulent specific dissipation rate production was needed. Also what can be noted is that the corrections are very localised around the weir. The paradigm did not come up with a correction far upstream of the weir.

In figure 12.5 it is shown how the turbulent fields are corrected by the paradigm. As already expected a reduction in the eddy viscosity and turbulent kinetic energy can be seen, while the turbulent specific dissipation rate is increased around the shear layer between the jet and the free stream.

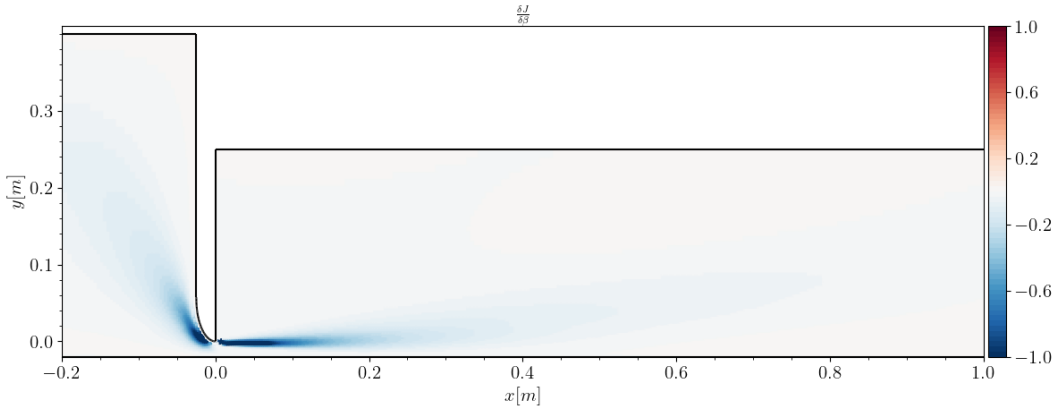


Figure 12.1: The adjoint gradient, $\delta J/\delta \beta$, using the baseline $k-\omega$ model for the 2cm gate opening case.

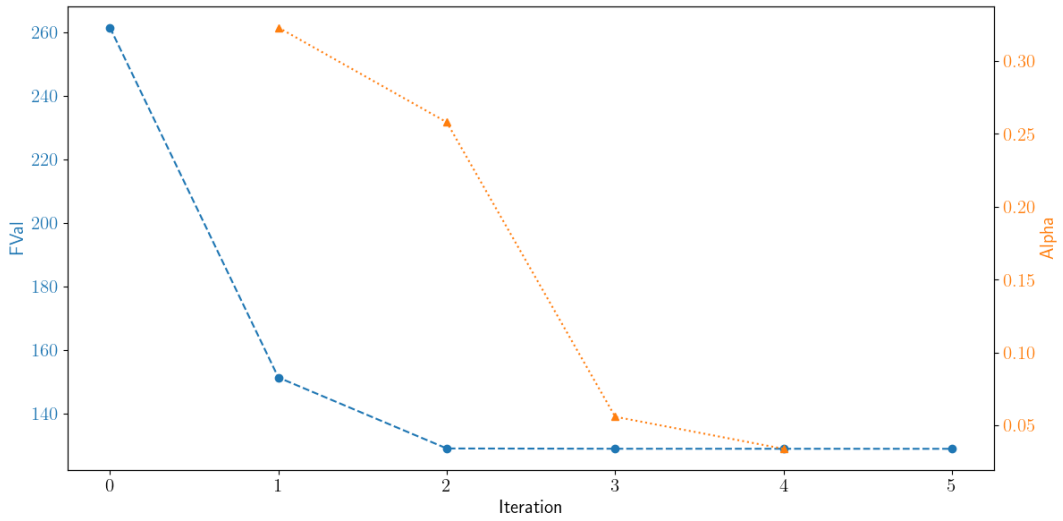


Figure 12.2: The values of the cost function and the step size, alpha, over the gradient descent iterations.

The turbulent kinetic energy has been changed quite a bit, see figure 12.5a. Initially, the magnitude has been reduced by a factor of 0.6. Also, it is far more localised around the shear layer of the jet with the free stream. Especially above the jet, the turbulent kinetic energy has been reduced greatly. Something else that can be noticed is that the field diffuses less downstream. Far downstream the turbulent kinetic energy is actually increased after correcting the model. This might indicate the jet velocity has been increased.

In the turbulent specific dissipation rate field, there are not many surprises. The magnitude has been increased almost by a factor of two in the shear layer of the jet. This dissipates going downstream where far downstream the values are almost equal.

The eddy viscosity field is shown in figure 12.5c. This field is a result of the two turbulence fields discussed earlier so it is thus no surprise that a reduction is seen. Especially close to the gate a reduction can be seen around the jet close to the bottom. Far downstream the magnitude is actually higher than in the baseline model.

The changes in these turbulence fields had as a goal to get the velocity field closer to the data. In figure 12.6 the baseline and corrected model are compared with the experimental PIV data and the results from the high fidelity, DES, simulation. Here a clear improvement can be seen in the velocity field prediction when using the correction. The velocities in the jet core have been increased. Close to the gate, the velocities in

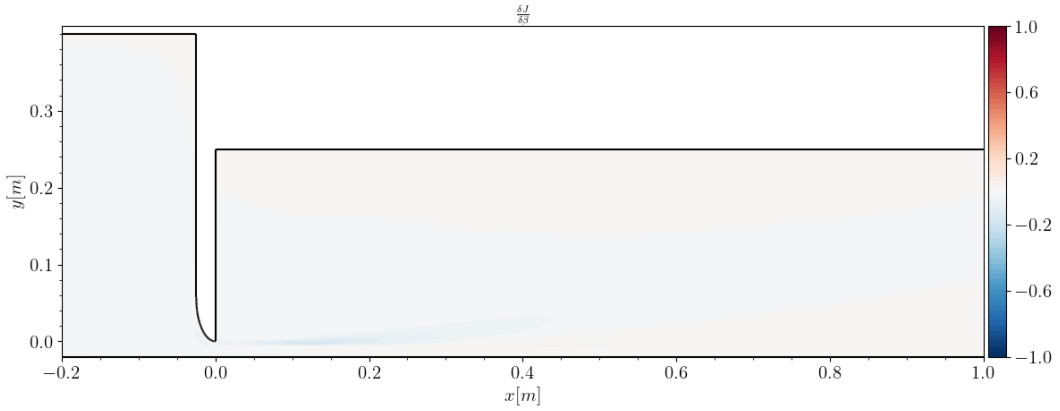


Figure 12.3: The adjoint gradient, $\delta J/\delta \beta$, using the corrected $k-\omega$ model for the 2cm gate opening case.

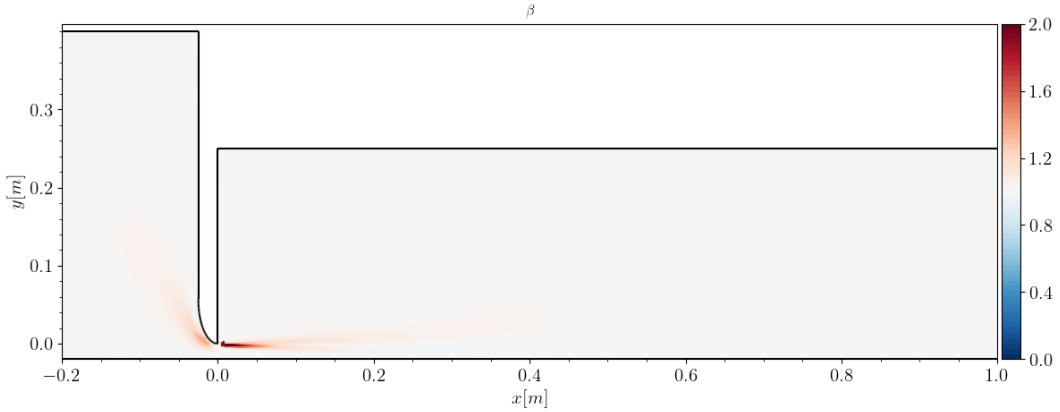
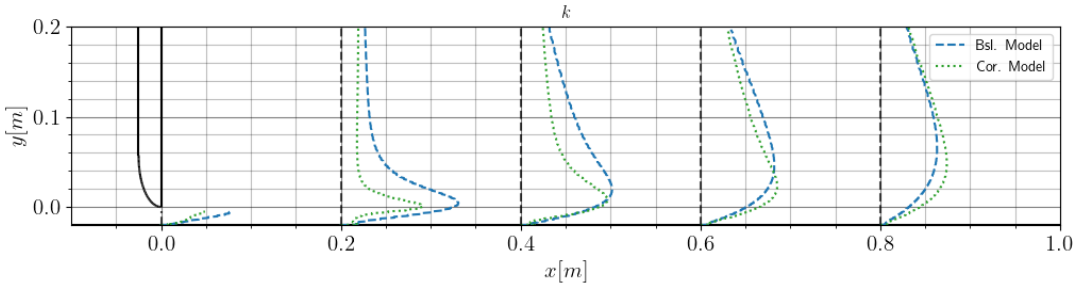


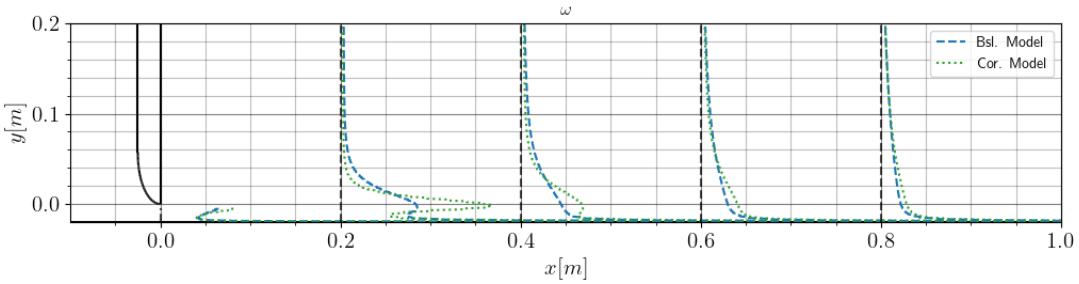
Figure 12.4: The corrective field found after the gradient descent optimisation process, reducing the cost function by a factor of 2 for the 2cm gate opening case.

the core match the data quite well. However, due to the correction, a slight overshoot can be seen further downstream. Also, the jet is less diffused going downstream. Reducing the eddy viscosity in this area had the desired effect of getting higher velocity gradients in the shear layer. Around 0.2m the corrected model comes very close to the experimental data, and a huge improvement with respect to the baseline model is seen. Further downstream the improvement is less, however here the baseline model has been already quite good. The solution from using the high fidelity DES model is still better than the corrected model. Although the jet core velocities look to be over predicted across the whole length of the jet. It is however not strange that the high fidelity model gives a better prediction. The simulation is in 3D and the model is unsteady. This will obviously have a big effect on the performance. A steady solver in a 2D domain will not be able to be corrected to fully match a 3D unsteady problem.

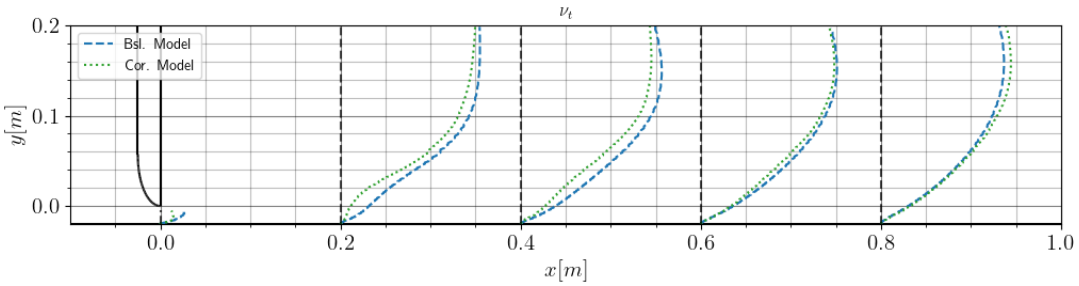
In figure 12.7 the delta between the simulations and the reference data is shown. Overall the delta has been decreased. Here it becomes clear that the velocities in the core of the jet have been slightly overshot so that they are higher than in the reference data. In the shear layer, a clear improvement can be seen. In the free stream above the recirculating flow has been reduced, however also here it has been overshot slightly. The prediction in the jet core is way better than compared to the results from the DES simulation. However, in the shear layer, the DES simulation is still superior.



(a) Turbulent kinetic energy field comparison between the baseline and corrected turbulence model.



(b) Turbulent specific dissipation rate field comparison between the baseline and corrected turbulence model. The field is scaled so that 0.2m equals to a ω of 200 1/s.



(c) Turbulent viscosity field comparison between the baseline and corrected turbulence model. The field is scaled so that 0.2m equals to a ν_t of 0.006m^2/s.

Figure 12.5: Comparison between the baseline and the corrected model for the turbulence fields in the 2cm gate opening case.

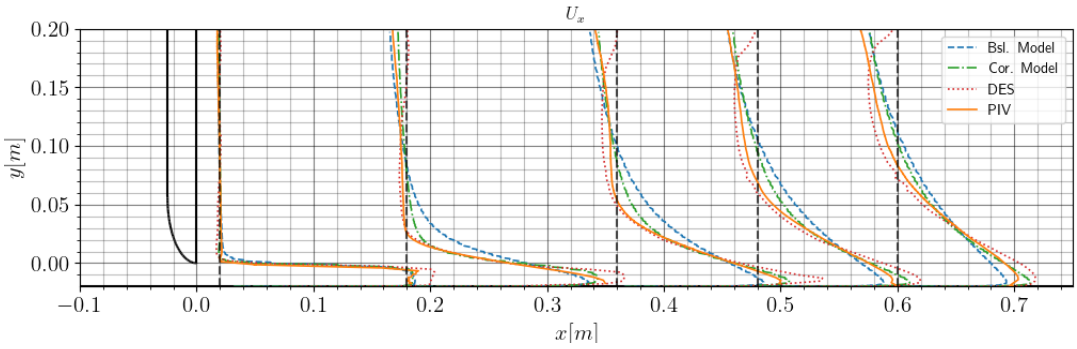


Figure 12.6: Comparison of the stream wise velocity between the baseline model, the corrected model and experimental data. The field is scaled so that 0.2m equals to a u_x of 2m/s.

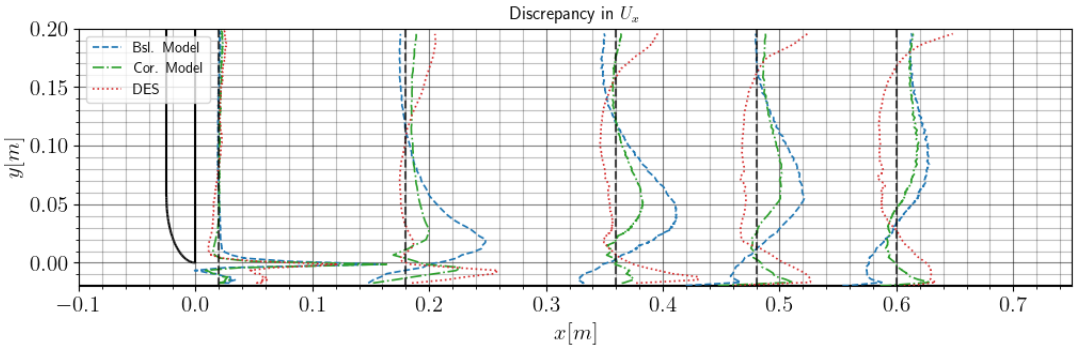


Figure 12.7: Difference in stream wise velocity between the baseline, and corrected simulation and the experimental data. The field is scaled so that 0.2m equals to a Δu_x of 0.4m/s.

12.2. 6cm Gate Opening

In section 10.2 the stability of the 6cm gate opening cased has been discussed. It was found that with the originally chosen value for the kinematic viscosity, the simulation showed transient behaviour and the solver could not find a solution. The value has been increased to stabilise the solution. From this point, the field inversion process has been started.

In figure 12.8 the cost function sensitivity to the corrective field is shown when the baseline turbulence model is used. This shows the first direction in the gradient-based optimisation phase. Around the tip of the weir, the adjoint solver found some sensitivity to changing the corrective field. The sensitivity is spread out in a plume-like fashion behind the tip of the weir.

The adjoint solver thus suggests increasing the corrective term in this region to increase the turbulent specific dissipation rate. Increasing this variable, however, will, in turn, decrease the turbulent viscosity and thus increase the effective Reynolds number. The problem with this is that while setting up the baseline case unsteady behaviour was found in this region as well. This was solved by increasing the kinematic viscosity and thus stabilising the flow.

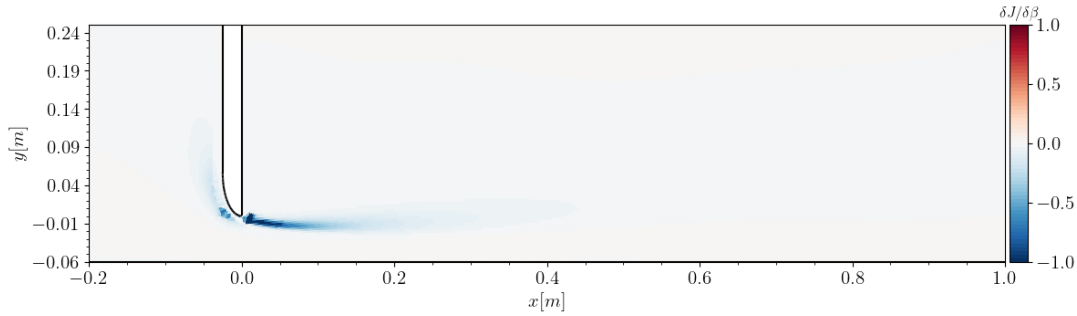


Figure 12.8: The adjoint gradient, $\delta J/\delta \beta$, using the baseline $k-\omega$ model for the 6cm gate opening case.

For this reason, the problem of an unsteady flow behind the weir came back in the optimisation process. Two iterations of the gradient descent optimisation were performed. After this second iteration, the residuals of the simulation were not converging to a low enough value anymore. As a result, the adjoint solver was not able to converge to a solution as it needs a well converged primal solution.

The field inversion process was able to reduce the cost function by 10% after two iterations. The resulting corrective field is shown in figure 12.9. It shows that the production of turbulent specific dissipation rate is increased around and downstream of the tip of the weir. In practice this makes sense as too much mixing was found in this region. The jet was much more washed out in the simulations than compared to the experimental data. Lowering the viscosity due to turbulence in this region would solve this problem.

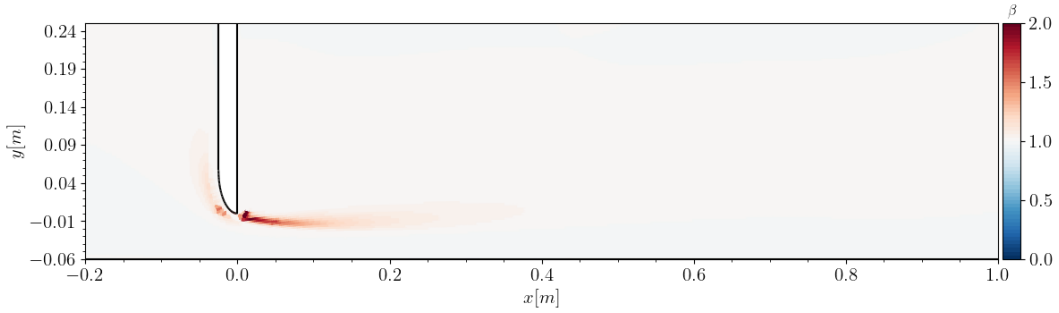


Figure 12.9: The corrective field found after two steps of the gradient descent optimisation process, reducing the cost function by 10% for the 6cm gate opening case.

The found corrective field has the predicted effect of increasing the turbulent specific dissipation rate by increasing the production locally, as can be seen in figure 12.10b. As a result, the turbulent kinetic energy decreases in the shear layer of the jet and the free stream, figure 12.10a. In figure 12.10c the eddy viscosity is shown to be decreased in basically the whole flow field above the jet. This decrease in eddy viscosity could be the reason for the case to go towards instability again.

The effect of changing the eddy viscosity is shown in figure 12.11. Here the stream-wise velocity predicted by the baseline model and the corrected model is compared to the experimental data. The corrected simulations do show a better prediction of the velocity field, although slight. In figure 12.12 the discrepancy between the simulation and the experimental data for both the baseline and corrected model is shown to get a better image of what is happening. Here it immediately becomes clear that the difference between simulation and experimental data is reduced.

The velocity in the core of the jet is slightly increased with respect to the baseline. Although the baseline model predicted the magnitude of the velocities in this region quite well. Above the jet the velocities have been reduced, resulting in a sharper transition from the jet to the free stream.

12.3. Computational Cost

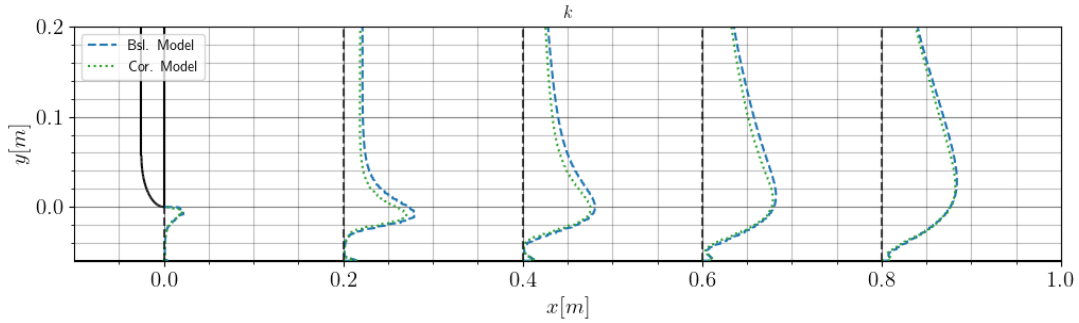
The goal of this paradigm is to improve low fidelity CFD simulations by correcting the turbulence model. A big question is thus how this compares to simply running a higher fidelity simulation.

In the previous section, the resulting flow fields have been compared, so in this section, the computational cost will be discussed.

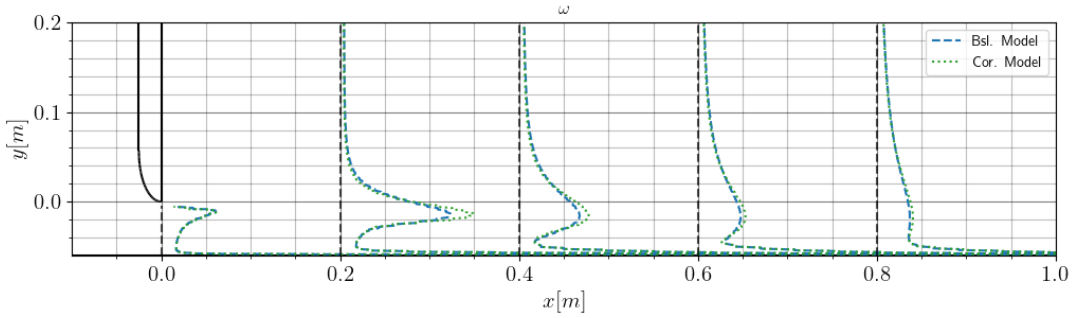
It has already been shown that the adjoint solver takes almost five times longer to converge compared to the normal CFD solver. During the inversion process however, the simulations are all initialised from a converged solution, which makes for considerably lower convergence times.

The field inversion process for the underflow weir with a 2cm gate opening took a little over twelve hours using eight cores. Obviously, this is just for the 2D case. It only took ten minutes to get a solution with a low fidelity model. It is thus a significant increase in terms of computational time. However, compared to the high fidelity simulations it is still faster.

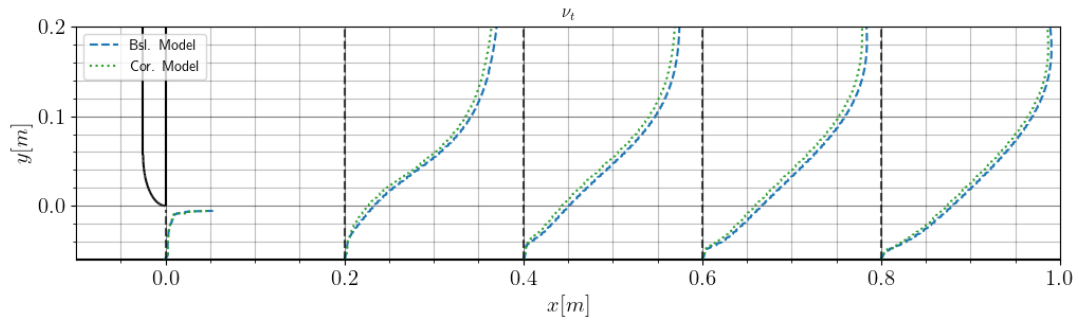
Also, the process of field inversion only has to be applied a few times for a set of different cases. The next step is to learn from the results and to train a model that is capable of further predicting the needed correction for unseen cases. As with most machine learning approaches, the training time is the most time-consuming. When the model is trained the prediction is very fast. The time needed for the corrected solution will be of the same order as for a solution from the baseline model.



(a) Turbulent kinetic energy field comparison between the baseline and corrected turbulence model.



(b) Turbulent specific dissipation rate field comparison between the baseline and corrected turbulence model. The field is scaled so that 0.2m equals to a ω of 200 1/s.



(c) Turbulent viscosity field comparison between the baseline and corrected turbulence model. The field is scaled so that 0.2m equals to a ν_t of 0.006m^2/s.

Figure 12.10: Comparison between the baseline and the corrected model for the turbulence fields in the 6cm gate opening case.

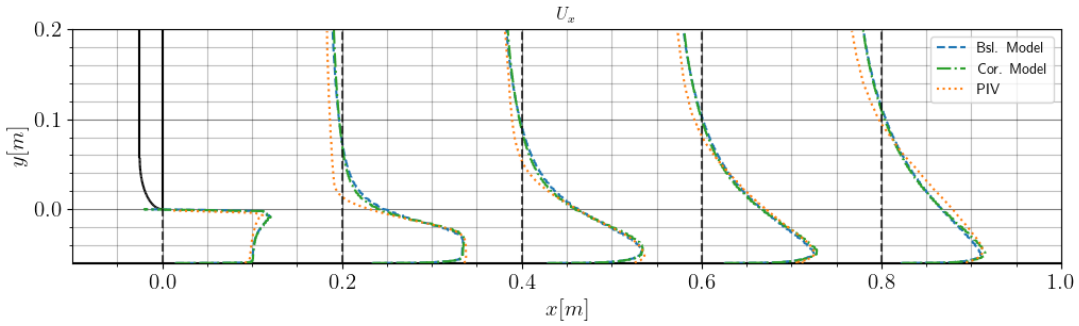


Figure 12.11: Comparison of the stream wise velocity between the baseline model, the corrected model and experimental data. The field is scaled so that 0.2m equals to a u_x of 2m/s.

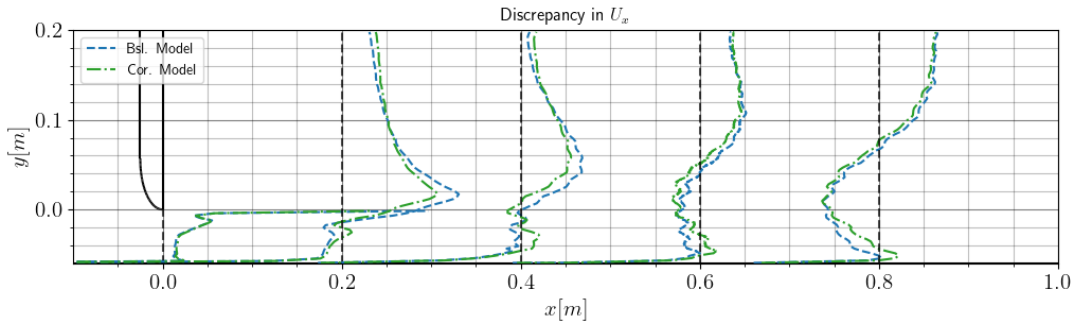


Figure 12.12: Difference in stream wise velocity between the baseline, and corrected simulation and the experimental data. The field is scaled so that 0.2m equals to a Δu_x of 0.4m/s.

V

Conclusion

13

Conclusion

In this section, the research questions are reviewed again and sought to be answered. This will be done by first reflecting upon the results achieved in this research. Afterwards the research questions are answered using the conclusions from the results. Then the significance of this contribution will be discussed. And at last further recommendations will be shared.

A few conclusions can be made specific to the underflow case. First, a localised correction around the gate opening and the weir tip was found. Initially, there was some fear around the sensitivity of the inlet conditions. It was hypothesised that they had a significant effect on the results. This effect was definitely visible when using the baseline solver, showing some significantly different behaviour for the 6cm gate opening case. However, the corrective field showed that only the regions around the weir were sensitive to these small changes.

It is a good thing the adjoint solver finds a sensitivity around the areas of interest, the weir gate in this case. Finding a correction far away, upstream, of the weir could have been a real scenario. The location of the found correction is however a direct result of the decision on where the corrective term is applied in the model equation. The correction is applied to the production term of the turbulent specific dissipation rate transport equation. This means that the effect of this correction will only be significant when production in the flow is high. In a well designed simulation domain, the velocity gradients are often only present near the areas of interest.

A strong dependence on these kinds of flow features will also help in the next step. Ultimately machine learning would be used to train a model which uses flow features to predict the correction needed. When the correction is located around the prominent flow features, a correlation is more likely to be found.

1. Can the paradigm of field inversion be applied to more complex problems like wall-bounded jet flows behind weirs? In this research, the paradigm of field inversion has been applied to the underflow weir case. Two cases with a different gate opening have been used to test the method. For both cases a corrective field was found which improved the resulting flow field with respect to the reference data. For one case the cost function has been reduced by a factor of two.

The complexity of these cases has been increased compared to cases in earlier results. However, compared to a CFD simulation used in engineering the test cases have been on the simple side. First of all, the simulations have been mainly two-dimensional. While some three-dimensionality was present in the mesh, no real 3D effects were present in the flow field. Next to this, the positive results were all achieved on a fairly smooth mesh, while in practice this is often not feasible.

The main drawback is the stability concerns. The adjoint solver needs a well-converged flow field as an input. When the residuals from the normal simulations are above $1e^{-5}$ the adjoint solver has been seen to show convergence problems. For more complex problems it becomes more and more difficult to achieve low residuals and thus in parallel, it becomes more difficult to apply this paradigm.

Having said that, this approach has shown its capabilities. The paradigm was able to find a corrective field to improve the velocity flow field predictions, with respect to the reference data. No stability problems were found when correcting the model's equations. Only when the flow field itself became unsteady problems were found.

1.1 Is a corrective term capable of improving a complex k-omega simulation with respect to a PIV baseline? And to what extent? It has been shown that the first part of this question is true. The velocity fields, after

correcting the model, resemble the experimental data much closer. The chosen corrective term thus is capable of changing the model such that the predictions are improved. While the velocity field has been improved and better resembles the experimental data, it still is not perfect. While the initial conditions could have been improved still a perfect fit on the experimental data will not be achieved. Using this formulation of the corrective term is arguably the same as locally tuning the model coefficients. The same model assumptions remain and will be limiting. One model limitation has been shown with the 6cm gate opening case. Here, after correcting the case, it did not converge enough anymore. When testing different initial conditions the case became unsteady with some values. The $k-\omega$ turbulence model is a steady solver. This will be a cause of errors when this model is used for unsteady cases, something that can not be corrected by this formulation. Another significant model limitation is the Boussinesq hypothesis used to model the Reynolds stress term. Here the eddy viscosity is introduced to relate the turbulence stresses to the mean flow. This introduces the assumption that turbulence is isotropic, while this is not the case. The proposed correction also does not correct this assumption.

It has been shown that with a different implementation of the corrective term in a different location in the model better results can be achieved. The limitations of the implementation presented in this research are completely dependent on the flow. If the model assumptions do not influence the flow too much the current implementation will be very much capable of correcting the simulation. The benefit of this method is the adaptability of the equations. The complete adjoint solver is almost independent of the corrective term. To change the correction to apply to a different term is rather trivial. Therefore it becomes easy to investigate the effectiveness of the different terms in the models. This can give valuable insight into how the models work and how they can be improved.

1.2 Can the continuous adjoint formulation be used on more complex and unstructured grids? A clear mesh dependency was found when running the adjoint solver on a mesh with refinement zones. The found cost function sensitivity was found to be very different from the one using a smooth mesh. It does make sense that the sensitivity in larger cells is higher than in smaller cells. However, this does change the steps taken in the optimisation process, and will maybe give a different corrective field. The regularisation term will limit the values in the bigger cells to some extent, however. Some problems might occur with the stability of the case as large regions are corrected and the correction is not smooth. So while the continuous adjoint can be applied on an unstructured grid, a smooth mesh around the region of interest works best. When the mesh is smooth in the region where sensitivity is found, finding the optimal corrective field will be easier.

2. Can experimentally obtained data from PIV be used for the field inversion paradigm? In previous research, high fidelity simulations have been used as a source of reference data. In this research, the paradigm has been extended to take velocity field data from any source. It has been shown that experimental data could be used for field inversion. A few limitations were found, however, mainly due to the limitations of the model itself. The baseline model should be capable to predict a flow field that resembles the data close enough. When the model is not able to predict certain physics, this paradigm will not be able to help that. However, when the physics of the model does resemble the physics of the flow field real experimental data can be used. It has been shown that using the experimental data gathered by Deltares the underflow weir case could be improved so that the defined cost function has been reduced by a factor of two.

Due to the adjustability of the adjoint solver, the cost function can be changed easily. The paradigm can easily be adapted to accept all types of data. Therefore, this methodology can become very valuable due to its adaptability. When for example only pressure data is known from pressure sensors, changing the equations accordingly is easy.

2.1 Is it possible to apply field inversion with courser reference data? An interpolation step is introduced in this research to map the velocity field to the data grid of the experiments. The density of data points from an experiment is often much smaller than the density of cells used in simulations. Using this implementation it is possible to apply the paradigm of field inversion using coarse reference data.

Even using a single point of data will work in this paradigm and can be used to determine a corrective field. Applications of this could be for example pressure data on airfoil surfaces.

Using coarser data has more effect on the accuracy of the end result. One can imagine that when high gradients in the flow are present, enough data points should be present to capture this.

2.2 Is it possible to apply field inversion with limited reference data? Often, when using experimental data, not only will the data grid be different, but the amount of data will be limited as well. Not everywhere in the domain there will be data available. When using PIV measurements this is true for flow close to walls. Due to reflections, it is very difficult to capture the flow field velocities close to surfaces. Also, not the complete domain used in CFD will be captured in the experiments. Often only the areas of interest will be captured.

Several experiments have been done with different windows of data, testing the capabilities of the field inversion process. The resulting cost function sensitivity, computed by the adjoint solver, gave slightly different results. This makes sense as different windows of data are available.

From the equations, it has become clear that the resulting adjoint gradient is a linear combination of the adjoint gradients when using only single points of data. The cost function is a simple sum of the squared difference between the simulation output and the given data. The adjoint gradient is thus a sum of the adjoint gradients when due to every single data point. Knowing this it makes sense that a different set of data points will give a different result.

When using a limited set of data points it is important to have data in the area of interest, often the place where the largest discrepancies are present. It does not make sense to use data to find a correction where the model is already predicting the flow field accurately.

The objective for this research has been *to improve the RANS closure modelling for predicting wall-bounded jet flows behind a weir extending on the paradigm of field inversion using experimental PIV data as reference data.* This goal can be considered achieved. In this research, the field inversion paradigm has been extended to be able to be applied to bigger and more complex problems. It has been shown that more complex grids do not have to form a problem when around the area of interest a somewhat smooth mesh is present. Experimental data has been used in order to find a correction factor for a regularly used turbulence model. When the baseline model is capable of modelling the physics, the paradigm will be able to find a correction that improves the predictive capabilities.

When compared to a very high fidelity model the performance is still somewhat worse. However, the computational cost has been shown to be lower. Also, this is the first step of the paradigm. After field inversion has been applied to several cases in which experimental data is available, the next step is to learn from the results. A correlation can be found using machine learning techniques to be able to further predict the corrective field for unseen problems. The predictive phase of the machine learning phase will be very fast. Therefore the computational cost of simulating an unseen case using the machine learning model to predict the necessary correction will be in the same order as using the baseline model.

Next training a machine learning model with the data, this paradigm can also be used to learn from experimental data. The final result showed a clear deficit in the baseline model. Too much artificial viscosity was modelled which lead to too much diffusion of velocity. By decreasing the eddy viscosity the model was able to better predict the velocities present in the domain. Now the correction has been applied to the production term of the turbulent specific dissipation rate equation. Clearly in regions behind the weir too much production occurred. However in the boundary layer with the bottom also high production occurred which did not need correcting. No data on the boundary layer was available so it is difficult to make a conclusion. This knowledge could be used to better understand how the model can be improved in the future. As well, different correction terms can be applied to get a wider picture of the model problems. This can be an efficient method to identify the problems with a model in several flow cases.

14

Recommendations

With the knowledge gained during this research, some recommendations will be made in this chapter.

Firstly, in this research, the simulations and experiments have been 2D only. No real problems are to be expected when extending the paradigm to 3D. However, the reference data used in this research has also been 2D. The data used was taken in the middle of the experimental setup. When extending this methodology to 3D it is recommended to use more planes of experimental data. The adjoint solver finds the locations of high sensitivity of the cost function with respect to the corrective field. One can imagine that when the model has corrected laterally far away from the data points, the effect will be less than when close to the plane with data. Probably the highest sensitivities will be found in a region around the highest discrepancies between the simulations and the data. In regions far away from the data, the simulation can still incorrectly predict the flow field while no correction will be found. This ties in with the discussion answering the research question about limited amounts of data. The paradigm will find a correction to lower the cost function as much as possible. For an improved simulation, reference data needs to be present in all the problematic areas.

Secondly, problems were found applying the paradigm of field inversion to meshes that are not smooth enough. In more complex simulations, it is therefore advised to have a smooth mesh around the places of interest. For example, it will not be possible to have a completely smooth mesh when simulating the aerodynamics around a race car. Refinement zones will have to be applied to get the accuracy in the important places and reduce the mesh count in less important sections of the domain. However, locally in a certain section of the domain, the mesh could be made smooth without too many problems. This paradigm could be used to locally correct the used model. When data is available from pressure sensors or from PIV measurements these could be used to find the necessary correction in a localised section of the mesh. If the mesh in the region where data is available is sufficiently smooth a usable cost function sensitivity could be found using the adjoint solver. The adjoint solver could even be run on a small part of the domain. This in turn can be used to find a local correction field to improve the accuracy of the simulations. In essence, this is already done in practices, where locally in the domain the model parameters are set to different values. With this approach, this can be done with experimental evidence.

VI

Appendices

A

Continuous adjoint of the $k-\omega$ model

In this appendix the full detailed adjoint derivation is shown.

A.1. Base model

The basis model for the derivation is the $k-\omega$ RANS model [38]. The governing equations of the model are shown below, including the applied correction term β.

For clarity in the derivation below, the velocity is given as v and the adjoint velocity as u instead of u and u_a.

$$R_i^v = \frac{\partial(v_i v_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\Big((\nu+\nu_t)\big(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\big)\Big) = 0 \tag{A.1}$$

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{A.2}$$

$$R^k = \frac{\partial(v_j k)}{\partial x_j} - \frac{\partial}{\partial x_j}\Big((\sigma_k \nu_t + \nu)\frac{\partial k}{\partial x_j}\Big) - \nu_t P + \frac{2}{3}\frac{\partial v_j}{\partial x_j}k + C_\mu \omega k = 0 \tag{A.3}$$

$$R^\omega = \frac{\partial(v_j \omega)}{\partial x_j} - \frac{\partial}{\partial x_j}\Big((\sigma_\omega \nu_t + \nu)\frac{\partial \omega}{\partial x_j}\Big) - \gamma\beta P + \frac{2}{3}\gamma\frac{\partial v_j}{\partial x_j}\omega + \alpha\omega^2 = 0 \tag{A.4}$$

Here the production term is defined as:

$$P = \big(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\big)\frac{\partial v_i}{\partial x_j} \tag{A.5}$$

The standard model coefficients are used in this research.

A.2. System of equations

The system of equations for the continuous adjoint, as discussed in 8, are shown below.

$$\frac{\delta J}{\delta \beta} = \frac{\partial J}{\partial \beta} + \int_\Omega \psi \frac{\partial \mathbf{R}}{\partial \beta} d\Omega \tag{A.6}$$

$$0 = \frac{\partial J}{\partial \mathbf{u}} + \int_\Omega \psi \frac{\partial \mathbf{R}}{\partial \mathbf{u}} d\Omega \tag{A.7}$$

The objective function J is given as

$$J = \sum_{i=0}^{N} \frac{1}{2\sigma_{m,i}^2}\big(H_{i,j}\, v_{\mathrm{CFD},j} - v_{\mathrm{PIV},i}\big)^2 + \sum_{m=0}^{M} \frac{1}{2\sigma_{\beta,m}^2}\big(\beta_m - \beta_{0,m}\big)^2 \tag{A.8}$$

Where $H_{i,j}$ is a transformation function, mapping the velocity field from the CFD space to the PIV space. The system of equations will be derived in more detail in order in the following sub sections.

A.2.1. Partial derivatives with respect to β

First equation A.6 will be written out in detail

$$\frac{\delta J}{\delta \beta} = \frac{\partial J}{\partial \beta} + \int_\Omega \psi \frac{\partial \mathbf{R}}{\partial \beta} d\Omega \tag{A.9}$$

$$\frac{\delta J}{\delta \beta} = \frac{\partial J_\Omega}{\partial \beta} + \int_\Omega u \frac{\partial R^\nu}{\partial \beta} d\Omega + \int_\Omega q \frac{\partial R^p}{\partial \beta} d\Omega + \int_\Omega k_a \frac{\partial R^k}{\partial \beta} d\Omega + \int_\Omega \omega_a \frac{\partial R^\omega}{\partial \beta} d\Omega \tag{A.10}$$

The terms will be worked out in detail in order. However, the only terms that include β are the cost function and the ω-equation.

Cost function derivative

First we start with the derivative of the cost function:

$$J = \sum_{i=0}^{N} \frac{1}{2\sigma_{m,i}^2} \left(H_{i,j} v_{\mathrm{CFD},j} - v_{\mathrm{PIV},i}\right)^2 + \sum_{m=0}^{M} \frac{1}{2\sigma_{\beta,m}^2} \left(\beta_m - \beta_{0,m}\right)^2 \tag{A.11}$$

$$\frac{\partial J_\Omega}{\partial \beta} = \frac{1}{\sigma_\beta^2} (\beta - \beta_0) \tag{A.12}$$

ω-equation derivative

The governing equation is given below

$$R^\omega = \frac{\partial (v_j \omega)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu) \frac{\partial \omega}{\partial x_j}\right) - \gamma \beta P + \frac{2}{3}\gamma \frac{\partial v_j}{\partial x_j}\omega + \alpha \omega^2 = 0 \tag{A.13}$$

This can easily be derived to get

$$\int_\Omega \omega_a \frac{\partial R^\omega}{\partial \beta} d\Omega = -\int_\Omega \omega_a \gamma P d\Omega \tag{A.14}$$

Resulting equation

The resulting equation is thus:

$$\frac{\delta J}{\delta \beta} = \frac{1}{\sigma_\beta^2} (\beta - \beta_0) - \int_\Omega \omega_a \gamma P d\Omega \tag{A.15}$$

A.2.2. Partial derivatives with respect to $\vec{\mathbf{u}}$

Now equation A.7 will be worked out in detail, again in order. Four new governing equations will be formed, called the adjoint governing equations. The four equations are formed as shown below:

$$0 = \frac{\partial J}{\partial \nu} + \int_\Omega \Psi_i \frac{\partial \mathbf{R}_i}{\partial \nu} d\Omega \tag{A.16}$$

$$0 = \frac{\partial J}{\partial p} + \int_\Omega \Psi_i \frac{\partial \mathbf{R}_i}{\partial p} d\Omega \tag{A.17}$$

$$0 = \frac{\partial J}{\partial k} + \int_\Omega \Psi_i \frac{\partial \mathbf{R}_i}{\partial k}) d\Omega \tag{A.18}$$

$$0 = \frac{\partial J}{\partial \omega} + \int_\Omega \Psi_i \frac{\partial \mathbf{R}_i}{\partial \omega} d\Omega \tag{A.19}$$

Cost function derivative

First we start again with the cost function. Only the derivative with respect to the velocity field is needed, as the cost function only includes this variable.

$$J = \sum_{i=0}^{N} \frac{1}{2\sigma_{m,i}^2} \left(H_{i,j} v_{\mathrm{CFD},j} - v_{\mathrm{PIV},i}\right)^2 + \sum_{m=0}^{M} \frac{1}{2\sigma_{\beta,m}^2} \left(\beta_m - \beta_{0,m}\right)^2 \tag{A.20}$$

$$\int_\Omega \frac{\partial J_\Omega}{\partial v} d\Omega = \frac{1}{V_\Omega} \int_\Omega \frac{1}{\sigma_m^2} H^T \left(H v_{\mathrm{CFD}} - v_{\mathrm{PIV}}\right) d\Omega \tag{A.21}$$

Adjoint momentum equation

The **momentum equation** is given by:

$$R_i^v = \frac{\partial (v_i v_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right) = 0 \tag{A.22}$$

$$\int_\Omega u_i \frac{\partial R^{v_i}}{\partial v} d\Omega = \int_\Omega u_i \frac{\partial v_i}{\partial x_j} + \underbrace{u_i v_j \frac{\partial}{\partial x_j}}_{\mathbf{I}} + u_i \cancelto{\mathrm{div}(v)=0}{\frac{\partial v_j}{\partial x_j}} + \underbrace{u_i v_i \frac{\partial}{\partial x_j}}_{\mathbf{II}} - \underbrace{u_i \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial}{\partial x_j} + \frac{\partial}{\partial x_i}\right)\right)}_{\mathbf{III}} d\Omega \tag{A.23}$$

$$\begin{aligned} \mathbf{I} &= \int_\Omega u_i v_j \frac{\partial}{\partial x_j} d\Omega \\ &= \int_\Gamma u_i v_j n_j d\Gamma - \int_\Omega \frac{\partial (u_i v_j)}{\partial x_j} d\Omega \end{aligned} \tag{A.24}$$

$$\begin{aligned} \mathbf{II} &= \int_\Omega u_i v_i \frac{\partial}{\partial x_j} d\Omega \\ &= \int_\Gamma u_i v_i n_j d\Gamma - \int_\Omega \frac{\partial (u_i v_i)}{\partial x_j} d\Omega \end{aligned} \tag{A.25}$$

$$\begin{aligned} \mathbf{III} &= -\int_\Omega u_i \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial}{\partial x_j} + \frac{\partial}{\partial x_i}\right)\right) d\Omega \\ &= -\int_\Gamma u_i n_j (\nu + \nu_t)\left(\frac{\partial}{\partial x_j} + \frac{\partial}{\partial x_i}\right) d\Gamma + \int_\Omega \frac{\partial u_i}{\partial x_j} (\nu + \nu_t)\left(\frac{\partial}{\partial x_j} + \frac{\partial}{\partial x_i}\right) d\Omega \\ &= -\int_\Gamma u_i n_j (\nu + \nu_t)\left(\frac{\partial}{\partial x_j} + \frac{\partial}{\partial x_i}\right) d\Gamma + \int_\Gamma \frac{\partial u_i}{\partial x_j} (\nu + \nu_t)(n_j + n_i) d\Gamma - \int_\Omega \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) d\Omega \end{aligned} \tag{A.26}$$

So we get:

$$\begin{aligned} \int_\Omega u_i \frac{\partial R^{v_i}}{\partial v} d\Omega = \int_\Omega \Bigg[&- \frac{\partial (u_i v_j)}{\partial x_j} - \frac{\partial u_i}{\partial x_j} v_i - \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right)\Bigg] d\Omega \\ &+ \int_\Gamma \left[u_i v_j n_j + u_i v_i n_j - u_i n_j (\nu + \nu_t)\left(\frac{\partial}{\partial x_j} + \frac{\partial}{\partial x_i}\right) + \frac{\partial u_i}{\partial x_j} (\nu + \nu_t)(n_j + n_i) \right] d\Gamma \end{aligned} \tag{A.27}$$

Now the **continuity equation**:

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{A.28}$$

$$\int_\Omega q \frac{\partial R^p}{\partial v_i} d\Omega = -\int_\Omega q \frac{\partial}{\partial x_j} d\Omega \tag{A.29}$$

$$\int_\Omega q \frac{\partial R^p}{\partial v_i} d\Omega = \int_\Omega \frac{\partial q}{\partial x_j} d\Omega - \int_\Gamma q n_j d\Gamma \tag{A.30}$$

The **k-equation**:

$$R^k = \frac{\partial(v_j k)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_k \nu_t + \nu)\frac{\partial k}{\partial x_j}\right) - \nu_t P + \frac{2}{3}\frac{\partial v_j}{\partial x_j}k + C_\mu \omega k = 0 \tag{A.31}$$

$$\int_\Omega k_a \frac{\partial R^k}{\partial v_i} d\Omega = \int_\Omega \Bigg[k_a \frac{\partial k}{\partial x_j} + \underbrace{k_a k \frac{\partial}{\partial x_j}}_{\mathbf{I}} - \underbrace{2k_a \nu_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\frac{\partial}{\partial x_j}}_{\mathbf{II}} + \underbrace{\frac{2}{3}k_a k \frac{\partial}{\partial x_j}}_{\mathbf{III}} \Bigg] d\Omega \tag{A.32}$$

Using:

$$P = \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\frac{\partial v_i}{\partial x_j} \tag{A.33}$$

$$\frac{\partial P}{\partial v_i} = 2\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\frac{\partial}{\partial x_j} \tag{A.34}$$

Working out the terms:

$$\begin{aligned} \mathbf{I} &= \int_\Omega k_a k \frac{\partial}{\partial x_j} d\Omega \\ &= -\int_\Omega \frac{\partial(k_a k)}{\partial x_j} d\Omega + \int_\Gamma k_a k n_j d\Gamma \end{aligned} \tag{A.35}$$

$$\begin{aligned} \mathbf{II} &= -\int_\Omega 2k_a \nu_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\frac{\partial}{\partial x_j} d\Omega \\ &= +\int_\Omega \frac{\partial}{\partial x_j}\left(2k_a \nu_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right) d\Omega - \int_\Gamma 2k_a \nu_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) n_j d\Gamma \end{aligned} \tag{A.36}$$

$$\begin{aligned} \mathbf{III} &= \int_\Omega \frac{2}{3} k_a k \frac{\partial}{\partial x_j} d\Omega \\ &= -\int_\Omega \frac{2}{3}\frac{\partial(k_a k)}{\partial x_j} d\Omega + \int_\Gamma \frac{2}{3} k_a k n_j d\Gamma \end{aligned} \tag{A.37}$$

So we get:

$$\begin{aligned} \int_\Omega k_a \frac{\partial R^k}{\partial v_i} d\Omega = \int_\Omega \left[-k\frac{\partial k_a}{\partial x_j} - \frac{2}{3}\frac{\partial(k_a k)}{\partial x_j} + \frac{\partial}{\partial x_j}\left(2k_a \nu_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right)\right] d\Omega \\ + \int_\Gamma \left[\frac{5}{3} k_a k n_j - 2k_a \nu_t \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) n_j\right] d\Gamma \end{aligned} \tag{A.38}$$

The ***ω*-equation**
Again using equation A.34.

$$R^{\omega} = \frac{\partial(v_j\omega)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega}{\partial x_j}\right) - \gamma\beta P + \frac{2}{3}\gamma\frac{\partial v_j}{\partial x_j}\omega + \alpha\omega^2 = 0 \tag{A.39}$$

$$\int_\Omega \omega_a \frac{\partial R^{\omega}}{\partial v} d\Omega = \int_\Omega \Bigg[\omega_a \frac{\partial \omega}{\partial x_j} + \underbrace{\omega_a \omega \frac{\partial}{\partial x_j}}_{\mathbf{I}} \underbrace{- \omega_a 2\gamma\beta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\frac{\partial}{\partial x_j}}_{\mathbf{II}} + \underbrace{\omega_a \frac{2}{3}\gamma\omega\frac{\partial}{\partial x_j}}_{\mathbf{III}}\Bigg] d\Omega \tag{A.40}$$

Working out the terms:

$$\begin{aligned}\mathbf{I} &= \int_\Omega \omega_a \omega \frac{\partial}{\partial x_j} d\Omega \\ &= -\int_\Omega \frac{\partial(\omega_a\omega)}{\partial x_j} d\Omega + \int_\Gamma \omega_a \omega n_j d\Gamma\end{aligned} \tag{A.41}$$

$$\begin{aligned}\mathbf{II} &= -\int_\Omega \omega_a 2\gamma\beta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\frac{\partial}{\partial x_j} d\Omega \\ &= \int_\Omega \frac{\partial}{\partial x_j}\left(2\omega_a\gamma\beta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right) d\Omega - \int_\Gamma 2\omega_a\gamma\beta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) n_j d\Gamma\end{aligned} \tag{A.42}$$

$$\begin{aligned}\mathbf{III} &= \int_\Omega \omega_a \frac{2}{3}\gamma\omega\frac{\partial}{\partial x_j} d\Omega \\ &= -\int_\Omega \frac{2}{3}\gamma\frac{\partial(\omega_a\omega)}{\partial x_j} d\Omega + \int_\Gamma \frac{2}{3}\gamma\omega_a\omega n_j d\Gamma\end{aligned} \tag{A.43}$$

So we get:

$$\begin{aligned}\int_\Omega \omega_a \frac{\partial R^{\omega}}{\partial v_i} d\Omega = \int_\Omega \Bigg[&-\omega\frac{\partial \omega_a}{\partial x_j} - \frac{2}{3}\gamma\frac{\partial(\omega_a\omega)}{\partial x_j} + \frac{\partial}{\partial x_j}\left(2\omega_a\gamma\beta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right)\Bigg] d\Omega \\ &+ \int_\Gamma \left[\left(1 + \frac{2}{3}\gamma\right)\omega_a\omega n_j - 2\omega_a\gamma\beta\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)n_j - u_i n_j(\nu + \nu_t)\left(\frac{\partial}{\partial x_j} + \frac{\partial}{\partial x_i}\right)\right] d\Gamma\end{aligned} \tag{A.44}$$

Now we can combine all the volume integral terms to get R_i^u and all the boundary integral terms to get D_i^u and P_i^u which make up the final governing and boundary condition equations of the adjoint momentum equation.

$$\begin{aligned}R_i^u = \frac{\partial J}{\partial v_i} - \frac{\partial(u_i v_j)}{\partial x_j} - \frac{\partial u_i}{\partial x_j}v_i - \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) + \frac{\partial q}{\partial x_j} - k\frac{\partial k_a}{\partial x_j} - \omega\frac{\partial \omega_a}{\partial x_j} \\ - \frac{2}{3}\frac{\partial(k_a k)}{\partial x_j} - \frac{2}{3}\gamma\frac{\partial(\omega_a\omega)}{\partial x_j} + 2\frac{\partial}{\partial x_j}\left((k_a\nu_t + \omega_a\gamma\beta)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right)\end{aligned} \tag{A.45}$$

$$\begin{aligned}D_i^u = u_i v_j n_j + u_i v_i n_j + (\nu + \nu_t)n_j\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - q n_j + \frac{5}{3}k_a k n_j + \left(1 + \frac{2}{3}\gamma\right)\omega_a\omega n_j \\ - 2k_a\nu_t n_j\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - 2\omega_a\gamma\beta n_j\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\end{aligned} \tag{A.46}$$

$$P_i^u = -(\nu + \nu_t)u_i n_j - (\nu + \nu_t)u_j n_i \tag{A.47}$$

Adjoint continuity equation

The **momentum equation** is given as:

$$R_i^v = \frac{\partial(v_i v_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\Big((\nu+\nu_t)\big(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\big)\Big) = 0 \tag{A.48}$$

$$\int_\Omega u_i \frac{\partial R_i^v}{\partial p} d\Omega = \int_\Omega u_i \frac{\partial}{\partial x_i} d\Omega \tag{A.49}$$

$$\int_\Omega u_i \frac{\partial R_i^v}{\partial p} d\Omega = -\int_\Omega \frac{\partial u_i}{\partial x_i} d\Omega + \int_\Gamma u_i n_i d\Gamma \tag{A.50}$$

The **continuity equation** is given as:

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{A.51}$$

$$\int_\Omega q \frac{\partial R^p}{\partial p} d\Omega = 0 \tag{A.52}$$

The **k-equation** is given as:

$$R^k = \frac{\partial(v_j k)}{\partial x_j} - \frac{\partial}{\partial x_j}\Big((\sigma_k \nu_t + \nu)\frac{\partial k}{\partial x_j}\Big) - \nu_t P + \frac{2}{3}\frac{\partial v_j}{\partial x_j} k + C_\mu \omega k = 0 \tag{A.53}$$

$$\int_\Omega k_a \frac{\partial R^k}{\partial p} d\Omega = 0 \tag{A.54}$$

And finally, the $\boldsymbol{\omega}$**-equation** is given as:

$$R^\omega = \frac{\partial(v_j \omega)}{\partial x_j} - \frac{\partial}{\partial x_j}\Big((\sigma_\omega \nu_t + \nu)\frac{\partial \omega}{\partial x_j}\Big) - \gamma\beta P + \frac{2}{3}\gamma\frac{\partial v_j}{\partial x_j}\omega + \alpha\omega^2 = 0 \tag{A.55}$$

$$\int_\Omega \omega_a \frac{\partial R^\omega}{\partial p} d\Omega = 0 \tag{A.56}$$

Gathering all the terms in the volume and boundary integrals to for R^q and D^q we get:

$$R^q = -\frac{\partial u_i}{\partial x_i} = 0 \tag{A.57}$$

$$D^q = u_i n_i = 0 \tag{A.58}$$

Adjoint k-equation

The derivation from the **momentum equation** goes as follows, using $\nu_t = \frac{k}{\omega}$:

$$R_i^v = \frac{\partial(v_i v_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\Big((\nu+\nu_t)\big(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\big)\Big) = 0 \tag{A.59}$$

$$\int_\Omega u_i \frac{\partial R_i^v}{\partial k} d\Omega = -\int_\Omega u_i \frac{\partial}{\partial x_j}\left(\frac{1}{\omega}\Big(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\Big)\right) d\Omega \tag{A.60}$$

$$\int_\Omega u_i \frac{\partial R_i^v}{\partial k} d\Omega = \int_\Omega \frac{\partial u_i}{\partial x_j}\frac{1}{\omega}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right) d\Omega - \int_\Gamma u_i n_j \frac{1}{\omega}\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right) d\Gamma \tag{A.61}$$

The derivation from the **continuity equation** goes as follows:

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{A.62}$$

$$\int_\Omega q \frac{\partial R^p}{\partial k} d\Omega = 0 \tag{A.63}$$

The derivation from the **k-equation** goes as follows:

$$R^k = \frac{\partial(v_j k)}{\partial x_j} - \frac{\partial}{\partial x_j}\Big((\sigma_k \nu_t + \nu)\frac{\partial k}{\partial x_j}\Big) - \nu_t P + \frac{2}{3}\frac{\partial v_j}{\partial x_j}k + C_\mu \omega k = 0 \tag{A.64}$$

$$\int_\Omega k_a \frac{\partial R^k}{\partial k} d\Omega = \int_\Omega \Bigg[\underbrace{k_a v_j \frac{\partial}{\partial x_j}}_{\mathbf{I}} + k_a \frac{\partial v_j}{\partial x_j} \underbrace{- k_a \frac{\partial}{\partial x_j}\Big((\sigma_k \nu_t + \nu)\frac{\partial}{\partial x_j}\Big)}_{\mathbf{II}} \underbrace{- k_a \frac{\partial}{\partial x_j}\Big(\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j}\Big)}_{\mathbf{III}} - \frac{k_a}{\omega}P + k_a \frac{2}{3}\frac{\partial v_j}{\partial x_j} + k_a C_\mu \omega \Bigg] d\Omega \tag{A.65}$$

Working out the terms:

$$\begin{aligned} \mathbf{I} &= \int_\Omega k_a v_j \frac{\partial}{\partial x_j} d\Omega \\ &= -\int_\Omega \frac{\partial(k_a v_j)}{\partial x_j} d\Omega + \int_\Gamma k_a v_j n_j d\Gamma \end{aligned} \tag{A.66}$$

$$\begin{aligned} \mathbf{II} &= -\int_\Omega k_a \frac{\partial}{\partial x_j}\Big((\sigma_k \nu_t + \nu)\frac{\partial}{\partial x_j}\Big) d\Omega \\ &= \int_\Omega \frac{\partial k_a}{\partial x_j}(\sigma_k \nu_t + \nu)\frac{\partial}{\partial x_j} d\Omega - \int_\Gamma k_a n_j (\sigma_k \nu_t + \nu)\frac{\partial}{\partial x_j} d\Gamma \\ &= -\int_\Omega \frac{\partial}{\partial x_j}\Big((\sigma_k \nu_t + \nu)\frac{\partial k_a}{\partial x_j}\Big) d\Omega + \int_\Gamma \frac{\partial k_a}{\partial x_j}(\sigma_k \nu_t + \nu) n_j d\Gamma - \int_\Gamma k_a n_j (\sigma_k \nu_t + \nu)\frac{\partial}{\partial x_j} d\Gamma \end{aligned} \tag{A.67}$$

$$\begin{aligned} \mathbf{III} &= -\int_\Omega k_a \frac{\partial}{\partial x_j}\Big(\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j}\Big) d\Omega \\ &= \int_\Omega \frac{\partial k_a}{\partial x_j}\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} d\Omega - \int_\Gamma k_a n_j \frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} d\Gamma \end{aligned} \tag{A.68}$$

So we get:

$$\begin{aligned} \int_\Omega k_a \frac{\partial R^k}{\partial k} d\Omega = \int_\Omega \Bigg[&-\frac{\partial(k_a v_j)}{\partial x_j} + k_a \frac{5}{3}\frac{\partial v_j}{\partial x_j} - \frac{\partial}{\partial x_j}\Big((\sigma_k \nu_t + \nu)\frac{\partial k_a}{\partial x_j}\Big) + \frac{\partial k_a}{\partial x_j}\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} - \frac{k_a}{\omega}P + k_a C_\mu \omega \Bigg] d\Omega \\ &+ \int_\Gamma \Bigg[k_a v_j n_j + \frac{\partial k_a}{\partial x_j}(\sigma_k \nu_t + \nu) n_j - k_a n_j (\sigma_k \nu_t + \nu)\frac{\partial}{\partial x_j} - k_a n_j \frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} \Bigg] d\Gamma \end{aligned} \tag{A.69}$$

The derivation from the **ω-equation** goes as follows:

$$R^{\omega} = \frac{\partial(v_j\omega)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega}{\partial x_j}\right) - \gamma\beta P + \frac{2}{3}\gamma\frac{\partial v_j}{\partial x_j}\omega + \alpha\omega^2 = 0 \tag{A.70}$$

$$\int_\Omega \omega_a \frac{\partial R^{\omega}}{\partial k} d\Omega = -\int_\Omega \omega_a \frac{\partial}{\partial x_j}\left(\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j}\right) d\Omega \tag{A.71}$$

$$\int_\Omega \omega_a \frac{\partial R^{\omega}}{\partial k} d\Omega = \int_\Omega \frac{\partial \omega_a}{\partial x_j}\left(\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j}\right) d\Omega - \int_\Gamma \omega_a n_j \left(\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j}\right) d\Gamma \tag{A.72}$$

Gathering all the terms in the volume and boundary integrals to for R^{k_a}, D^{k_a} and P^{k_a} we get:

$$\begin{aligned} R^{k_a} = -\frac{\partial(k_a v_j)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_k \nu_t + \nu)\frac{\partial k_a}{\partial x_j}\right) + \frac{\partial k_a}{\partial x_j}\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} + \frac{\partial \omega_a}{\partial x_j}\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j} \\ + \frac{1}{\omega}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + k_a\frac{5}{3}\frac{\partial v_j}{\partial x_j} - \frac{k_a}{\omega}P + k_a C_\mu \omega = 0 \end{aligned} \tag{A.73}$$

$$D^{k_a} = -u_i n_j \frac{1}{\omega}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + k_a v_j n_j + \frac{\partial k_a}{\partial x_j}(\sigma_k \nu_t + \nu) n_j - k_a n_j \frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} - \omega_a n_j \left(\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j}\right) \tag{A.74}$$

$$P^{k_a} = -k_a n_j (\sigma_k \nu_t + \nu) \tag{A.75}$$

Adjoint ω-equation

The derivation from the **momentum equation** goes as follows, using $\nu_t = \frac{k}{\omega}$:

$$R_i^v = \frac{\partial(v_i v_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}\Big((\nu+\nu_t)\big(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\big)\Big) = 0 \tag{A.76}$$

$$\int_\Omega u_i \frac{\partial R_i^v}{\partial \omega} d\Omega = \int_\Omega u_i \frac{\partial}{\partial x_j}\left(\frac{k}{\omega^2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right) d\Omega \tag{A.77}$$

$$\int_\Omega u_i \frac{\partial R_i^v}{\partial \omega} d\Omega = -\int_\Omega \frac{\partial u_i}{\partial x_j}\frac{k}{\omega^2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) d\Omega + \int_\Gamma u_i n_j \frac{k}{\omega^2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) d\Gamma \tag{A.78}$$

The derivation from the **continuity equation** goes as follows:

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{A.79}$$

$$\int_\Omega q \frac{\partial R^p}{\partial \omega} d\Omega = 0 \tag{A.80}$$

The derivation from the **k-equation** goes as follows:

$$R^k = \frac{\partial(v_j k)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_k \nu_t + \nu)\frac{\partial k}{\partial x_j}\right) - \nu_t P + \frac{2}{3}\frac{\partial v_j}{\partial x_j} k + C_\mu \omega k = 0 \tag{A.81}$$

$$\int_\Omega k_a \frac{\partial R^k}{\partial \omega} d\Omega = \int_\Omega \left[k_a \frac{\partial}{\partial x_j}\left(\sigma_k \frac{k}{\omega^2}\frac{\partial k}{\partial x_j}\right) + k_a \frac{k}{\omega^2} P + C_\mu k_a k \right] d\Omega \tag{A.82}$$

$$\int_\Omega k_a \frac{\partial R^k}{\partial \omega} d\Omega = \int_\Omega \left[-\frac{\partial k_a}{\partial x_j}\sigma_k \frac{k}{\omega^2}\frac{\partial k}{\partial x_j} + k_a \frac{k}{\omega^2} P + C_\mu k_a k \right] d\Omega + \int_\Gamma k_a n_j \sigma_k \frac{k}{\omega^2}\frac{\partial k}{\partial x_j} d\Gamma \tag{A.83}$$

The derivation from the ***ω*-equation** goes as follows:

$$R^\omega = \frac{\partial(v_j \omega)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega}{\partial x_j}\right) - \gamma\beta P + \frac{2}{3}\gamma\frac{\partial v_j}{\partial x_j}\omega + \alpha\omega^2 = 0 \tag{A.84}$$

$$\int_\Omega \omega_a \frac{\partial R^\omega}{\partial \omega} d\Omega = \int_\Omega \Bigg[\underbrace{\omega_a v_j \frac{\partial}{\partial x_j}}_{\text{I}} + \omega_a \frac{\partial v_j}{\partial x_j} \underbrace{- \omega_a \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial}{\partial x_j}\right)}_{\text{II}} \underbrace{+ \omega_a \frac{\partial}{\partial x_j}\left(\sigma_\omega \frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j}\right)}_{\text{III}} + \omega_a \frac{2}{3}\gamma\frac{\partial v_j}{\partial x_j} + 2\omega_a \alpha\omega \Bigg] d\Omega \tag{A.85}$$

Working out the terms:

$$\begin{aligned}\mathbf{I} &= \int_\Omega \omega_a v_j \frac{\partial}{\partial x_j} d\Omega \\ &= -\int_\Omega \frac{\partial(\omega_a v_j)}{\partial x_j} d\Omega + \int_\Gamma \omega_a v_j n_j d\Gamma\end{aligned} \tag{A.86}$$

$$\begin{aligned}\mathbf{II} &= -\int_\Omega \omega_a \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial}{\partial x_j}\right) d\Omega \\ &= \int_\Omega \frac{\partial \omega_a}{\partial x_j}(\sigma_\omega \nu_t + \nu)\frac{\partial}{\partial x_j} d\Omega - \int_\Gamma \omega_a n_j (\sigma_\omega \nu_t + \nu)\frac{\partial}{\partial x_j} d\Gamma \\ &= -\int_\Omega \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega_a}{\partial x_j}\right) d\Omega + \int_\Gamma \frac{\partial \omega_a}{\partial x_j}(\sigma_\omega \nu_t + \nu) n_j d\Gamma - \int_\Gamma \omega_a n_j (\sigma_\omega \nu_t + \nu)\frac{\partial}{\partial x_j} d\Gamma\end{aligned} \tag{A.87}$$

$$\begin{aligned}\mathbf{III} &= \int_\Omega \omega_a \frac{\partial}{\partial x_j}\left(\frac{\sigma_\omega k}{\omega^2}\frac{\partial \omega}{\partial x_j}\right) d\Omega \\ &= -\int_\Omega \frac{\partial \omega_a}{\partial x_j}\sigma_\omega \frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} d\Omega + \int_\Gamma \omega_a n_j \sigma_\omega \frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} d\Gamma\end{aligned} \tag{A.88}$$

So we get:

$$\begin{aligned}\int_\Omega \omega_a \frac{\partial R^\omega}{\partial k} d\Omega = \int_\Omega \Bigg[&-\frac{\partial(\omega_a v_j)}{\partial x_j} + \left(1+\gamma\frac{2}{3}\right)\omega_a \frac{\partial v_j}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega_a}{\partial x_j}\right) - \frac{\partial \omega_a}{\partial x_j}\sigma_\omega \frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} + 2\omega_a \alpha\omega \Bigg] d\Omega \\ &+ \int_\Gamma \left[\omega_a v_j n_j + \frac{\partial \omega_a}{\partial x_j}(\sigma_\omega \nu_t + \nu) n_j - \omega_a n_j (\sigma_\omega \nu_t + \nu)\frac{\partial}{\partial x_j} + \omega_a n_j \sigma_\omega \frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} \right] d\Gamma\end{aligned} \tag{A.89}$$

Gathering all the terms in the volume and boundary integrals to for R^{ω_a}, D^{ω_a} and P^{ω_a} we get:

$$\begin{aligned}R^{\omega_a} = &-\frac{\partial(\omega_a v_j)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega_a}{\partial x_j}\right) - \frac{\partial \omega_a}{\partial x_j}\sigma_\omega \frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} - \frac{\partial k_a}{\partial x_j}\sigma_k \frac{k}{\omega^2}\frac{\partial k}{\partial x_j} + k_a \frac{k}{\omega^2} P \\ &- \frac{\partial u_i}{\partial x_j}\frac{k}{\omega^2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \left(1+\gamma\frac{2}{3}\right)\omega_a \frac{\partial v_j}{\partial x_j} + 2\omega_a \alpha\omega + C_\mu k_a k = 0\end{aligned} \tag{A.90}$$

$$D^{\omega_a} = \omega_a v_j n_j + \frac{\partial \omega_a}{\partial x_j}(\sigma_\omega \nu_t + \nu) n_j + \omega_a n_j \sigma_\omega \frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} + u_i n_j \frac{k}{\omega^2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + k_a n_j \sigma_k \frac{k}{\omega^2}\frac{\partial k}{\partial x_j} \tag{A.91}$$

$$P^{\omega_a} = -\omega_a n_j (\sigma_\omega \nu_t + \nu) \tag{A.92}$$

Summary

So to summarise all the equations are shown below:

$$R_i^u = \frac{\partial J}{\partial v_i} - \frac{\partial (u_i v_j)}{\partial x_j} - \frac{\partial u_i}{\partial x_j} v_i - \frac{\partial}{\partial x_j}\left((\nu + \nu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right) + \frac{\partial q}{\partial x_j} - k\frac{\partial k_a}{\partial x_j} - \omega\frac{\partial \omega_a}{\partial x_j} - \frac{2}{3}\frac{\partial (k_a k)}{\partial x_j} - \frac{2}{3}\gamma\frac{\partial(\omega_a \omega)}{\partial x_j} + 2\frac{\partial}{\partial x_j}\left((k_a \nu_t + \omega_a \gamma\beta)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)\right) \quad \text{(A.93)}$$

$$R^q = -\frac{\partial u_i}{\partial x_i} = 0 \quad \text{(A.94)}$$

$$R^{k_a} = -\frac{\partial (k_a v_j)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_k \nu_t + \nu)\frac{\partial k_a}{\partial x_j}\right) + \frac{\partial k_a}{\partial x_j}\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} + \frac{\partial \omega_a}{\partial x_j}\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j} + \frac{1}{\omega}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \frac{k_a}{\omega}P + k_a\frac{5}{3}\frac{\partial v_j}{\partial x_j} + k_a C_\mu \omega = 0 \quad \text{(A.95)}$$

$$R^{\omega_a} = -\frac{\partial (\omega_a v_j)}{\partial x_j} - \frac{\partial}{\partial x_j}\left((\sigma_\omega \nu_t + \nu)\frac{\partial \omega_a}{\partial x_j}\right) - \frac{\partial \omega_a}{\partial x_j}\sigma_\omega\frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} - \frac{\partial k_a}{\partial x_j}\sigma_k\frac{k}{\omega^2}\frac{\partial k}{\partial x_j} - \frac{k}{\omega^2}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + k_a\frac{k}{\omega^2}P + \left(1 + \gamma\frac{2}{3}\right)\omega_a\frac{\partial v_j}{\partial x_j} + 2\omega_a\alpha\omega + C_\mu k_a k = 0 \quad \text{(A.96)}$$

$$D_j^u = u_i v_j n_j + u_i v_i n_j + (\nu + \nu_t) n_j\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - q n_j + \frac{5}{3}k_a k n_j + \left(1 + \frac{2}{3}\gamma\right)\omega_a \omega n_j - 2k_a \nu_t n_j\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - 2\omega_a \gamma\beta n_j\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \quad \text{(A.97)}$$

$$D^q = u_i n_i \quad \text{(A.98)}$$

$$D^{k_a} = -u_i n_j\frac{1}{\omega}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + k_a v_j n_j + \frac{\partial k_a}{\partial x_j}(\sigma_k \nu_t + \nu) n_j - k_a n_j\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j} - \omega_a n_j\left(\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j}\right) \quad \text{(A.99)}$$

$$D^{\omega_a} = \omega_a v_j n_j + \frac{\partial \omega_a}{\partial x_j}(\sigma_\omega \nu_t + \nu) n_j + \omega_a n_j \sigma_\omega\frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j} + u_i n_j\frac{k}{\omega^2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + k_a n_j \sigma_k\frac{k}{\omega^2}\frac{\partial k}{\partial x_j} \quad \text{(A.100)}$$

$$P_i^u = -(\nu + \nu_t) u_i n_j - (\nu + \nu_t) u_j n_i \quad \text{(A.101)}$$

$$P^{k_a} = -k_a n_j(\sigma_k \nu_t + \nu) \quad \text{(A.102)}$$

$$P^{\omega_a} = -\omega_a n_j(\sigma_\omega \nu_t + \nu) \quad \text{(A.103)}$$

A.3. Boundary Conditions

In this section a detailed overview is given of the derivation for the boundary conditions.

In deriving the adjoint methodology all the terms multiplied with the derivative of the primal variables with respect to β were gathered and set to zero. The reason being as this derivative is difficult to compute. However in some cases at the boundaries this derivative is set to zero. This can be used to derive the boundary conditions for the adjoint variables. The respective boundary integral doesn't have to be zero anymore for the complete term to drop out.

A.3.1. Inlet and no-slip walls

On the inlet and no-slip walls Dirichlet boundary conditions are used for the primal variables, except for the pressure, for which a Neumann boundary condition is used. This means:

$$v_i = f_v(z) \tag{A.104}$$
$$k = f_k(z) \tag{A.105}$$
$$\omega = f_\omega(z) \tag{A.106}$$
$$\frac{\partial p}{\partial x_j} = 0 \tag{A.107}$$

With the functions f for the values at the inlet boundary. This means that with a change in β the boundary primal values and pressure derivative should not change. So we can say:

$$\frac{\partial v_i}{\partial \beta} = 0 \tag{A.108}$$
$$\frac{\partial k}{\partial \beta} = 0 \tag{A.109}$$
$$\frac{\partial \omega}{\partial \beta} = 0 \tag{A.110}$$
$$\frac{\partial}{\partial x_j}\frac{\partial p}{\partial \beta} = 0 \tag{A.111}$$

This means that D_i^v, D^k, D^ω can be left out from this derivation.
So we have:

$$D_i^q = 0, \quad P_{ij}^u = 0, \quad P_j^k = 0 \quad \text{and} \quad P_j^\omega = 0 \tag{A.112}$$

From this follows:

$$D_i^q = 0 = u_i n_i = u_n \tag{A.113}$$

$$\begin{aligned} P_{ij}^u = 0 &= -(\nu+\nu_t)u_i n_j - (\nu+\nu_t)u_j n_i \\ 0 &= -u_i n_j - u_j n_i \\ 0 &= -u_n \frac{n_j}{n_i} - u_j n_i \\ 0 &= u_j n_i \\ 0 &= u_j \end{aligned} \tag{A.114}$$

$$\begin{aligned} P^{k_a} = 0 &= -k_a n_j(\sigma_k \nu_t + \nu) \\ 0 &= k_a \end{aligned} \tag{A.115}$$

$$\begin{aligned} P^{\omega_a} = 0 &= -\omega_a n_j(\sigma_\omega \nu_t + \nu) \\ 0 &= \omega_a \end{aligned} \tag{A.116}$$

A.3.2. Outlet

At the outlet Neumann boundary conditions are applied on all the primal velocity in turbulence variables. A Dirichlet boundary condition is applied on the primal pressure.

$$\frac{\partial}{\partial x_j}\frac{\partial (v_i)}{\partial \beta} = 0, \quad \frac{\partial p}{\partial \beta} = 0, \quad \frac{\partial}{\partial x_j}\frac{\partial k}{\partial \beta} = 0, \quad \text{and} \quad \frac{\partial}{\partial x_j}\frac{\partial \omega}{\partial \beta} = 0 \tag{A.117}$$

Which means that we have to set the remaining equations to zero:

$$D_j^u = 0, \quad D_j^{k_a} = 0, \quad \text{and} \quad D_j^{\omega_a} = 0 \tag{A.118}$$

First we solve D_{ij}^u for the adjoint pressure q. If we take the normal component:

$$D_i^u n_i = 0 = u_i v_j n_j n_i + u_j v_j n_i n_i + (\nu + \nu_t) n_j n_i \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - q n_i n_i + \frac{5}{3} k_a k n_i n_i + \left(1 + \frac{2}{3}\gamma\right)\omega_a \omega n_i n_i - 2 k_a \nu_t n_j n_i \left(\frac{\partial v_j}{\partial x_i}\right) - 2\omega_a \gamma \beta n_j n_i \left(\frac{\partial v_j}{\partial x_i}\right) \tag{A.119}$$

$$q = u_n v_n + u_j v_j + 2(\nu + \nu_t) n_j \frac{\partial u_n}{\partial x_j} + \frac{5}{3} k_a k + \left(1 + \frac{2}{3}\gamma\right)\omega_a \omega \tag{A.120}$$

Now we solve D_{ij}^u for the adjoint velocity u. We do this by taking the tangential component:

$$D_i^u t_i = 0 = u_i v_j n_j t_i + u_j v_j n_i t_i + (\nu + \nu_t) n_j t_i \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - q n_i t_i + \frac{5}{3} k_a k n_i t_i + \left(1 + \frac{2}{3}\gamma\right)\omega_a \omega n_i t_i - 2 k_a \nu_t n_j t_i \left(\frac{\partial v_j}{\partial x_i}\right) - 2\omega_a \gamma \beta n_j t_i \left(\frac{\partial v_j}{\partial x_i}\right) \tag{A.121}$$

$$0 = u_t v_n + (\nu + \nu_t)\left(n_j \frac{\partial u_t}{\partial x_j} + t_i \frac{\partial u_n}{\partial x_i}\right) - 2(k_a \nu_t + 2\omega_a \gamma \beta) t_i \left(\frac{\partial v_n}{\partial x_i}\right) \tag{A.122}$$

$$u_t = -\frac{1}{v_n}\left[(\nu + \nu_t)\left(n_j \frac{\partial u_t}{\partial x_j} + t_i \frac{\partial u_n}{\partial x_i}\right) - 2(k_a \nu_t + 2\omega_a \gamma \beta) t_i \left(\frac{\partial v_n}{\partial x_i}\right)\right] \tag{A.123}$$

Finally the boundary conditions for the adjoint turbulence variables:

$$D^{k_a} = 0 = -u_i n_j \frac{1}{\omega}\left(\cancelto{0}{\frac{\partial v_i}{\partial x_j}} + \frac{\partial v_j}{\partial x_i}\right) + k_a v_j n_j + \frac{\partial k_a}{\partial x_j}(\sigma_k \nu_t + \nu) n_j - k_a n_j \cancelto{0}{\frac{\sigma_k}{\omega}\frac{\partial k}{\partial x_j}} - \omega_a n_j \cancelto{0}{\frac{\sigma_\omega}{\omega}\frac{\partial \omega}{\partial x_j}} \tag{A.124}$$

$$k_a = \frac{1}{v_n}\left[u_i \frac{1}{\omega}\frac{\partial v_n}{\partial x_i} - \frac{\partial k_a}{\partial x_j}(\sigma_k \nu_t + \nu) n_j\right] \tag{A.125}$$

$$D^{\omega_a} = 0 = \omega_a v_j n_j + \frac{\partial \omega_a}{\partial x_j}(\sigma_\omega \nu_t + \nu) n_j + \omega_a n_j \sigma_\omega \cancelto{0}{\frac{k}{\omega^2}\frac{\partial \omega}{\partial x_j}} + u_i n_j \frac{k}{\omega^2}\left(\cancelto{0}{\frac{\partial v_i}{\partial x_j}} + \frac{\partial v_j}{\partial x_i}\right) + k_a n_j \sigma_k \cancelto{0}{\frac{k}{\omega^2}\frac{\partial k}{\partial x_j}} \tag{A.126}$$

$$\omega_a = -\frac{1}{v_n}\left[\frac{\partial \omega_a}{\partial x_j}(\sigma_\omega \nu_t + \nu) n_j + u_i \frac{k}{\omega^2}\frac{\partial v_n}{\partial x_i}\right] \tag{A.127}$$

A.3.3. Slip wall

For the slip wall there is no Dirichlet boundary condition on the tangential velocity. Also, the rest of the primal variables have Neumann boundary conditions applied to them. Therefore we have:

$$\frac{\partial (n_j v_j)}{\partial \beta} = 0, \quad \frac{\partial}{\partial x_j}\frac{\partial p}{\partial \beta} = 0, \quad \frac{\partial}{\partial x_j}\frac{\partial k}{\partial \beta} = 0, \quad \text{and} \quad \frac{\partial}{\partial x_j}\frac{\partial \omega}{\partial \beta} = 0 \tag{A.128}$$

From this follows:

$$D_i^u = 0, \quad D_j^q = 0, \quad D_j^{k_a} = 0, \quad D_j^{\omega_a} = 0 \quad \text{and} \quad P_{ij}^u = 0 \tag{A.129}$$

Many of these we already have seen. First off, $D^q = 0$ and $P^u = 0$ suggest that the adjoint velocity must me zero along the slip wall, following the derivation for the inlet and no slip walls.

Now with this knowledge the other boundary conditions can be derived.

For q we have:

$$D_j^u = 0 = u_i v_j n_j + u_i v_i n_j + (\nu + \nu_t) n_j \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - q n_i + \frac{5}{3} k_a k n_i + \left(1 + \frac{2}{3} \gamma \right) \omega_a \omega n_i - 2(k_a \nu_t + \omega_a \gamma \beta) n_j \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \tag{A.130}$$

$$q = u_n v_n + u_i v_i + 2(\nu + \nu_t) n_j n_i \frac{\partial u_i}{\partial x_j} + \frac{5}{3} k_a k + \left(1 + \frac{2}{3} \gamma \right) \omega_a \omega - 4(k_a \nu_t + \omega_a \gamma \beta) n_j n_i \frac{\partial v_i}{\partial x_j} \tag{A.131}$$

A lot of these terms will drop out because of the boundary conditions on the slip wall for other variables. However no division by zero will be present so the outlet condition can be used here.

For k_a we have:

$$D^{k_a} = 0 = -u_i n_j \frac{1}{\omega} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + k_a v_j n_j + \frac{\partial k_a}{\partial x_j} (\sigma_k \nu_t + \nu) n_j - k_a n_j \frac{\sigma_k}{\omega} \frac{\partial k}{\partial x_j} - \omega_a n_j \frac{\sigma_\omega}{\omega} \frac{\partial \omega}{\partial x_j} \tag{A.132}$$

$$0 = n_j \frac{\partial k_a}{\partial x_j} \tag{A.133}$$

and for ω_a we have:

$$D^{\omega_a} = 0 = \omega_a v_j n_j + \frac{\partial \omega_a}{\partial x_j} (\sigma_\omega \nu_t + \nu) n_j + \omega_a n_j \sigma_\omega \frac{k}{\omega^2} \frac{\partial \omega}{\partial x_j} + u_i n_j \frac{k}{\omega^2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + k_a n_j \sigma_k \frac{k}{\omega^2} \frac{\partial k}{\partial x_j} \tag{A.134}$$

$$0 = n_j \frac{\partial \omega_a}{\partial x_j} \tag{A.135}$$

Bibliography

[1] G. Oldenziel A. Zubova, T. O'Mahoney. Cfd validation of flow downstream of an underflow gate. page 75, 12 2020.

[2] Michael Breuer, Nikolaus Peller, Ch Rapp, and Michael Manhart. Flow over periodic hills–numerical and experimental study in a wide range of reynolds numbers. *Computers & Fluids*, 38(2):433–457, 2009.

[3] Sai Hung Cheung, Todd A Oliver, Ernesto E Prudencio, Serge Prudhomme, and Robert D Moser. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliability Engineering & System Safety*, 96(9):1137–1149, 2011.

[4] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

[5] Karthikeyan Duraisamy, Ze J Zhang, and Anand Pratap Singh. New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting*, page 1284, 2015.

[6] WN Edeling, Pasquale Cinnella, and Richard P Dwight. Predictive rans simulations via bayesian model-scenario averaging. *Journal of Computational Physics*, 275:65–91, 2014.

[7] Wouter Nico Edeling, Pasquale Cinnella, Richard P Dwight, and Hester Bijl. Bayesian estimates of parameter variability in the k–ε turbulence model. *Journal of Computational Physics*, 258:73–94, 2014.

[8] JG Eriksson, RI Karlsson, and J Persson. An experimental study of a two-dimensional plane turbulent wall jet. *Experiments in fluids*, 25(1):50–60, 1998.

[9] Michael B Giles, Mihai C Duta, Jens-Dominik Muller, and Niles A Pierce. Algorithm developments for discrete adjoint methods. *AIAA journal*, 41(2):198–205, 2003.

[10] K Hanjalic, BE Launder, et al. A reynolds stress model of turbulence and its application to thin shear flows. *J. Fluid Mech*, 52(4):609–638, 1972.

[11] Marco A Iglesias, Kody JH Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.

[12] Mikael LA Kaandorp and Richard P Dwight. Data-driven modelling of the reynolds stress tensor using random forests with invariance. *Computers & Fluids*, 202:104497, 2020.

[13] B.E. Launder and W. Rodi. The turbulent wall jet. *Progress in Aerospace Sciences*, 19:81–128, 1979. ISSN 0376-0421. doi: https://doi.org/10.1016/0376-0421(79)90002-2. URL `https://www.sciencedirect.com/science/article/pii/0376042179900022`.

[14] Sophia Lefantzi, Jaideep Ray, Srinivasan Arunajatesan, and Lawrence Dechant. Estimation of k- ε parameters using surrogate models and jet-in-crossflow data. Technical report, Technical report, Sandia National Laboratories, Livermore, CA, USA, 2015.

[15] Florian R Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, 32(8):1598–1605, 1994.

[16] Todd Oliver and Robert Moser. Uncertainty quantification for rans turbulence model predictions. In *APS division of fluid dynamics meeting abstracts*, volume 62, pages LC–004, 2009.

[17] Todd A Oliver and Robert D Moser. Bayesian uncertainty quantification applied to rans turbulence models. In *Journal of Physics: Conference Series*, volume 318, page 042032. IOP Publishing, 2011.

[18] Eric J. Parish and Karthik Duraisamy. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305:758–774, 2016. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2015.11.012. URL https://www.sciencedirect.com/science/article/pii/S0021999115007524.

[19] Stephen B Pope. *Turbulent flows*. IOP Publishing, 2001.

[20] Svetlana Poroseva and Scott M Murman. Velocity/pressure-gradient correlations in a forans approach to turbulence modeling. In *44th AIAA Fluid Dynamics Conference*, page 2207, 2014.

[21] N. Rajaratnam. *Turbulent Jets*. ISSN. Elsevier Science, 1976. ISBN 9780080869964. URL https://books.google.nl/books?id=g3R4gFObzloC.

[22] Jaideep Ray, Sophia Lefantzi, Srinivasan Arunajatesan, and Lawrence Dechant. Bayesian parameter estimation of ak-ε model for accurate jet-in-crossflow simulations. *AIAA Journal*, 54(8):2432–2448, 2016.

[23] Jaideep Ray, Sophia Lefantzi, Srinivasan Arunajatesan, and Lawrence Dechant. Learning an eddy viscosity model using shrinkage and bayesian calibration: A jet-in-crossflow case study. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 4(1), 2018.

[24] Martin Schmelzer, Richard P Dwight, and Paola Cinnella. Discovery of algebraic reynolds-stress models using sparse symbolic regression. *Flow, Turbulence and Combustion*, 104(2):579–603, 2020.

[25] Emil Sekula and Jose Redondo. The structure of turbulent jets, vortices and boundary layer: Laboratory and field observations. *Nuovo Cimento C Geophysics Space Physics C*, 31:893–907, 09 2008. doi: 10.1393/ncc/i2009-10358-y.

[26] Anand Pratap Singh and Karthik Duraisamy. Using field inversion to quantify functional errors in turbulence closures. *Physics of Fluids*, 28(4):045110, 2016.

[27] Anand Pratap Singh, Karthikeyan Duraisamy, and Ze Jia Zhang. Augmentation of turbulence models using field inversion and machine learning. In *55th AIAA Aerospace Sciences Meeting*, page 0993, 2017.

[28] Anand Pratap Singh, Shivaji Medida, and Karthik Duraisamy. Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA journal*, 55(7):2215–2227, 2017.

[29] Jeffrey Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri Mavriplis. Cfd vision 2030 study: a path to revolutionary computational aerosciences, 2014.

[30] Brendan Tracey, Karthik Duraisamy, and Juan Alonso. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. In *51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 259, 2013.

[31] A. F. van Korlaar. Field inversion and machine learning in turbulence modelling, 2019.

[32] August Verhoff. The two-dimensional, turbulent wall jet with and without an external free stream. Technical report, Princeton Univ NJ, 1963.

[33] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids*, 2(3):034603, 2017.

[34] Q Wang and EA Dow. Quantification of structural uncertainties in the k- ω turbulence model. In *Proceedings of the Summer Program*, page 41, 2010.

[35] J Weatheritt and RD Sandberg. The development of algebraic stress models using a novel evolutionary algorithm. *International Journal of Heat and Fluid Flow*, 68:298–318, 2017.

[36] Jack Weatheritt and Richard Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship. *Journal of Computational Physics*, 325:22–37, 2016.

[37] David C Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA journal*, 26(11):1299–1310, 1988.

[38] David C Wilcox. Formulation of the kw turbulence model revisited. *AIAA journal*, 46(11):2823–2838, 2008.

[39] Jin-Long Wu, Heng Xiao, and Eric Paterson. Data-driven augmentation of turbulence models with physics-informed machine learning. *arXiv preprint arXiv:1801.02762*, 2018.

[40] AS Zymaris, DI Papadimitriou, Kyriakos C Giannakoglou, and C Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics*, 229(13):5228–5245, 2010.